IMPROVING CAMERA POSE ESTIMATION USING SWARM PARTICLE ALGORITHMS

A. Elashry, C. Toth

Dept. of Civil, Environmental and Geodetic Engineering, The Ohio State University,
470 Hitchcock Hall, 2070 Neil Avenue Columbus, OH 43210, USA - (Elashry.1, toth.2)@osu.edu

KEY WORDS: Camera pose estimation, Essential matrix, Swarm particle algorithms, Improved Gray Wolf Optimization (IGWO)

ABSTRACT:

Most computer vision and photogrammetry applications rely on accurately estimating the camera pose, such as visual navigation, motion tracking, stereo photogrammetry, and structure from motion. The Essential matrix is a well-known model in computer vision that provides information about the relative orientation between two images, including the rotation and translation, for calibrated cameras with a known camera matrix. To estimate the Essential matrix, the camera calibration matrices, which include focal length and principal point location must be known, and the estimation process typically requires at least five matching points and the use of robust algorithms, such as RANSAC to fit a model to the data as a robust estimator. From the usually large number of matched points, choosing five points, the Essential matrix can be determined based on a simple solution, which could be good or bad. Obtaining a globally optimal and accurate camera pose estimation, however, requires additional steps, such as using evolutionary algorithms (EA) or swarm algorithms (SA), to prevent getting trapped in local optima by searching for solutions within a potentially huge solution space.

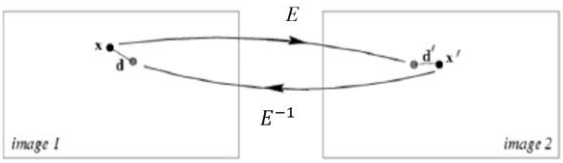
This paper aims to introduce an improved method for estimating the Essential matrix using swarm particle algorithms that are known to efficiently solve complex problems. Various optimization techniques, including EAs and SAs, such as Particle Swarm Optimization (PSO), Gray Wolf Optimization (GWO), Improved Gray Wolf Optimization (IGWO), Genetic Algorithm (GA), Salp Swarm Algorithm (SSA) and Whale Optimization Algorithm (WOA), are explored to obtain the global minimum of the reprojection error for the five-point Essential matrix estimation based on using symmetric geometric error cost function. The experimental results on a dataset with known camera orientation demonstrate that the IGWO method has achieved the best score compared to other techniques and significantly speeds up the camera pose estimation for larger number of point pairs in contrast to traditional methods that use the collinearity equations in an iterative adjustment.

1. INTRODUCTION

The determination of camera relative orientation is an important aspect in the fields of photogrammetry and computer vision, particularly in applications such as SfM, surface extraction, SLAM, and vision-based localization and navigation. In order to accomplish this, it is crucial to accurately estimate the relative orientation between images obtained from either a single camera (Kalantari et al., 2011) or multiple cameras (Zhao et al., 2017). This can be accomplished using the classical bundle adjustment or by employing the fundamental/essential matrix camera model based pose estimation. The essential matrix, which describes the geometric relationship between a pair of images, can be estimated using a variety of methods such as direct (Kalantari et al., 2011, Stewenius et al., 2006) or iterative methods (Helmke et al., 2004); both need a good initial values. The essential matrix-based estimation has become more important recently as it can be easily implemented in real-time applications. The RANSAC algorithm is often used to remove outliers and find the optimal model. The 3 x 3 essential matrix provides information about the rotation and translation between two cameras (Longuet-Higgins, 1981). Various methods have been developed for solving the problem of estimating the essential and fundamental matrices based on the number of corresponding points. In the case of calibrated cameras, the essential matrix can be directly computed since the camera matrix is known. However, for uncalibrated cameras where the camera matrix or interior parameters, focal length and principal point are unknown, the fundamental matrix estimation is used (Botterill et al., 2011). Note that image deformation due to lens distortion is always assumed to be removed prior. To determine the essential matrix, at least five-point correspondence is required and typically RANSAC is used to fit the model (Kruppa, 1913); note that seven-point and eight-point algorithms are also popular. The Levenberg-Marquardt optimization algorithm is utilized by some methods, such as (Botterill et al., 2011), which is repeatedly applied with random points to obtain the essential matrix. The first five-point algorithm for estimating the essential matrix was introduced by (Kruppa, 1913), which had eleven solutions and was later improved by (Faugeras and

Maybank, 1990) to have ten solutions. Additionally, (Nister, 2004) was the first algorithm that was well-suited for numerical implementation, using the roots of the tenth-degree polynomial. Accumulative errors are a major concern with the relative orientation of cameras in many applications, i.e., the errors increase with images/time; especially when a longer strip or block is formed for navigation purposes. The bundle adjustment is commonly used as a least squares solution for simultaneously orienting multiple images as well as estimating measured point coordinates in a mapping frame. However, without an accurate initial guess and ground control, it may lead to potential bending of the image strip/navigation path or even divergence. In pose estimation, identifying a good cost function and finding a minimum (global) of that cost function is a difficult task, and one must avoid the situation where the direct or iterative methods get stuck at a local minimum. The objective of this current study is to minimize the reprojection error (Butt et al., 2018) and obtain the global optimal solution. A cost function is shown in Equation 1 for the five-point essential matrix estimation

$$\sum_i d(x_i, \hat{x}_i)^2 + (x_i', \hat{x}_i')^2 \tag{1}$$



. Figure 1. Symmetric geometric error

where x_i and $\hat{x}_i$ are two corresponding points in the two images, and x_i' and $\hat{x}_i'$ are assumed to be correct corresponding points under the constraint (E is the Essential matrix):

$$x_i'^{T} E x_i = 0 \tag{2}$$

The cost function chosen for this study is the Symmetric geometric error cost function, see Equation 3 (Butt et al., 2018):

$$\sum_i \frac{\left(x_i'^{T} E x_i\right)^2}{(Ex_i)_1^2 + (Ex_i)_2^2 + \left(E^T x_i'\right)_1^2 + \left(E^T x_i'\right)^2} \tag{3}$$

In this study, two types of global optimization techniques, namely evolutionary algorithms (EA) and swarm algorithms (SA), are examined for their effectiveness in solving optimization problems (Nakane et al., 2020). EAs and SAs are classified as metaheuristic optimization algorithms, which use an initial candidate solution and create random solutions to search for a solution in the solution space. They evaluate the solutions using cost functions to avoid being trapped in local minima (Saremi et al., 2017). Population-based metaheuristics optimization has advantages such as simplicity, flexibility, and the ability to avoid local optima, making it an important optimization and search technique (Saremi et al., 2017). EAs are based on the principle of survival of the fittest, while SAs are inspired by natural phenomena, such as animal behaviours. Examples of SAs include Gray Wolf Optimization (GWO) and Whale Optimization Algorithm (WOA). These techniques do not require gradient or derivative search spaces to obtain optimal solutions, thus avoiding expensive or derivative computations. To efficiently estimate the camera pose, genetic and swarm particle algorithms are used to improve the estimation of *E*, i.e., to reduce the error in the camera pose as well as being computationally inexpensive. Our experimental results demonstrate that the Improved Gray Wolf optimization (IGWO) technique outperforms the other techniques.

2. EVOLUTIONARY ALGORITHMS

As depicted in Figure 2, Evolution algorithms are part of the population-based Metaheuristics algorithm family, alongside Swarm particle algorithms that emulate animal behaviors. Population based Metaheuristics algorithms utilize multiple candidate solutions to locate the optimal solution while preventing local optima. Metaheuristic algorithms are mostly inspired by the biological evolution process (Michalewicz and Schoenauer, 1996).

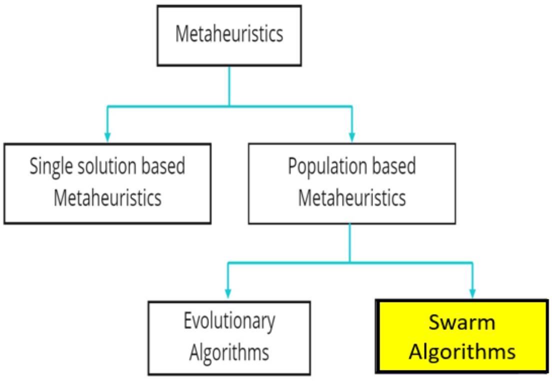

Metaheuristics
Single solution based Metaheuristics
Population based Metaheuristics
Evolutionary Algorithms
Swarm Algorithms

Figure 2. Classification of Metaheuristics algorithms

2.1 Genetic algorithm

The genetic algorithm (GA) is a specific type of evolutionary algorithm that simulates the process of biological evolution in the numerical domain. It is commonly utilized for solving optimization problems and is strongly influenced by the Darwinian theory. Introduced by (Michalewicz and Schoenauer, 1996), the GA can be used to solve a variety of problems. All evolutionary algorithms commence with initialization, which involves generating random solutions for optimization problems. In the case of the GA, an initial population of randomly generated solutions to the optimization problem is created. For instance, the initial guess of the Essential matrix is used to initialize a population of random solutions. Subsequently, the initial population is passed to the evolution loop, which is illustrated in Figure 3; note that the algorithm iterates through the loop until certain criteria are met.

In the first step of the loop, the selection is performed based on the cost function, where members of the initial population with the best cost function are chosen using various selection criteria such as Roulette Wheel selection (Jebari and Madiafi, 2013). The better members of the population influence the selection criteria, with superior individuals having a higher chance of being selected. The algorithm then undergoes a reproduction process where offspring are generated by combining parent solutions, much like in nature. The offspring may cooperate to create new solutions that are potentially better than that of their parents and inherit their advantages. The processes of mating and crossing over are simulated to produce offspring, which can be performed in various ways, such as single-point crossover, double-point crossover, and uniform crossover (Soon et al., 2013). The mutation process randomly alters certain genes to enable the algorithm to explore and create innovative solutions to the problem. Another selection process is conducted using the population of offspring and the initial population of parents, thus completing the evolution loop. The algorithm terminates execution based on the termination criteria. The flowchart of the GA algorithm is presented in Figure 4.

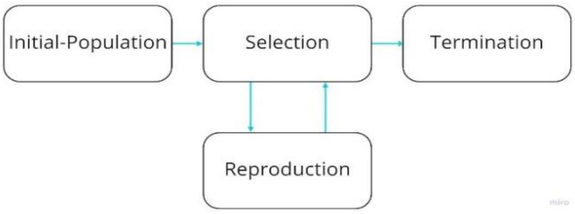


Figure 3. Evolution algorithm loop

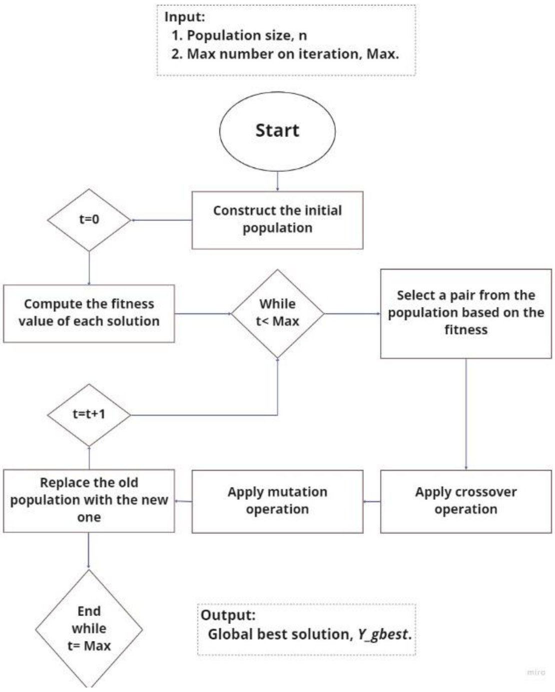

Input:
1. Population size, n
2. Max number on iteration, Max.
Start
Construct the initial population
t=0
Compute the fitness value of each solution
While t< Max
Select a pair from the population based on the fitness
t=t+1
Replace the old population with the new one
Apply mutation operation
Apply crossover operation
End while t= Max
Output:
Global best solution, Y_gbest.

Figure 4. Flowchart of GA algorithm

3. PARTICLE SWARM ALGORITHMS

3.1 Particle swarm optimization (PSO)

Particle swarm algorithms (PSO) belong to the category of population based Metaheuristics optimization algorithms, which are inspired by the social behavior of intelligent swarms, such as packs of wolves, fish schools and bird flocks were first introduced by (Eberhart and Kennedy, 1995, Kennedy and Eberhart, 1995). The PSO algorithm aims to search the solution space based on the cost or objective function and needs an initial guess of the solution. The swarm members positions and velocities are updated based on the environment's changing conditions and response (Wang et al., 2018). The swarm continues to search for the optimal solution with unrestricted movement in the search space, while maintaining stable movement. However, if there is a significant change in the search process, the swarm can adjust its movement. In PSO, each member of the swarm is a particle representing a potential solution in the search space and is initiated randomly from the initial guess. The movement of the particles (position and velocity) is determined based on the cost function values (Blackwell et al., 2007). Additionally, each particle can remember its optimal position, p_{best}, and update its new position and velocity based on the swarm's global optimal position, g_{best}, as shown in the following steps:

$$v_x = v_x - 2 \times rand \times (p_{bestx} - x) + 2 \times rand \times (g_{bestx} - x) \quad (4)$$

$$x = x + v_x \quad (5)$$

The flowchart in Figure 5 explains the PSO steps to get the global optimal solution.

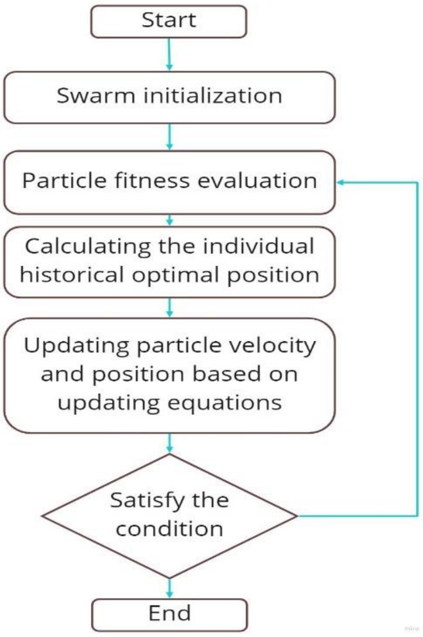

Start
Swarm initialization
Particle fitness evaluation
Calculating the individual historical optimal position
Updating particle velocity and position based on updating equations
Satisfy the condition
End

Figure 5. Flowchart of PSO algorithm

3.2 Gray Wolf Optimization (GWO)

The Gray Wolf Optimization (GWO) algorithm, as described in (Mirjalili et al., 2014), is a type of Metaheuristic optimization algorithm inspired by the hunting behaviour of grey wolves (Canis lupus). The algorithm emulates the leadership hierarchy in a pack of wolves, which is divided into four types of wolves: alpha α, beta β, delta δ, and omega ω. Alpha is the leader and responsible for decision-making and managing the pack and is considered the fittest solution. Beta is the second-best solution and serves as an advisor to the leader. Delta plays a crucial role in the pack by functioning as hunters, scouts, and sentinels, and is considered the third-best solution. Omega is the rest of the candidate solutions and serves to protect the pack. The GWO algorithm begins with a random initial solution, or population, that contains candidate solutions with information about the solution space. The algorithm then evolves over time through iterations based on information sharing to avoid getting stuck in a local optimum solution. The algorithm follows the phases of hunting, which include tracking, chasing, pursuing, and encircling the prey, until finally attacking the prey, as shown in Figure 6.

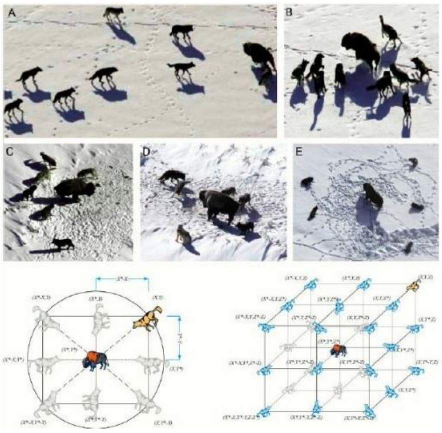


Figure 6: Grey wolves hunting behaviour and their possible next locations in 2D and 3D positions.

These three best solutions (α,β, and δ) help the other candidates to update their positions (solutions). The following equations are utilized to get the optimal solution.

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$$

where A and C are coefficient vectors, X_p is the prey (cost function) position vector, and X_n is the grey wolf position. $\vec{A} = 2\vec{a} \cdot \overrightarrow{r_1} - \vec{a}$, $\vec{C} = 2 \cdot \overrightarrow{r_2}$ and X_α,X_β,X_δ are the best three solutions. where components of $\vec{a}$ are linearly decreased from 2 to 0 over the iterations, and r1, and r2 are random vectors in [0, 1]. $\overrightarrow{A}$ is a random value in the interval [-2a, 2a] which approaching the prey can be controlled by decreasing the value of a, which affects the value of A, therefore updating the position. $|A| < 1$ means the wolves approach the prey. Also, in the searching process for the prey, we try to adjust $|A| > 0$ to force the wolf to diverge from the prey trying to find the best prey or best candidates, the flowchart of the algorithm is shown in Figure 7, which shows the algorithm steps for the whole population (N). For more details on GWO, see (Mirjalili et al., 2014).

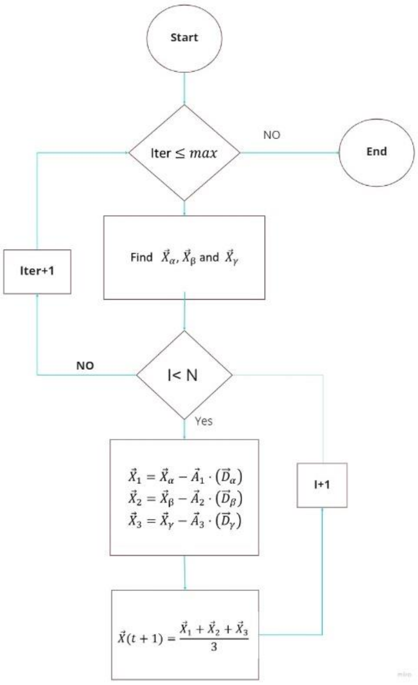

Start
Iter ≤ max
NO
End
Find $\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\gamma$
Iter+1
NO
I< N
Yes
$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha)$
$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta)$
$\vec{X}_3 = \vec{X}_\gamma - \vec{A}_3 \cdot (\vec{D}_\gamma)$
I+1
$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$

Figure 7. Flowchart of the GWO algorithm

3.3 Improved Gray Wolf optimization (IGWO)

GWO is a search technique that is considered powerful because it uses the three best solutions in each iteration for guidance. However, GWO suffers from an imbalance between exploration and exploitation, as noted in previous studies (Fister Jr et al., 2013, Tu et al., 2019). To address this limitation, Dimension Learning based Hunting (DLH) (Nadimi-Shahraki et al., 2021) proposes a new movement search strategy that differs from the GWO algorithm, which is based on the hunting behaviour of individual wolves. DLH suggests several approaches to create a neighbourhood for each wolf, enabling the sharing of information between the wolves, which allows the candidate solution to learn from its neighbours. This strategy aims to enhance global search in comparison to local search. Along with the $X_{i-GWO}(t)$ candidate solution from GWO, DLH proposes a new candidate solution $X_{i-DLH}(t)$ based on the Euclidean distance (radius) R between the current position $X_i(t)$ and the best candidate position $X_{i\text{–}GWO}(t)$. The specific equation for this candidate solution is shown in Equation 6.

$$R_i(t) = ||X_i(t) - X_{i\text{–}GWO}(t+1)|| \qquad (6)$$

Then the neighbour's $N_i(t)$ based on the distances between the wolves as shown in Equation 7 is constructed, and then the new solution is calculated where $X_{n,d}(t)$ is selected randomly from the neighbours and $X_{r,d}(t)$ from the population as shown in Equation 8.

$$N_i(t) = \{X_j(t) \mid D_i\,(X_i(t),X_j(t))R_i(t),X_j(t) \in Population\} \quad (7)$$

$$Xi{-}DLH,d(t{+}1) = Xi{-}d(t){+}rand{\times}(Xn,d(t){-}Xr,d(t)) \qquad (8)$$

The selection and updating phase, which involves comparing the fitness functions of the two candidate solutions from GWO $X_{i\text{–}GWO}(t)$ and DLH $X_{i\text{–}DLH,d}(t)$ using Equation 9, is used to determine the best candidate solution. This phase is essential in the algorithm, and it helps choose the better solution between $X_{i\text{–}GWO}(t)$ and $X_{i\text{–}DLH,d}(t)$. The algorithm steps are illustrated in the following flowchart. More information on IGWO can be found in (Nadimi-Shahraki et al., 2021).

$$X_i(t+1) = \begin{cases} X_{i-GWO}(t+1), & \text{if } f(X_{i-GWO}) < f(X_{i-DLH}) \\ X_{i-DLH}(t+1) & \text{otherwise} \end{cases} \qquad (9)$$

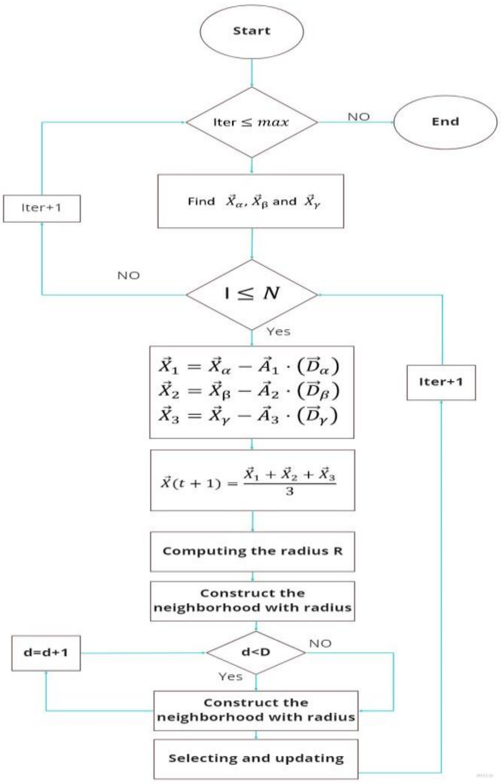

Start
Iter $\leq max$
NO
End
Iter+1
Find $\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\gamma$
NO
$\mathrm{I} \leq N$
Yes
$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha)$
$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta)$
$\vec{X}_3 = \vec{X}_\gamma - \vec{A}_3 \cdot (\vec{D}_\gamma)$
Iter+1
$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$
Computing the radius R
Construct the neighborhood with radius
d=d+1
d<D
NO
Yes
Construct the neighborhood with radius
Selecting and updating

Figure 8. The flowchart of the IGWO algorithm

4. EXPERIMENTAL RESULTS

The objective of this research is to propose a method for improving the accuracy of camera pose estimation using various metaheuristic optimization algorithms. These algorithms are employed to obtain the global optimal solution (relative orientation), which is determined by minimizing the reprojection errors represented by the symmetric geometric error cost function. Since there is no single optimization technique suitable for solving all problems, multiple metaheuristic optimization algorithms that are suited for addressing complex problems were tested. The experimental dataset used the SPIN Lab dataset included image pairs with known camera intrinsic parameters and lens distortion removed; example images are shown in Figure 9. The known camera poses were used as a reference to evaluate the performance of the metaheuristic optimization algorithms. The pose estimation accuracy was further improved by the metaheuristic optimization algorithms applied to obtain more precise results compared to the initial approximations obtained from the traditional methods. For this study, different numbers of populations were utilized as candidate solutions generated from the initial solution by the five-point algorithm (Kukelova et al., 2008). Figure 10 illustrates that the cost function experienced a significant reduction, and after conducting multiple trials with 100 iterations across all experiments, the Improved Grey Wolf Optimization (IGWO) algorithm achieved the minimum for the selected cost function. IGWO also achieved the best mean square error of 0.029, with a cost function of $2.701x10^{-5}$, outperforming both PSO and GWO algorithms. Conversely, the WOA algorithm had poor performance. With respect to the time taken by the algorithms, as illustrated in Figure 12, it took the GWO algorithm around 0.2 seconds to initiate the camera pose estimation by the essential matrix, then calculating the relative orientation and performing triangulation, the IGWO method took longer as it included an additional step for improving accuracy.

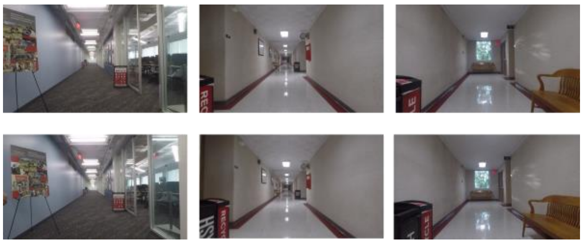


Figure 9. Samples of SPIN Lab dataset

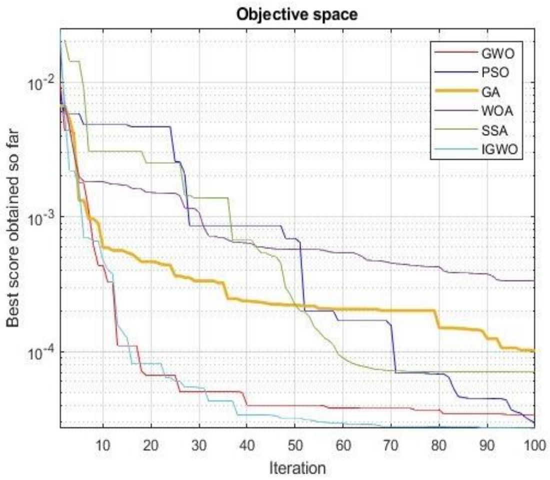


Figure 10. Results obtained from metaheuristic optimization algorithms.

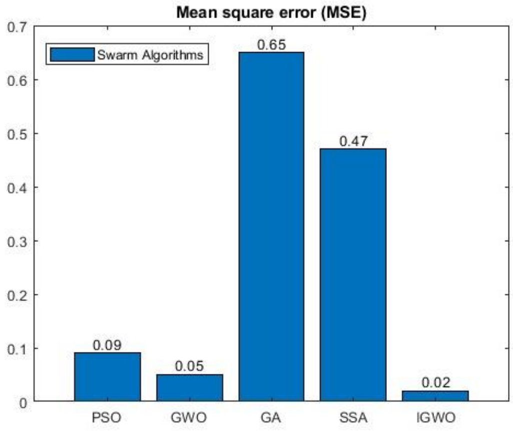


Figure 11. Histogram of comparing between the algorithms.

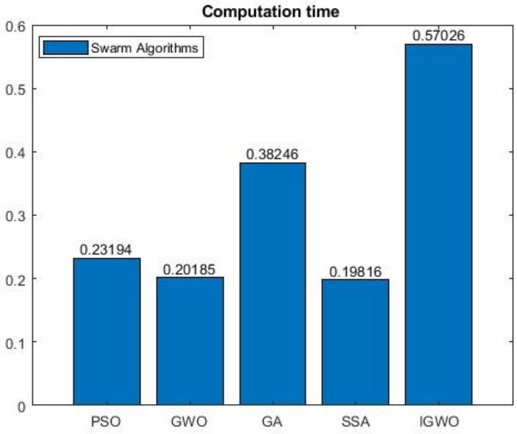


Figure 12. Histogram of computational time for the different algorithms

5. CONCLUSION

This study proposes a robust algorithm that employs particle swarm algorithms (PSA) that provide faster solutions compared to classical computations for pose estimation between two cameras. The PSA is inspired by natural behaviors observed in animals, birds, and insects. However, since the performance of PSA can vary widely for more complex problems, various PSAs were applied to identify the optimal solution for the problem at hand. The results of multiple trials indicate that the Improved Grey Wolf Optimization (IGWO) algorithm has the most accurate solution, followed by Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO). To compare the algorithms, they were applied to the same dataset and the results, as shown in Figures 10-11, demonstrate a significant difference in performance when using the PSA algorithms. These results were obtained by evaluating the performance of different optimization algorithms that utilize animal behaviors to solve optimization problems. The algorithms were compared based on their strategy and approach to solving the problem, and their performance was measured by the achieved cost function and mean square error as shown in Figure 11.

REFERENCES

Blackwell, T. M., Kennedy, J. and Poli, R., 2007. Particle swarm optimization. Swarm Intelligence 1(1), pp. 33–57.

Botterill, T., Mills, S. and Green, R., 2011. Fast ransac hypothesis generation for essential matrix estimation. In: 2011 International Conference on Digital Image Computing: Techniques and Applications, IEEE, pp. 561–566.

Butt, M. M., Hai, Z. and XiaoChen, Q., 2018. A benchmark for rotation extraction between two images in visual navigation. In: 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), IEEE, pp. 1–4.

Eberhart, R. and Kennedy, J., 1995. A new optimizer using particle swarm theory. In: MHS’95. Proceedings of the sixth international symposium on micro machine and human science, Ieee, pp. 39–43.

Faugeras, O. D. and Maybank, S., 1990. Motion from point matches: multiplicity of solutions. International Journal of Computer Vision 4(3), pp. 225–246.

Fister Jr, I., Yang, X.-S., Fister, I., Brest, J. and Fister, D., 2013. A brief review of nature-inspired algorithms for optimization. arXiv preprint arXiv:1307.4186.

Helmke, U., Huper, K., Lee, P., Moore, J. et al., 2004. Essential¨ matrix estimation via newton-type methods. In: 16th International Symposium on Mathematical Theory of Network and System (MTNS), Leuven.

Jebari, K. and Madiafi, M., 2013. Selection methods for genetic algorithms. International Journal of Emerging Sciences 3(4), pp. 333–344.

Kalantari, M., Hashemi, A., Jung, F. and Guedon, J.-P., 2011.´ A new solution to the relative orientation problem using only 3 points and the vertical direction. Journal of Mathematical Imaging and Vision 39(3), pp. 259–268.

Kennedy, J. and Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, Vol. 4, pp. 1942–1948 vol.4.

Kruppa, E., 1913. To determine an object from two perspectives with inner orientation. Holder.¨

Kukelova, Z., Bujnak, M. and Pajdla, T., 2008. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In: BMVC, Vol. 2number 5, p. 2008.

Longuet-Higgins, H. C., 1981. A computer algorithm for reconstructing a scene from two projections. Nature 293(5828), pp. 133–135.

Michalewicz, Z. and Schoenauer, M., 1996. Evolutionary algorithms for constrained parameter optimization problems. Evolutionary computation 4(1), pp. 1–32.

Mirjalili, S., Mirjalili, S. M. and Lewis, A., 2014. Grey wolf optimizer. Advances in engineering software 69, pp. 46–61.

Nadimi-Shahraki, M. H., Taghian, S. and Mirjalili, S., 2021. An improved grey wolf optimizer for solving engineering problems. Expert Systems with Applications 166, pp. 113917.

Nakane, T., Bold, N., Sun, H., Lu, X., Akashi, T. and Zhang, C., 2020. Application of evolutionary and swarm optimization in computer vision: a literature survey. IPSJ Transactions on Computer Vision and Applications 12(1), pp. 1–34.

Nister, D., 2004. An efficient solution to the five-point relative´ pose problem. IEEE transactions on pattern analysis and machine intelligence 26(6), pp. 756–770.

Saremi, S., Mirjalili, S. and Lewis, A., 2017. Grasshopper optimisation algorithm: theory and application. Advances in engineering software 105, pp. 30–47.

Soon, G. K., Guan, T. T., On, C. K., Alfred, R. and Anthony, P., 2013. A comparison on the performance of crossover techniques in video game. In: 2013 IEEE international conference on control system, computing and engineering, IEEE, pp. 493–498.

Stewenius, H., Engels, C. and Nister, D., 2006. Recent devel- ´ opments on direct relative orientation. ISPRS Journal of Photogrammetry and Remote Sensing 60(4), pp. 284–294.

Tu, Q., Chen, X. and Liu, X., 2019. Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection. IEEE Access 7, pp. 78012–78028.

Wang, D., Tan, D. and Liu, L., 2018. Particle swarm optimization algorithm: an overview. Soft computing 22(2), pp. 387–408.

Zhao, C., Fan, B., Hu, J., Tian, L., Zhang, Z., Li, S. and Pan, Q.,
2017. Pose estimation for multi-camera systems. In: 2017 IEEE International Conference on Unmanned Systems (ICUS), IEEE, pp. 533–538.