IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 April 2022, accepted 25 June 2022, date of publication 29 June 2022, date of current version 11 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3187093

RESEARCH ARTICLE

A Novel Bundle Adjustment Approach Based on Guess-Aided and Angle Quantization Multiobjective Particle Swarm Optimization (GAMOPSO) for 3D Reconstruction Applications

MAHER ALNDIWEE, MOHAMED MAZEN AL-MAHAIRI, AND RAOUF HAMDAN
Department of Computer and Automation Engineering, Faculty of Mechanical and Electrical Engineering, Damascus University, Damascus, Syria
Corresponding author: Maher Alndiwee (Maher.Alndiwee@gmail.com)

ABSTRACT The 3D reconstruction process is very important in a variety of computer vision applications. Bundle adjustment has a significant impact on 3D reconstruction processes, namely in Simultaneously Localization and Mapping (SLAM) and Structure from Motion (SfM). Bundle adjustment, which optimizes camera parameters and 3D points as a very important final stage, suffers from memory and efficiency requirements in very large-scale reconstruction. Multi-objective optimization (MOO) is used in solving a variety of realistic engineering problems. Multi-Objective Particle Swarm Optimization (MOPSO) is regarded as one of the state of the art for meta-heuristic MOO. MOPSO has utilized the concept of crowding distance as a measure to differentiate between solutions in the search space and provide a high level of exploration. However, this method ignores the direction of the exploration which is not sufficient to effectively explore the search space. In addition, MOPSO starts the search from a fully randomly initialized swarm without taking any prior knowledge about the initial guess into account, which is considered impractical in applications where we can estimate initial values for solutions like bundle adjustment. In this paper, we introduced a novel hybrid MOPSO-based bundle adjustment algorithm that takes advantage of initial guess, angle quantization technique, and traditional optimization algorithms like RADAM to improve the mobility of MOPSO solutions; the results showed that our algorithm can help improve the accuracy and efficiency of bundle adjustment (BA).

INDEX TERMS 3D reconstruction, bundle adjustment, multiobjective optimization (MOO), MOPSO.

I. INTRODUCTION

Fine-tuning is the procedure in which parameters of the system should be modified precisely to match specific observations. The solver of these problems usually starts from the initial values of the parameters and adjusts them to fit the model with given observations. A well-known fine-tuning technique called Bundle Adjustment is considered one of the most important procedures to improve the precision of the 3D reconstruction process. The bundle adjustment was solved as an NLSE problem using many iterative methods like Gauss-Newton, Levenberg-Marquardt, and others. These methods are sometimes vulnerable to fall in local minima especially when the initial solution has poor quality, so they become a bottleneck for computation and memory. [1]

The Particle Swarm Optimization algorithm, which was inspired by the motion of bird flocks and schooling fish, was originally suggested by Eberhart and Kennedy in 1995. Since then, it has attracted the attention of many researchers around the world [2]. Due to its efficiency, PSO has been used in a variety of applications to date. A multi-objective version was proposed later [3] to adapt the algorithm to solve multi-objective optimization problems. Since then, many

The associate editor coordinating the review of this manuscript and approving it for publication was Ran Cheng.

improvements were proposed to the MOPSO which enabled it to be used in different applications [4]. This research aims to improve the quality of the Bundle Adjustment process using an improved version of MOPSO. So, in this paper, we will present how to adapt the MOPSO algorithm to solve fine-tuning problems, and we will test the modified algorithm on standard multi-objective test functions as proof of concept, then we will use it to solve the bundle adjustment. The key contributions of this study can be summarized as follows.

- proposing the concept of guess-aided initialization to evolutionary algorithms to adopt them to fine-tuning problems.
- Introducing a generalized version of the angle quantization concept and deploying it in MOPSO.
- Proposing a novel approach for selecting the leader that can help to improve the exploitation abilities of the MOPSO based on hybridization with traditional gradient descent approaches.
- Providing a new method to solve the bundle adjustment problem in a more efficient way than the traditional methods.

The remainder of this paper is organized as follows. Section II presents the background. While Section III presents the proposed GAMOPSO algorithm followed by Section IV experiment sets for both standard MOO test functions and bundle adjustment, in Section V we will show the results and discussions, followed by Section VI which presents the summary and conclusions.

II. BACKGROUND

A. MULTI-OBJECTIVE OPTIMIZATION

Optimization problems that have at least two objective functions are called multi-objective optimization problems. These objectives sometimes conflict with each other. In other words, they are represented by different units of measurement and may have the same importance in the decision-making process. [1] Assuming that we want to minimize the objective functions, the problem of multi-objective optimization can be represented as follows:

$$Min\, f(x) := [f_1(x), f_2(x)..,f_k(x)] \tag{1}$$

Subject to the following restrictions:

$$g_i(x) \leq 0;\ i = 1,2,\ldots, m \tag{2}$$

$$h_j(x) = 0;\ j = 1,2,\ldots,p \tag{3}$$

where**:**

$x = [x1, x2, \ldots, xn]$ represents the optimized variables vector

f_i:$Rn \rightarrow R$, $i= 1, 2, \ldots, k$ objective functions to be minimized

g_i, h_j:$Rn \rightarrow R$,$i= 1, 2, \ldots, m$, $j= 1, 2, \ldots, p$ inequalities and equations of constraints imposed on the problem.

A decision vector $\bar{x}$ is called Pareto-optimal if it is not dominated by any other decision vector, the final result of the optimization problem is a set of Pareto-optimal vectors called Pareto-front. [2]

B. PREVIOUS WORK

Conventional bundle adjustment methods are mainly based on the Levenberg-Marquardt algorithm or its improved versions, which mainly aim to reduce its computational complexity by manipulating the structure of matrices or using the reduced camera system and other methods. The Levenberg algorithm uses the sum of squares of the projection error as a cost function. So, for n cameras and m points, Levenberg has $O((m + n)^3)$ complexity and $O((m + n)m.n)$ memory usage. Many methods have been introduced to reduce the complexity of BA, including the Schur complement method, which helps reduce the complexity to $O(m^3 + mn)$ with memory usage $O(m.n)$, and it can be reduced more depending on secondary sparse structures. [3] Despite the possibility of reducing complexity, iterative methods are often vulnerable to fall in local minima, especially if initial estimates are inaccurate, which negatively affects the accuracy of the bundle adjustment process using it.

Many researchers investigated the use of random search algorithms due to their ability to be immunized against local minima. The researchers in [4] used the particle swarm optimization algorithm (PSO) to find the rotation angles, they used the sum of the absolute projection error as an objective function. The algorithm helped reduce the number of control points needed to find the camera parameters, but the solutions were only for the camera orientation and the values were far from the ground truth values, which means a failure of the bundle adjustment. While in [5] the authors provided a bundle adjustment algorithm for aerial panoramic images based on the genetic algorithm and the objective function was the sum of the absolute value of the residual errors, they focused on estimating the orientation of the camera. In [6] the adaptive torque researchers used the PSO algorithm, in addition to the techniques of crossing over and mutations used from the Genetic Algorithm (GA) to increase the algorithm's exploration capabilities, and this algorithm is used to improve the initial estimations which later used to solve the bundle adjustment using the Gauss-Newton method. Researchers in [7] studied the performance of different optimization algorithms to solve the problem of bundle adjustment in reconstructing panoramic images. These algorithms include; Bat Algorithm BA, Gray Wolf Optimizer (GWO), Arithmetic Optimization Algorithm (AOA), Salp Swarm Algorithm (SSA), and Particles Swarm optimization algorithm (PSO). They concluded that the optimization algorithms gave better performance than the traditional Levenberg algorithm. However, this method is suitable for reconstructing panoramic images resulting from a purely rotational motion without translation.

It is worth noticing that most of the previous work focused on the orientation of the camera or special applications, and to overcome this shortage, we are proposing a novel bundle adjustment algorithm based on a modified version of MOPSO as illustrated in the next section.

III. PROPOSED ALGORITHM

The proposed algorithm is a modified version of MOPSO that can handle fine-tuning engineering problems. Algorithm 1 shows the basic steps of the Guess-aided, Angle quantization-based Multi-Objective Particle Swarm Optimization (GAMOPSO) algorithm, where the algorithm starts the process of initializing the initial swarm based on the initial guesses and with a percentage of the swarm determined by the InitPercent parameter, as this percentage of the initial particles is created according to a Gaussian distribution in the adjacent area of the given initial guess.

After that, the process of determining dominant solutions begins through the DetermineDomination method, where solution A is considered to dominate solution B if the values of all objective functions of solution A are better or equal to the corresponding values of objective functions of solution B, and there is at least one value of objective functions of A that is better than the corresponding value of the objective functions of solution B. To this extent, the initialization process for the initial swarm is finished, and the algorithm's iterative cycles begin as follows at each iteration:

1- Create a copy of the current swarm and call it newSwarm

2- For each particle p in the new swarm apply the following steps:

- Choose a leader for the particle
- Calculate the speed according to (4):

$$v_{ij}= wv_{ij}+c_1r\left(p_{ij}-x_{ij}\right)+c_2r\left(p_{nj}-x_{ij}\right) \tag{4}$$

- Move the particle p according to (5):

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij} \tag{5}$$

- Apply the mutation on the particle if it is within the mutationPercent portion, which gives a new particle p'
- If p' is better than p, put p' instead of p in the new swarm

3- Merge current swarm and newSwarm into Swarm

4- Calculate the Ranks according to the number of solutions dominating each solution within Swarm

5- Calculate the crowding distance for solutions

6- Calculate the angle quantization for solutions.

7- Extracting new solutions with a number equal to numberOfParticles from Swarm arranged according to the rank. When the ranks are equal and we need to choose one of the solutions, we can use the crowding distance, angle quantization, or both (based on the variant of GAMOPSO used), and the resulting swarm will be used as a current swarm for the next iteration.

8- Finding a Pareto front, or in other words, ranked first solutions from the resulting swarm which resample the best solutions found by the algorithm.

Algorithm 1 GAMOPSO

```
Input:
(1)InitGuess: the initial guess.
(2)InitPercent: the percent of initial solutions.
(3)maxItritions: maximum number of iterations.
(4)numberOfParticles: the number of particles in the swarm.
(5)mutationPercent: the mutation percentage.
(6)STD: the standard deviation of the Gaussian distribution.
Output:
(1)ParetoFront : the optimal solution found by the algorithm.
1: Start Algorithm
2: Swarm = GenerateInitialSwarm(InitGuess; InitP ercent;
numberOfParticles; STD)
3: Swarm=DetermineDomination(Swarm)
4: for itr=1:maxItritions do
5:    newSwarm=Swarm
6:    for i=1:numberOfParticles do
7:       p=newSwarm(i)
8:       pL=SelectLeader(p)
9:       V=computeVelocity(p,pL)
10:      p=moveParticle(p,V)
11:      if rand<mutationPercent then
12:         p'= mutate(p)
13:         if Dominates(p',p) then
14:            newSwarm(i)=p'
15:         else if Dominates(p,p') then
16:            newSwarm(i)=p
17:         else if rand >0.5 then
18:            newSwarm(i)=p'
19:         end if
20:      end if
21:   end for
22:   Swarm = [Swarm newSwarm]
23:   Ranks=FindRankedParetoFronts(Swarm)
24:   Distances=CalculateCrowdingDistance(Swarm)
25:   Angles=CalculateCrowdingAngels(Swarm)
26:   Swarm=ExtractNextPopulation(Swarm,Ranks, Distances,Angles)
27:   ParetoFront=FindPareto(Swarm)
28: end for
29: End Algorithm
```

A. STRUCTURELESS BUNDLE ADJUSTMENT

The first challenge developers face in any optimization problem is usually to formulate the problem relevantly, and, as we showed in the second section of this article, the representation of the bundle adjustment problem requires optimizing the camera variables and structure points, so the size of the optimized variable will be for n cameras and m points 6n+3m. When working on a large scale, we will have thousands of points and hundreds of camera positions at least, and therefore the use of the traditional formulation of the problem will have a very high computational cost, so an alternative representation could be used, which is found in [8] where they algebraically proved the possibility of eliminating structure points through the use of the three-view constraints shown in Figure 1 and Equations (6-8):

$$g_{2v}\left(x_k,x_l,z_k^j,z_l^j\right) = q_k\cdot\left(t_{k\to l}\times q_l\right) \tag{6}$$

$$g_{2v}\left(x_l,x_m,z_l^j,z_m^j\right) = q_l\cdot\left(t_{l\to m}\times q_m\right) \tag{7}$$

$$g_{3v}\left(x_k,x_l,x_m,z_k^j,z_l^j,z_m^j\right) = \left(q_l\times q_k\right)\cdot\left(q_m\times t_{l\to m}\right) - \left(q_k\times t_{k\to l}\right)\cdot\left(q_m\times q_l\right) \tag{8}$$

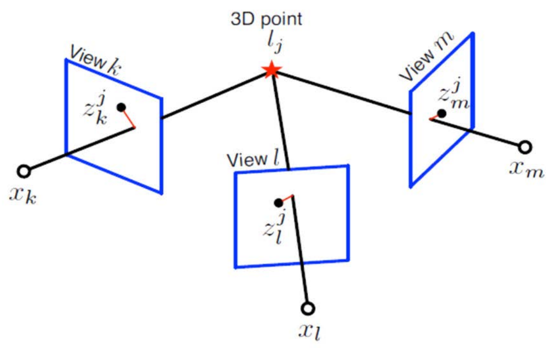

FIGURE 1. The three views constrain illustration [8]: the 3D point lj seen by the three views zk, zl, and zm. So, the constrain can be applied to find the best relative location of the three views.

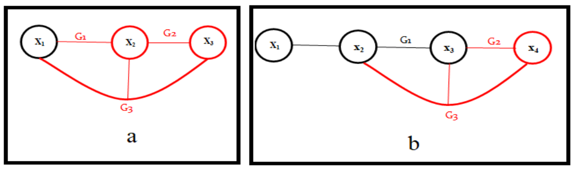

FIGURE 2. Solution representation in GAMOPSO, (a). the first call of GAMOPSO x_1 is fixed x_2, and x_3 are varying. (b). the second call of GAMOPSO x_1, x_2, and x_3 are fixed only x_4 is varying.

where:

$$q_i \doteq R_i^T K_i^{-1} p_i \tag{9}$$

g_{2_v} : A two-view constraint that matches points from two scenes

g_{3_v} : A three-view constraint that matches points from three scenes.

$t_{k\rightarrow l}$: The linear transition vector from frame k to frame l.

$\mathrm{R_i}$: the rotation matrix of the camera i.

$\mathrm{K_i}$: the matrix of intrinsic parameters of the camera i.

$\mathrm{P_i}$: the positions of the pixels corresponding to the structure points in the camera i.

It can be seen that this representation greatly helps in reducing the size of the optimized variables, as we only optimize the camera variables instead of the camera variables and the structure points, and thus this representation will reduce the computational and memory requirements significantly.

Further reduction was gained by sequentially processing the coming frames. In the first call of GAMOPSO, it processes the first three frames as in monocular bundle adjustment we can predict the position of the frames up to a specific scale, the first frame (x_1) is fixed as the reference frame and frames (x_2, x_3) position and orientation were optimized relatively to x1 to minimize the three objectives functions (G_1, G_2, G_3) as shown in Figure (2-a). On the other hand, when a new frame arrives, GAMOPSO will be called to optimize the position and orientation of the new frame (x_4 for example) to minimize only two of the objective functions (G_2, G_3) as G_1 was optimized in the previous call as shown in Figure(2-b).

B. GUESS-AIDED INITIALIZATION

As the fine-tuning problems start with an initial solution that needs to be optimized, so to give GAMOPSO this ability a guess-aided initialization mechanism was proposed. Algorithm 2 shows the initialization process based on the initial guess, where a portion of the particles equivalent to the required ratio (InitPercent) is generated around the initial guess within specified standard deviation (STD) using a Gaussian distribution. After the required number of particles around the initial guess, the remaining particles are randomly generated along with the search space. The goal of initial guess is to help the algorithm to converge quickly as soon as possible; To understand the idea, let us analyze the performance of the algorithm in all possible scenarios related to the quality of initial guess; actually, there are two scenarios:

- For the first one, we have an accurate guess, so part of the initial solution will be close to the optimal solution; thus, the convergence of the algorithm will be faster than the random initialization case.
- The second scenario is to have a bad guess, but since we have only a small portion of the solution initiated based on the guess and the greatest portion generated randomly throughout the space, in this case, the performance of GAMOPSO will be approximately equivalent to the performance of traditional MOPSO.

Each particle is tested in case the boundaries of the search space are exceeded; the values are fixed and made within the search space. The final step is to calculate the values of the objective functions of the particles. Figures (2) and (3) illustrate the differences between the random and Guess-aided initialization processes.

Algorithm 2 Guess Aided Initialization

```
Input:
(1)InitGuess: the initial guess.
(2)numberOfSolutions: the number of solutions.
(3)InitPercent: the percent of initial solutions.
(4)STD: the standard deviation of the Gaussian distribution.
Output:
(1)Swarm: the initialized swarm.
1: Start Algorithm
2: IGParticlesNum round(numberOfSolutions _ InitP ercent)
3: for i=1:numberOfSolutions do
4:    if i<= IGParticlesNum then
5:       Swarm(i).position= InitGuess+GausianRand(STD)
6:    else
7:       Swarm(i).position= LowerBound+Rand*[UpperBound LowerBound]
8:    end if
9:    Swarm(i).position = CheckBounds(Swarm(i).position)
10:   Swarm(i).Cost = objectiveFunction(Swarm(i).position)
11: end for
12: End Algorithm
```

C. ANGLE QUANTIZATION

The concept of angle quantization was proposed by [9] to improve the exploration process in the NSGA-II algorithm. Figure (4) illustrates the concept of angle quantization, as we can see that the algorithm has only selected the solutions with red color because they maintain equal distribution concerning

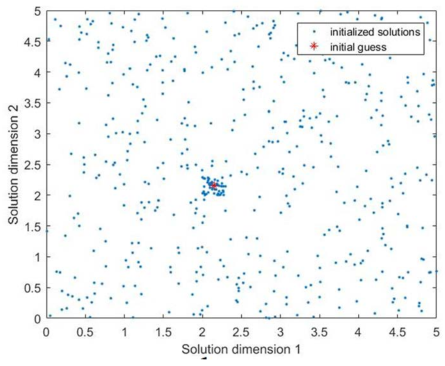

FIGURE 3. The randomly initialized swarm solutions were generated randomly without taking the initial guess into account.

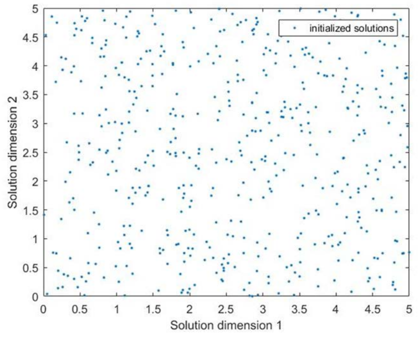

FIGURE 4. In the guess-aided initialized swarm, part of the solutions was generated around the initial guess.

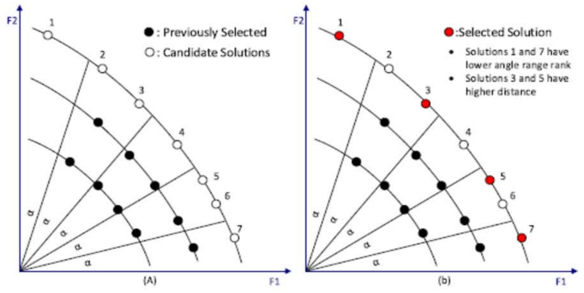
●: Previously Selected
○: Candidate Solutions
●: Selected Solution
Solutions 1 and 7 have lower angle range rank
Solutions 3 and 5 have higher distance
F1
F2
(A)
(b)

FIGURE 5. The concept of angle quantization-based selection, we chose the solutions from the less explored angular sector. [9].

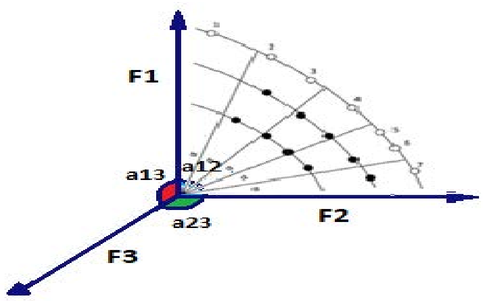

FIGURE 6. The extended angle quantization concept, where a12 is the angle after projecting the values of the objective functions on the plane specified by the objective functions F1 and F2 and the same for a13 on the plane specified by the objective functions F1 and F3 and so on.

angular sectors in addition to domination. So, the idea is to divide the objective space into equal angular sectors and calculate the number of solutions in each sector, then new solutions are chosen from the less explored sectors, which helps in a more balanced distribution of solutions over the whole searching space.

The algorithm will begin by sorting the solutions into non-dominated ranks. Next, each non-dominated rank is chosen one at a time to build the next generation, starting from the first rank, until reaching a rank at which the number of solutions exceeds the required number to complete the new swarm. In this stage, the algorithm will select the solutions that have a lower crowding angle range rank from the last rank. As a contribution of this research, the concept of angle quantization was adapted to the GAMOPSO algorithm and extended to be able to deal with more than two objective functions. To illustrate the idea, simply map the values of the objective functions on a plane specified by two objective functions each time, so that an independent ***crowding*** angle is generated between every two axes of the objective functions, as shown in Figure (5), where a12 is the angle after projecting the values of the objective functions on the plane specified by the objective functions F1 and F2 and the same for a13 on the plane specified by the objective functions F1 and F3, etc. So, in general, for problems with N objective functions, we need CA crowding angles as follows:

$$\mathbf{CA} = {}^{\mathbf{N}}_{\mathbf{2}}\mathbf{C} = \frac{\mathbf{N!}}{(\mathbf{N}-\mathbf{2})\,\mathbf{2!}} \tag{10}$$

A probabilistic method was used to choose the plane to be explored each time we have to select from the same rank solutions; simply we chose one of the crowding angles based on uniform distribution with probability equals:

$$\boldsymbol{P} = \frac{\mathbf{1}}{\boldsymbol{CA}} \tag{11}$$

Using uniform distribution guarantees equal chances for exploration on each objective axis, thus improving exploration of the whole space.

D. SELECTING THE HYBRID LEADER

The traditional MOPSO selects the particle leader based on the roulette wheel, where a solution is chosen randomly from the repository, but this mechanism does not guarantee rapid convergence of the algorithm, especially in complex applications such as bundle adjustment, so we developed a hybrid mechanism that helps to boost convergence by integrating the RADAM algorithm [10], which is one of the latest and best gradient algorithms, so we choose one of the superior solutions in the previous cycles and apply gradient for a small number of iterations to improve its quality, and then use it as a leader for a proportion of swarm particles and thus deploy the effect of this solution, which in turn may help in discovering better solutions as the remaining particles move towards the leader.

Basically, the third constraint -which was illustrated previously in Equation (3) contains elements from the first and second constraints; thus, it could be a good absolute

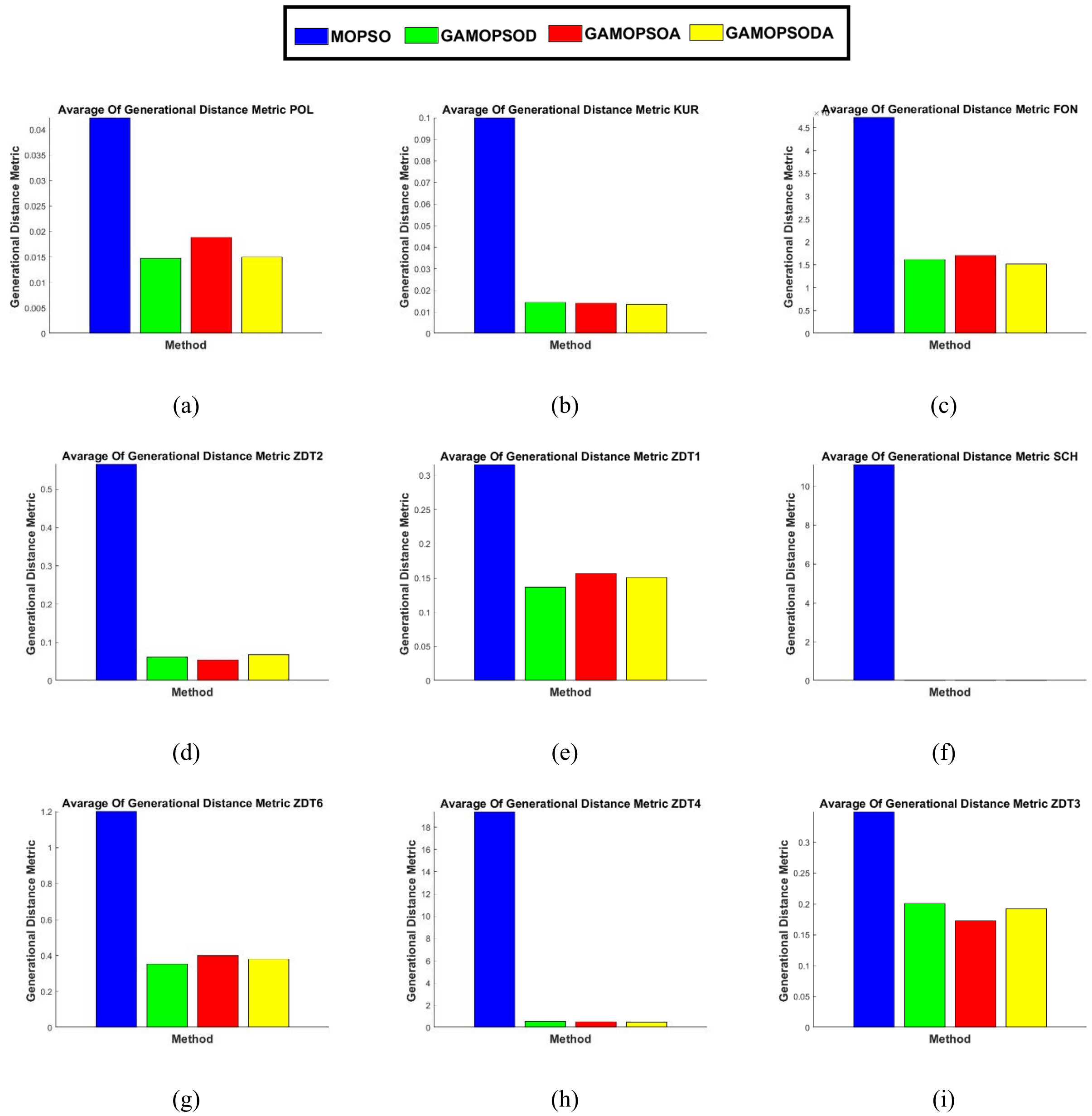

FIGURE 7. Results of the generational distance metric: The proposed algorithm variants outperform the standard MOPSO, which means that the optimal solutions found by the GAMOPSO variants are closer to the ideal solutions. (a) GD metric POL. (b) GD metric KUR. (c) GD metric FON. (d) GD metric ZDT2. (e) GD metric ZDT1. (f) GD metric SCH. (g) GD metric ZDT6. (h) GD metric ZDT4. (i) GD metric ZDT3.

indicator of the solution quality and used as a selection criterion for the superior solution. After selecting the superior solution, RADAM is used to apply a gradient on the second constraint - because the first constraint will be constant, as we illustrated in Figure (2-b), using its derivative in Equation (12) to compute the loss:

$$\begin{aligned} g'_{2v}(x_l,x_m) &= q'_l \cdot (t_{l\to m} \times q_m) \\ &\quad + q_l \cdot \left(t'_{l\to m} \times q_m\right) + q_l \cdot \left(t_{l\to m} \times q'_m\right) \end{aligned} \tag{12}$$

where:

q'_l : is the derivative of q_l with respect to camera position and orientation.

$t'_{l\to m}$: is the derivative of the transformation from frame l to frame m with respect to camera position and orientation.

q'_m : is the derivative of q_m with respect to camera position and orientation.

We used the epsilon greedy method to balance between the random behavior of the algorithm (using a roulette wheel to choose the leader) and the deterministic behavior (using an improved solution as a leader) as follows:

$$leader = \begin{cases} roulette\ \ wheel & r < \epsilon \\ improved\ \ solution & r \geq \epsilon \end{cases} \tag{13}$$

TABLE 1. The standard test functions used in experiments.

Function name	Optimal solutions	type	Variable range	Objective functions	Number of variables
FON: Fonseca–Fleming function	$x_1 = x_2 = x_3$	Non-convex	[-4, 4]	$f_1(x) = 1 - \exp(-\sum_{i=1}^{3}(x_i - \frac{1}{\sqrt{3}})^2)$ $f_2(x) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right)$	3
KUR: Kursawe function	[22]	Non-convex	[-5, 5]	$f_1(x) = \sum_{i=1}^{n-1}(-10\, exp\ (-0.2\sqrt{x_i^2 + x_{i+1}^2}))$ $f_2(x) = \sum_{i=1}^{n}(\lvert x\rvert^{0.8} + 5\sin x_i^3)$	3
POL: Poloni's two objective function	[14]	Non-convex	$[-\pi\ \pi]]$	$f_1(x) = 1 + (a1 - b1)^2 + (a2 - b2)^2$ $f_2(x) = (x(1) + 3)^2 + (x(2) + 1)^2$ $a1 = 0.5sin(1) - 2cos(1) + sin(2) - 1.5cos(2);$ $a2 = 1.5sin(1) - cos(1) + 2sin(2) - 0.5cos(2)$ $b1 = 0.5sin(x(1)) - 2cos(x(1)) + sin(x(2)) - 1.5cos(x(2))$ $b2 = 1.5\sin(x(1)) - \cos(x(1)) + 2\sin(x(2)) - 0.5\cos(x(2))$	2
SCH: Schaffer function N. 1	$f2 = (\sqrt(f1) - 2)^2$	Convex	$-a \leq x \leq a$ $a \in [10\ 10^5]$	$f_{1(x)} = x^2$ $F_2(x) = (x - 2)^2$	1
ZDT1: Zitzler-Deb-Thiele function N. 1	$x_1 \in [0, 1]$ $x_i = 0$ i = 2,3, . . . , n	Convex	[0, 1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right]$ $g(x) = 1 + 9\,(\sum_{i=2}^{n} x_i)\,/(n-1)$	30
ZDT2: Zitzler-Deb-Thiele function N. 2	$x_1 \in [0, 1]$ $x_i = 0$ i= 2,3, . . . , n	Non-convex	[0, 1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - (\frac{x_1}{g_{(x)}})^2\right]$ $g(x) = 1 + 9\,(\sum_{i=2}^{n} x_i)\,/(n-1)$	30
ZDT3: Zitzler-Deb-Thiele function N. 3	$x_1 \in [0, 1]$ $x_i = 0$ i = 2,3,. . , n	Convex	[0, 1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}\sin(10\,\pi\,x_1)\right]$ $g(x) = 1 + 9\,(\sum_{i=2}^{n} x_i)\,/(n-1)$	30
ZDT4: Zitzler-Deb-Thiele function N. 4	Many local optimum Pareto fronts	Convex	$X_1[0,1]$ $X_i[-5,5]$	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x)g(x))$ $g(x) = 91 + \sum_{i=2}^{10}(x_i^2 - 10\cos(4\,\pi x_i))$ $h(f_1(x)g(x)) = 1 - \frac{\sqrt{f_1(x)}}{g(x)}$	[2,10]
ZDT6: Zitzler-Deb-Thiele function N. 6	$x_1 \in [0, 1]$ $x_i = 0$ i= 2,3, . . . , n	Non-convex	[0, 1]	$f_1(x) = 1 - \exp(-4x_1)sin^6(6\pi x_1)$ $f_2(x) = g(x)\left[1 - (\frac{f_{1(x)}}{g_{(x)}})^2\right]$ $g(x) = 1 + 9\left[\frac{\sum_{i=2}^{10} x_i}{9}\right]^{0.25}$ $h(f_1(x), g(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$	10

IV. EXPERIMENT SETS

A. STANDARD MULTI-OBJECTIVE EXPERIMENTS DESCRIPTION

To investigate the effect of each suggested contribution, 3 variants of the GAMOPSO algorithm were implemented:

- **GAMOPSOD**: which uses the crowding distance only to choose from the same ranked particles.
- **GAMOPSOA**: which uses angle quantization only to choose from the same-ranked particles.
- **GAMOPSODA**: which uses probability to select either crowding distance or angle quantization to choose from the same ranked particles.

The three variants were compared with the MOPSO implementation suggested by [11].

A set of standard mathematical functions suggested for multi-objective optimization problems illustrated in Table (1) was used to evaluate the performance of algorithms. These functions have been used in many previous studies in this field. Veldhuizen [12] used a set of these functions and two of them were selected (FON) [13], and (KUR) in our case, in addition to (POL) [14] and (SCH). In 1999 Deb [15] proposed methodological approaches for developing test functions of multi-objective optimization algorithms, Zitzler and his colleagues [16] followed this methodology and proposed six optimization functions, five of them were used in this research ZDT1, ZDT2, ZDT3, ZDT4, ZDT6.

Three standard MOO metrics were used to compare the performance:

1) GENERATIONAL DISTANCE METRIC (GD)

This metric is given by the following formula:

$$GD(S, P) = \frac{1}{|S|}\left(\sum_{s\in S} \min_{r\in p} \parallel F(s) - F(r) \parallel^{p}\right)^{\frac{1}{p}} \quad (14)$$

where |S| is the number of points in the Pareto set approximation and P is a discrete representation of the Pareto front. Generally, $p = 2$. In this case, it is the Euclidean distance between the Pareto set approximation and the true Pareto. [17]

2) HYPER-VOLUME METRIC

Sometimes called an S-metric, the hypervolume is described as the volume of the space in the objective space dominated by the Pareto front approximation S and delimited from above by a reference point $r \in \mathbb{R}^m$ such that for all $z \in S$, $z \prec r$. The hypervolume metric is given by:

$$\mathbf{HV}(\mathbf{S}, \mathbf{r}) = \lambda_{\mathbf{m}} \left(\bigcup_{\mathbf{z} \in \mathbf{S}} [\mathbf{z}; \mathbf{r}] \right) \tag{15}$$

where λ_m is the m-dimensional Lebesgue measure[18]

3) COVERAGE OF TWO SETS METRIC (C)

Let A and B be two Pareto set approximations. The C-metric maps the ordered pair (A, B) to the interval [0; 1] and is defined as follows:

$$\mathbf{C}(\mathbf{A}, \mathbf{B}) = \frac{|\ \{\mathbf{b} \in \mathbf{B}, \text{ there exists } \mathbf{a} \in \mathbf{A} \text{ such that } \mathbf{a} \prec \mathbf{b}\}\ |}{|\mathbf{B}|} \tag{16}$$

If C(A,B) = 1, all elements of B are dominated by (or equal to) the elements of A. If C(A,B) = 0, all elements of B strictly dominate the elements of set A. Both orderings must be computed since C (A, B) is not always equal to 1 -C (A, B). This metric captures the proportion of points in a Pareto set approximation A dominated by the Pareto set approximation B [18]

The experiments were conducted 10 times to verify the performance of the algorithms, each time the seed of the random number generator was changed to ensure that the series of generated random numbers did not affect the results obtained by averaging the measurements of the ten experiments.

B. BUNDLE ADJUSTMENT EXPERIMENTS

To evaluate the performance of the proposed algorithm, bundle adjustment was applied to two artificial trajectories generated based on the simulation of camera movement and the projection of 3D points with noise to model the projection error of the camera projection error [19]. The first trajectory consists of 200 points distributed over 20 overlapping frames, and the second consists of 200 points distributed over 50 overlapping frames. GAMOPSO was compared to the traditional Levenberg-Marquardt-based bundle adjustment algorithm [20] and the finite-deference approximation-based bundle adjustment [21].

The experiments were repeated five times using a different seed for the random number generator to statistically test the performance of the algorithm.

V. RESULTS AND DISCUSSIONS

A. STANDARD MULTI-OBJECTIVE RESULTS

1) GENERATIONAL DISTANCE

As can be seen in Figure (7), there is a significant difference in the generational distance between the GAMOPSO variants and the standard MOPSO, and since the less, the better, the proposed variants outperform the standard MOPSO in this metric. This means that the optimal solutions found by the GAMOPSO variants are closer to the ideal solutions. Also, sometimes GAMOPSOA outperforms GAMOPSOD, so in some cases, angle quantization helped to improve the diversity of solutions more than the crowding distance, which helped the algorithm find more appropriate solutions.

2) HYPER-VOLUME METRIC

Figure (8) shows the average hypervolume metric, which provides a qualitative measure of convergence and solution sparsity at the same time. The more the merrier, the results show that the proposed algorithms are superior to the MOPSO algorithm in the hypervolume metric; therefore, the quality of the solutions is better and their spread is more balanced, as it covers a larger area of the search space than the benchmark algorithm, which confirms the superiority of the proposed algorithm in all its variants over the benchmark algorithm in terms of hypervolume.

3) SET COVERAGE METRIC

Figure (9) shows the two-set average coverage metric, which calculates the percentage of solutions in a set A dominated by at least one of the solutions of another set B. The results show that the coverage ratio of the solutions found by the GAMOPSO algorithm ranged from 60% to 100%, compared to a low opposite coverage that ranged between 0 and 5%, indicating that the quality of the solutions obtained from the proposed variants is better than the quality of the solutions obtained by the benchmark, which confirms the absolute superiority of the proposed algorithms over the benchmark MOPSO algorithm in this metric.

B. BUNDLE ADJUSTMENT RESULTS

1) FIRST TRAJECTORY RESULTS

A camera trajectory was generated over 20 frames that detected 200 distinct points with large rotation steps between frames (6 degrees), as shown in Figure (10). A comparison was made between the proposed algorithm and the two benchmark algorithms (Levenberg-Marquardt and finite-deference approximation). The results showed an obvious superiority of the proposed algorithm in terms of accuracy and execution time. Figure (11) shows the box plot of the results of the first trajectory, where the red line expresses the mean error value for each algorithm during the experiment, while the blue rectangle represents the range of values within which the error ranged during the experiment. It is worth noting that the error values in the five experiments for each of the two benchmark algorithms are fixed as they are

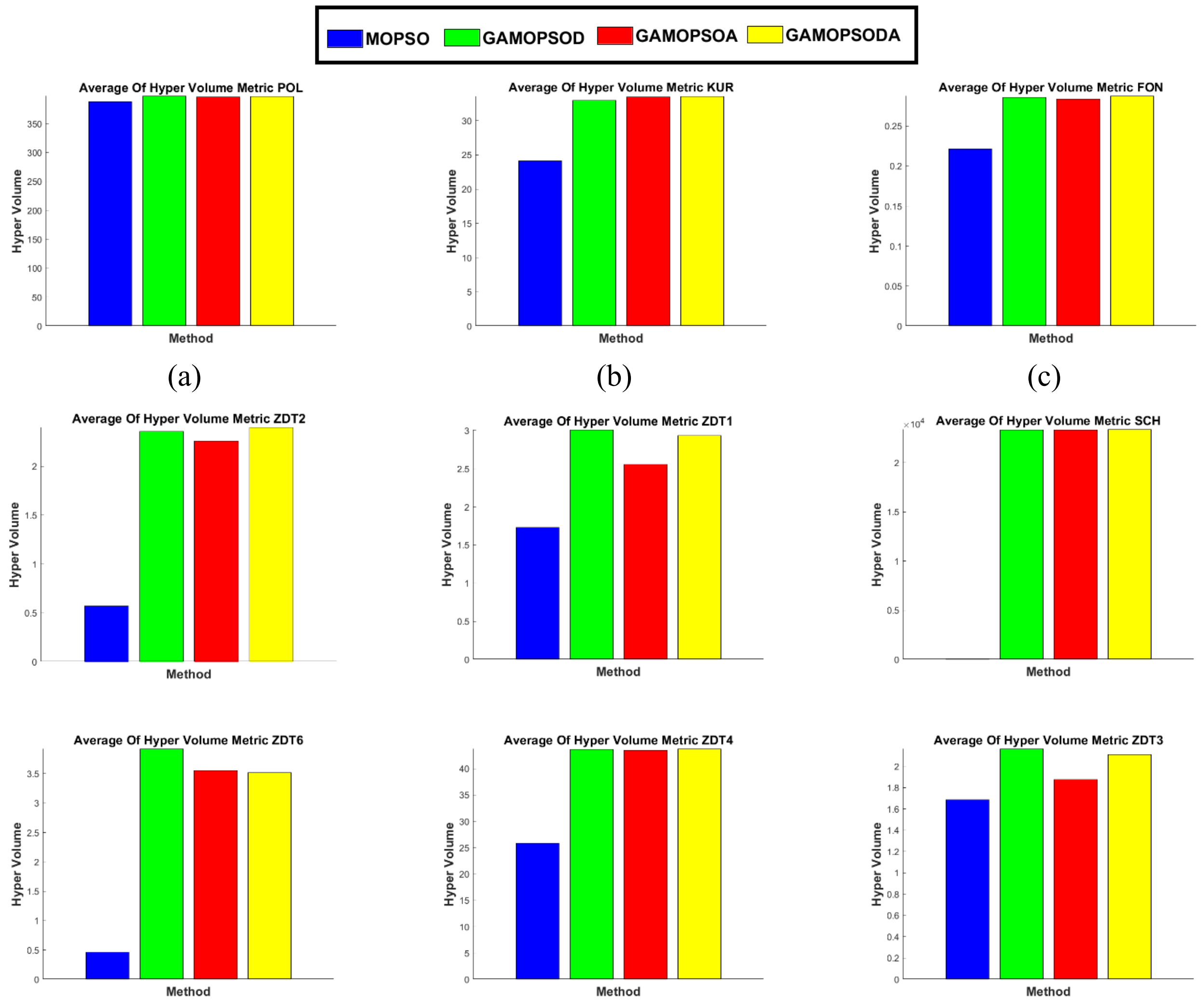

FIGURE 8. Results of the hypervolume metric: The variants of the proposed algorithm are superior to the MOPSO algorithm in the hypervolume metric, as the volume occupied by their solutions (shown in columns 2, 3, and 4 in each subfigure) is greater than the volume occupied by the solutions of the benchmark algorithm. (a) HV metric POL. (b) HV metric KUR. (c) HV metric FON. (d) HV metric ZDT2. (e) HV metric ZDT1. (f) HV metric SCH. (g) HV metric ZDT6. (h) HV metric ZDT4. (i) HV metric ZDT3.

deterministic, while the proposed algorithm uses the random search technique, so there is a discrepancy in the error values according to the seed, despite that, the highest value of the error for GAMOPSO is less by 5% from the minimum error value of the benchmark algorithms.

Figure (12) shows the execution time of the algorithms on the first trajectory, the proposed algorithm is faster and consumes approximately 50% of the time consumed by the Levenberg algorithm and approximately 60% of the time of Finite Deference.

2) SECOND TRAJECTORY RESULTS

A camera trajectory was generated over 50 frames that detected 200 distinct points with smaller rotation steps between frames (3 degrees), as shown in Figure (13). A comparison was made between the proposed algorithm and the two benchmark algorithms (Levenberg-Marquardt and finite-deference approximation). The results showed an obvious superiority of the proposed algorithm in terms of accuracy and execution time. Figure (14) shows the box plot of the results of the second trajectory, where the red line expresses the mean error value for each algorithm during the experiment, while the blue rectangle represents the range of values within which the error ranged during the experiment. It is worth noting that the error values in the five experiments for each of the two benchmark algorithms are fixed as they are deterministic, while the proposed algorithm uses the random search technique, so there is a discrepancy in the error values according to the seed, despite that, the highest value of the error for GAMOPSO is less by 20% from the minimum error value of the benchmark algorithms.

Figure (15) shows the execution time of the algorithms on the second trajectory, the proposed algorithm is faster and consumes approximately 56% of the time consumed by the Levenberg algorithm and approximately 58% of the time of finite differences.

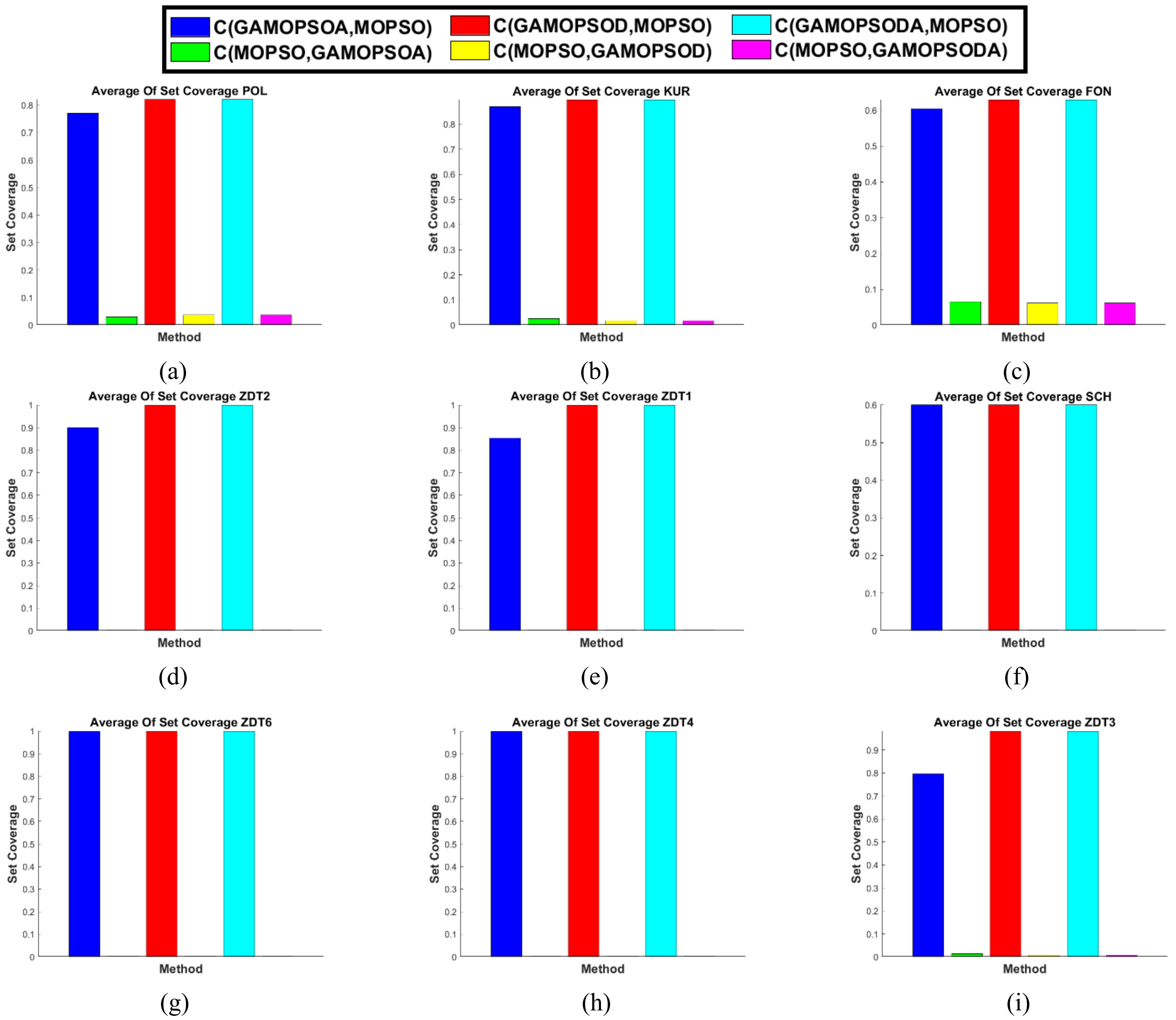

FIGURE 9. Set coverage metric results: the coverage ratio of the solutions from the variants of the GAMOPSO algorithm ranged from 60% to 100%, compared to a low ratio of the opposite coverage ranging between 0 and 5%, which indicates that the quality of the solutions obtained from the proposed variants is better than the quality of solutions obtained by the benchmark. (a) Set Coverage POL. (b) Set Coverage KUR. (c) Set Coverage FON. (d) Set Coverage ZDT2. (e) Set Coverage ZDT1. (f) Set Coverage SCH. (g) Set Coverage ZDT6. (h) Set Coverage ZDT4. (i) Set Coverage ZDT3.

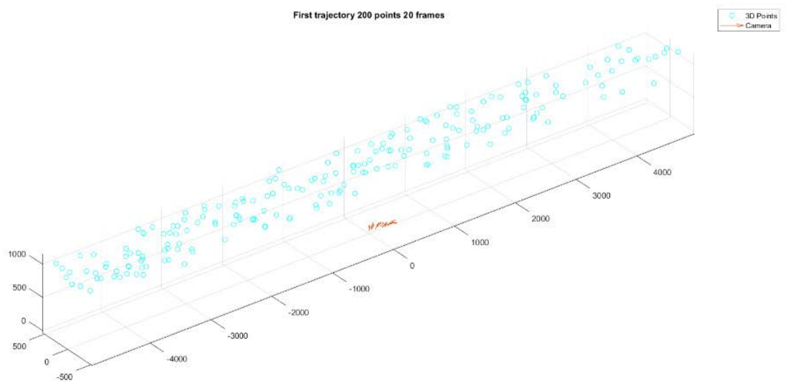

FIGURE 10. The first trajectory of the camera has 200 3D points, 20 frames, and a 6 degree angular step.

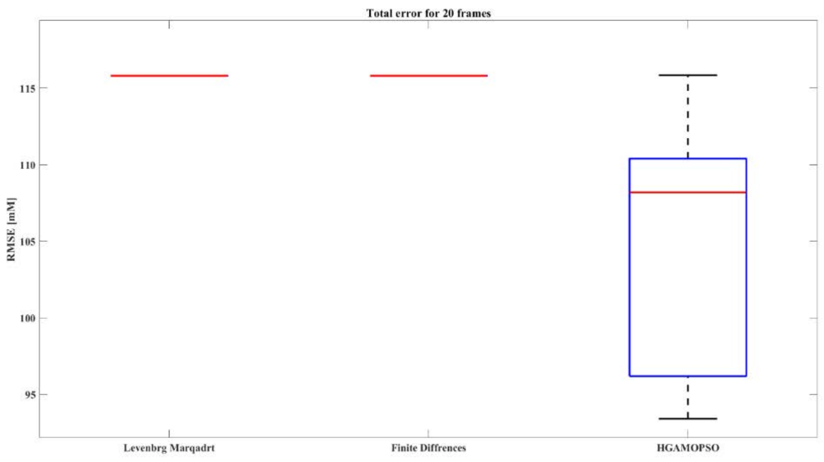

FIGURE 11. The RMSE value for the entire trajectory as a boxplot: The highest value of the error for GAMOPSO is less than 5% of the minimum error value of the benchmark algorithms.

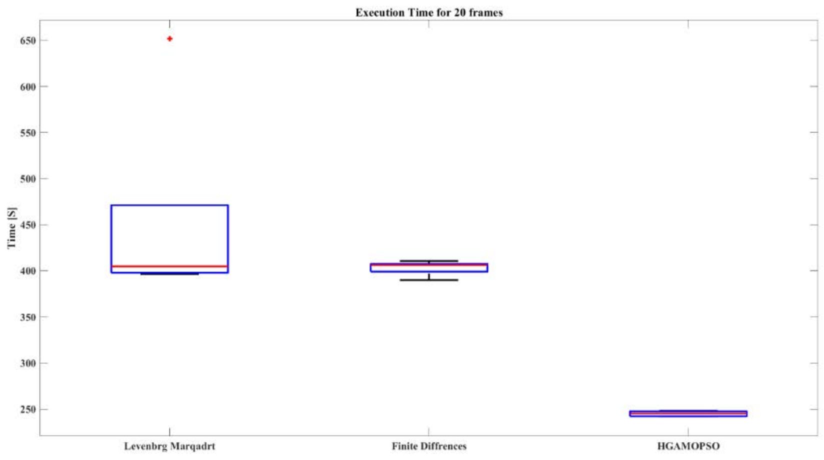

FIGURE 12. The total execution time of algorithms as boxplot: the proposed algorithm is faster and consumes approximately 50% of the time consumed by the Levenberg algorithm and approximately 60% of the time of finite deference.

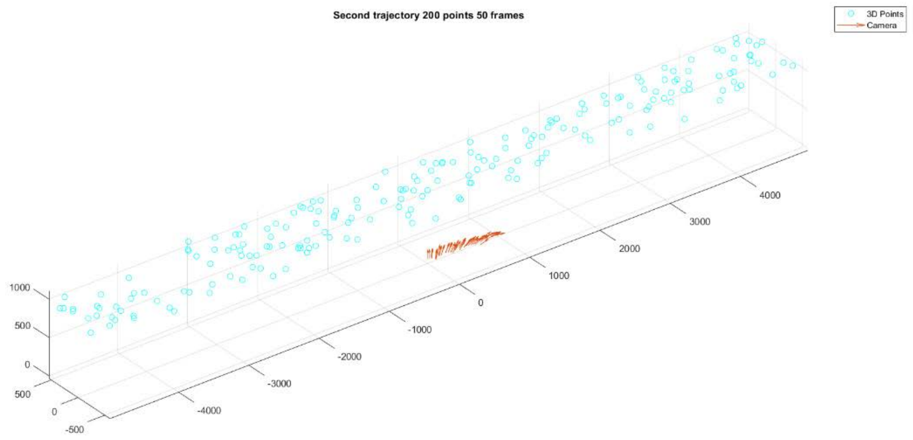

FIGURE 13. The Second trajectory of the camera is 200 3D points, 50 frames, and 3 degrees angular step.

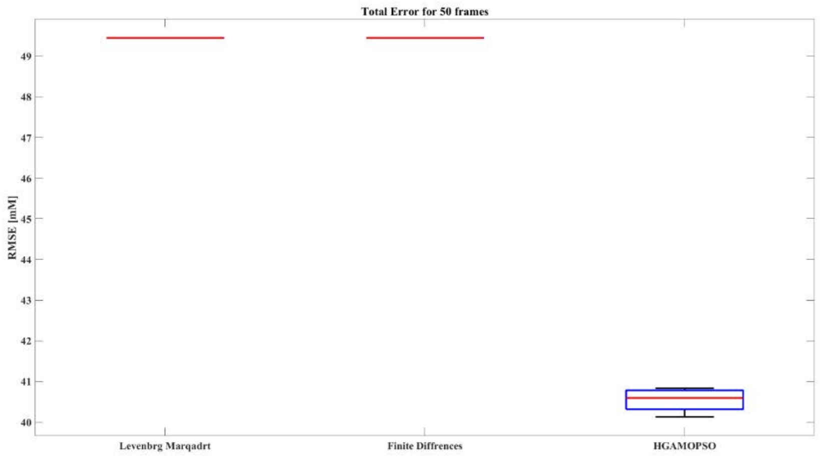

FIGURE 14. The RMSE value for the entire trajectory as a boxplot: The highest error value for GAMOPSO is less than 20% of the minimum error value of the benchmark algorithms.

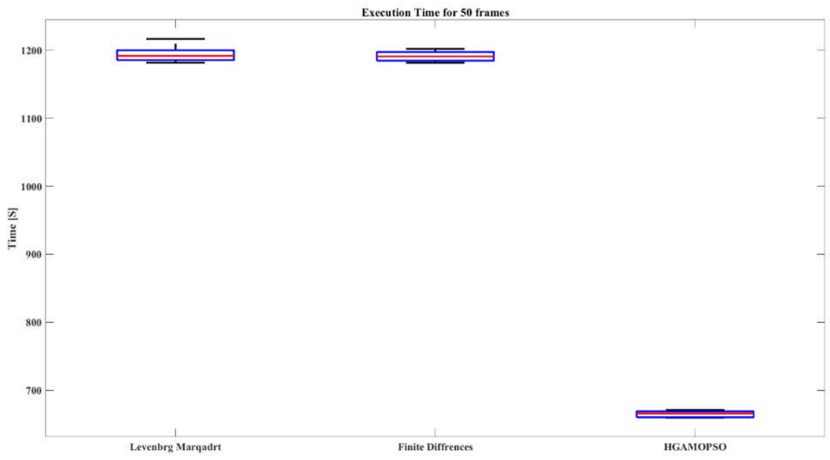

FIGURE 15. The total execution time of algorithms as boxplot: the proposed algorithm is faster and consumes approximately 56% of the time consumed by the levenberg algorithm and approximately 58% of the time of finite deference.

VI. CONCLUSION

Bundle adjustment plays a vital role in 3D reconstruction processes. Therefore, improving the performance of bundle adjustment can be useful in a variety of applications, including Visual Simultaneously Localization and Mapping (VSLAM), Structure from Motion (SfM), Virtual Reality (VR), and others. So future research could examine the use of this algorithm in real 3D reconstruction applications like Visual Simultaneous Localization and Mapping and Structure from Motion. Or it might apply the same techniques proposed in this paper for GAMOPSO like guess-aided initialization, and hybridization with gradient descent algorithms to improve other evolutionary algorithms like (NSGAII, NSGAII, etc.).

In this paper, we proposed a novel bundle adjustment algorithm based on a developed version of MOPSO. The proposed algorithm used initial guess, angle quantization technique, and RADAM optimizer to improve the mobility and quality of the solutions, the results showed that our algorithm can help improve the accuracy and the efficiency of BA.

REFERENCES

[1] A. L. Jaimes and S. Z. Martínez, *An Introduction to Multi Objective Optimization Techniques*. Hauppauge, NY, USA: Nova Science, 2011.

[2] S. Mostaghim and J. Teich, ''Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO),'' in *Proc. IEEE Swarm Intell. Symp. (SIS)*, Apr. 2003, pp. 26–33, doi: 10.1109/SIS.2003.1202243.

[3] K. Konolige, ''Sparse sparse bundle adjustment,'' in *Proc. Brit. Mach. Vis. Conf.* BMVA Press, 2010, pp. 102.1–102.11, doi: 10.5244/C.24.102.

[4] X. Li and S. W. Li, ''Obtaining approximate values of exterior orientation elements of multi-intersection images using particle swarm optimization,'' *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 39, pp. 35–40, Jul. 2012.

[5] C. Zhang, G. Wen, C. Wu, H. Wang, Z. Shang, and Q. Zhang, ''Genetic algorithm for bundle adjustment in aerial panoramic stitching,'' *Proc. SPIE*, vol. 9443, p. 94431, Mar. 2015, doi: 10.1117/12.2178852.

[6] W. Li and K. Zhong, ''Application of improved particle swarm optimization algorithm in solving camera extrinsic parameters,'' *J. Modern Opt.*, vol. 66, no. 18, pp. 1827–1835, Oct. 2019, doi: 10.1080/09500340.2019.1682203.

[7] M. J. R. Aguiar, T. D. R. Alves, L. M. Honório, I. C. S. Junior, and V. F. Vidal, "Performance evaluation of bundle adjustment with population based optimization algorithms applied to panoramic image stitching," *Sensors*, vol. 21, no. 15, p. 5054, Jul. 2021, doi: 10.3390/s21155054.

[8] V. Indelman and F. Dellaert, "Incremental light bundle adjustment: Probabilistic analysis and application to robotic navigation," *Cogn. Syst. Monogr.*, vol. 23, pp. 111–136, Sep. 2014, doi: 10.1007/978-3-662-43859-6_7.

[9] A. Metiaf, Q. Wu, and Y. Aljeroudi, "Searching with direction awareness: Multi-objective genetic algorithm based on angle quantization and crowding distance MOGA-AQCD," *IEEE Access*, vol. 7, pp. 10196–10207, 2019, doi: 10.1109/ACCESS.2018.2890461.

[10] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," 2019, *arXiv:1908.03265*.

[11] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004, doi: 10.1109/TEVC.2004.826067.

[12] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses and new innovations," M.S. thesis, Air Force Inst. Technol., Wright-Patterson AFB, OH, USA, 1999, doi: 10.5555/929368.

[13] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 28, no. 1, pp. 26–37, Jan. 1998, doi: 10.1109/3468.650319.

[14] C. Poloni, A. Giurgevich, L. Onesti, and V. Pediroda, "Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 403–420, Jun. 2000, doi: 10.1016/S0045-7825(99)00394-1.

[15] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems," *Evol. Comput.*, vol. 7, no. 3, pp. 205–230, Sep. 1999, doi: 10.1162/evco.1999.7.3.205.

[16] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, Jun. 2000, doi: 10.1162/106365600568202.

[17] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, L. Salomon, and S. Le. (2018). *Performance Indicators in Multi Objective Optimization*. Cah. du GERAD. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02464750

[18] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Berlin, Germany: Springer-Verlag, 1999.

[19] R. Giubilato. (2016). *Bundle Adjustment Using Lsqnonlin*. [Online]. Available: https://github.com/RiccardoGiubilato/Test_BA

[20] M. L. A. Lourakis and A. A. Argyros, "Is Levenberg–Marquardt the most efficient optimization algorithm for implementing bundle adjustment?" in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 1526–1531, doi: 10.1109/ICCV.2005.128.

[21] A. Chambolle and T. Pock, "Approximating the total variation with finite differences or finite elements," in *Handbook of Numerical Analysis: Geometric Partial Differential Equations II*, vol. 22. France: HAL Open Science, 2021, pp. 383–417, doi: 10.1016/bs.hna.2020.10.005.

[22] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Hoboken, NJ, USA: Wiley, 2001.

MAHER ALNDIWEE received the B.S. degree in computer and automation engineering and the M.S. degree in robot engineering and programming from Damascus University, Damascus, Syria, in 2012 and 2016 respectively, where he is currently pursuing the Ph.D. degree in robotics engineering and programming.

Since 2013, he has been working as a Teaching Assistant at the Faculty of Mechanical and Electrical Engineering, Damascus University. His research interests include computer vision, optimization, machine learning, deep learning, robotics, and AI algorithms.

Mr. Alndiwee's awards and honors include the Basel Al-Assad Awards for Academic Excellence, Top Student Award out of 80 students, from Damascus University, in 2008, 2011, and 2012, and the Silver Medal at Al-Basel Exhibition for creativity and invention, in 2017.

MOHAMED MAZEN AL-MAHAIRI received the B.Sc. degree in electronic engineering from Damascus University, Damascus, Syria, in 1986, and the Ph.D. degree in computer organization and architecture from Saint Petersburg Electrotechnical University "LETI", Russia, 1993.

He is an Associate Professor of the computer and automation engineering with the Faculty of Mechanical and Electrical Engineering, Damascus University. He is a member of the Distinction and Creativity Agency, the Syrian Computer Society, and Damascus University Journal. His research interests include multiprocessor systems, embedded systems, and design.

RAOUF HAMDAN received the B.Sc. degree in electronic engineering from Damascus University, Damascus, Syria, in 1990, the M.Sc. degree in photonique and image and the Ph.D. degree in computer vision and digital image processing from Louis Pasteur University, France, in 1996 and 2001, respectively.

He is an Associate Professor of the computer and automation engineering with the Faculty of Mechanical and Electrical Engineering, Damascus University. He is a member of the Syrian Computer Society and the Damascus University Journal. His research interests include computer vision, image sequence analysis, computer networks, network application programming, real-time systems, distributed and cloud computing, machine learning, and deep learning.