

Article

# Performance Evaluation of Bundle Adjustment with Population Based Optimization Algorithms Applied to Panoramic Image Stitching

Maria Júlia R. Aguiar , Tiago da Rocha Alves , Leonardo M. Honório \* , Ivo C. S. Junior   
and Vinícius F. Vidal 

Department of Electrical Engineering, Federal University of Juiz de Fora, Juiz de Fora 36036-900, Brazil; maria.aguiar@engenharia.ufjf.br (M.J.R.A.); tiago.alves@engenharia.ufjf.br (T.d.R.A.); ivo.junior@ufjf.edu.br (I.C.S.J.); vinicius.vidal@engenharia.ufjf.br (V.F.V.)

\* Correspondence: leonardo.honorio@ufjf.edu.br

**Abstract:** The image stitching process is based on the alignment and composition of multiple images that represent parts of a 3D scene. The automatic construction of panoramas from multiple digital images is a technique of great importance, finding applications in different areas such as remote sensing and inspection and maintenance in many work environments. In traditional automatic image stitching, image alignment is generally performed by the Levenberg–Marquardt numerical-based method. Although these traditional approaches only present minor flaws in the final reconstruction, the final result is not appropriate for industrial grade applications. To improve the final stitching quality, this work uses a RGBD robot capable of precise image positing. To optimize the final adjustment, this paper proposes the use of bio-inspired algorithms such as Bat Algorithm, Grey Wolf Optimizer, Arithmetic Optimization Algorithm, Salp Swarm Algorithm and Particle Swarm Optimization in order to verify the efficiency and competitiveness of metaheuristics against the classical Levenberg–Marquardt method. The obtained results showed that metaheuristics have found better solutions than the traditional approach.

**Keywords:** metaheuristics; bundle adjustment; panorama image; bat algorithm; grey wolf optimizer; arithmetic optimization algorithm; Salp Swarm Algorithm; particle swarm optimization



**Citation:** Aguiar, M.J.R.; Alves, T.d.R.; Honório, L.M.; Junior, I.C.S.; Vidal, V.F. Performance Evaluation of Bundle Adjustment with Population Based Optimization Algorithms Applied to Panoramic Image Stitching. *Sensors* **2021**, *21*, 5054. <https://doi.org/10.3390/s21155054>

Academic Editor: Junliang Xing

Received: 2 June 2021

Accepted: 19 July 2021

Published: 26 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Over the years, several computer vision simulation techniques and methods have been developed to extract information in order to understand the environment through the use of sensors and cameras. As computers and cameras have become more affordable, the use of digital images has grown, and thus generated an increasing interest in high-resolution images. The automatic panoramic image stitching, as an example, is an important field of study for the research community due to its use in several applications in computer vision [1], photogrammetry [2–4] and remote sensing [5–8].

Panoramic images are obtained to produce images with wide field of view (FOV) and depict large objects that cannot be captured in a single image. Therefore, multiple images of the same environment are combined to obtain the whole scene. Image stitching is the process of overlapping a set of images obtained from different points of view, time, visual angles and sensors to generate a high resolution image with a wider view [9,10].

Although seamless panoramic images the desired result for applications, and there are advances in this area of study in recent years, image stitching remains a challenge due to factors such as registration and blending [11]. Some questions remain unresolved and from a computational geometry perspective the result usually contains cut objects or blur [12]. The image stitching method is the joining of images based on their overlapping areas and can be defined in two main steps: alignment and blending [13].

In image registration process, several errors can arise when images are grouped. Geometric and photometric misalignment often result in undesirable discontinuities of objects and visibility in the overlap area between images [14]. However, perfect alignment is rarely achieved. In that regard, most efforts in this research field are focused on designing better alignment or compositing techniques to reduce or hide misalignment. Some algorithms are used in order to minimize the coloration discontinuities in the appearance of the final generated image, being some techniques of pixel matching and image blending [15–18]. The research on images alignment for stitching culminated, to some extent, in the use of bundle adjustment [19] to simultaneously optimize the relative positions of the images. Several types and classes of optimization-based algorithms [20–26] can be used to align all images to a common frame of reference typically, the classic Levenberg–Marquardt method is used to optimize the intrinsic and extrinsic parameters of the camera [27,28].

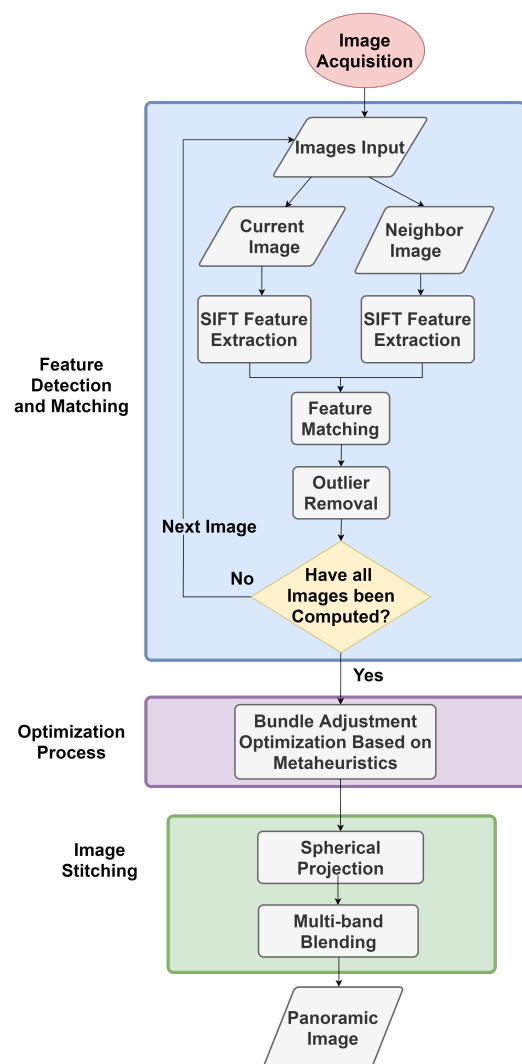
This paper proposes the comparison of the Levenberg–Maquardt algorithm, which is commonly used in the literature, against some meta heuristics that are well proven to be a good solution to multimodal optimization problems, that tend to be affected by local minima [29]. Thus, it aims to experiment a modified optimization approach for the Bundle Adjustment of a custom 360-degree image capture device. Therefore, this research’s main contribution is the comparison of the Levenberg–Marquardt method against bio-inspired algorithms, such as Bat Algorithm, Grey Wolf Optimizer, Salp Swarm Algorithm, Arithmetic Optimization Algorithm and Particle Swarm Optimization in Bundle adjustment optimization of the intrinsic and extrinsic parameters of cameras, focusing on improving the system’s image alignment stage. The remainder of this work is organized as follows: Section 2 details the proposed methodology and its mathematical and construction foundations. Section 3 presents the results and discussions about what was proposed and improved when using bio-inspired meta heuristics. The final concluding remarks and ideas for future works are presented in Section 4.

## 2. Proposed Study Methodology

This paper proposes a comparison of optimization methods for image alignment using bundle adjustment to generate spherical panoramic images. This section presents the overview of the proposed pipeline, followed by the main components of the methodology.

### 2.1. Framework Overview

Figure 1 shows the general architecture of the proposed method. The panoramic image generation pipeline is composed of several steps including image acquisition, feature extraction and matching, bundle adjustment, spherical projection, image stitching and blending. The first step is to capture images using a proprietary robotic system consisting of a camera and laser on a mechanical structure capable of rotation in pan and tilt directions under controlled conditions, which are used as input data for the system. Feature points are extracted from the images and combined with each other, resulting in a set of matched pairs. In the next step, the pairs are used to obtain the intrinsic and extrinsic parameters of the images by bundle adjustment, in which the re-projection error of the correspondences is minimized using optimization. Finally, a spherical projection is performed, as each image is projected onto a spherical surface and all image are stitched and blended together to obtain the final panorama. The components of the methodology are described in separated sections with detailed explanation.



**Figure 1.** Flowchart of the Proposed Method.

## 2.2. Data Acquisition

Image acquisition is performed using a robotic system illustrated in Figure 2. The robot is a scanning system capable of obtaining the reconstruction of an entire 3D environment or object that can be used in augmented and virtual reality applications. This concept inserts a camera and a laser into a mechanical structure that is capable of rotating under controlled conditions. It is composed of a Livox MID-40 LIDAR laser scanner, a Logitech C925e USB camera, two Dynamixel AX-18A servo motors and an InvenSense Inertial Measurement Unit (IMU) MPU-6050. The control strategy is a fault tolerant nonlinear feedback linearization [30–32]. This helps with image reconstruction, as the acquisition poses are previously known using the initial orientations from the IMU and servo angles as references in the Bundle Adjustment process. The angle positioning and intrinsic parameter information of each camera are included in the respective datasets. The acquired images are available in [33].

This module was used to acquire several high-resolution images for assembling 360° panoramas, using distance and positioning data from IMU and servo motors angles. This would be essential to ensure that the scene of interest is effectively recorded.

The servo motors are responsible for pan and tilt movements over a range of 360 and 120 degrees, respectively. The images were taken using a 50° pan and 15° tilt step to move the servomotor, covering the 360° region. In this way, sets of 56 images from the Arts and Design School at Federal University of Juiz de Fora were acquired. For this work, 1280 × 720 pixels HD image sets were used.



**Figure 2.** Data Acquisition System.

### 2.3. Feature Detection and Matching

Image alignment consists of overlapping two or more images of the same scene taken at different times, establishing geometric correspondences between pairs of images that represent the same environment [34]. In order, to register multiple images, it is necessary to estimate the transformations that align them according to an image reference within the data set. Therefore, the images are aligned to a common coordinate using the computed geometric transformation. The aligned images are overlaid on a larger image by merging the pixel values of the overlap area so that the border is continuous with a smooth transition between images.

The first step in the alignment process is to extract and find correspondences between images. The Scale Invariant Feature Transform (SIFT) algorithm developed by Lowe [35] was used for this method and is widely used in image stitching [17,36–38]. After calculating the descriptors for all images, points are found in the images that have similar descriptors. For each feature found, the two best candidate matches (nearest matches) were maintained. To improve the match set, the ratio test is first applied according to (1) in which  $d_1$  is the nearest neighbor distance,  $d_2$  is the second nearest neighbor distance and the ratio value less than a threshold ( $\theta$ ) set to 0.8 is considered a good match. After this condition, unwanted matches are discarded. Even so, in the resulting correspondences there can still be found outliers. Therefore, the good matches are filtered by calculating the fundamental matrix, capable of mapping points from one image to another with the Random Sample Consensus Algorithm (RANSAC) [39].

$$\frac{d_1}{d_2} \leq \theta \quad (1)$$

### 2.4. Optimization Process

After obtaining the set of geometrically consistent matches between images, bundle adjustment is used to solve all camera parameters together. The initial intrinsic and extrinsic parameters for each camera are known. The extrinsic parameters are derived from servo motors positions and the IMU during image acquisition obtaining rotation angles of each image. The focal length and optical centers were previously found by camera calibration. From this structured acquisition information it was possible to find the best neighbors of each image and perform image-matching with the best neighbors. In this step, the optimization process begins. The images are added with their correspondent neighborhood, and the bundle adjustment process is initialized with the values of neighbor's matches, focal length, optical center and pan and tilt angles for each camera. Then the parameters are update using Bio-inspired methods.

Bio-inspired techniques were utilized in order to verify which method finds the best result for the Bundle Adjustment problem, and compared with the classic Levenberg-Maquardt. Population based metaheuristics were used and some solutions are modified at each iteration, while others reach the next iteration. The modifications of the solu-

tion, generally, are made by specific properties of each algorithm and its populations [40]. For that, optimization algorithms as Bat Algorithm, Grey Wolf Optimizer, Salp Swarm Algorithm, Arithmetic Optimization Algorithm and Particle Swarm Optimization were used.

#### 2.4.1. Mathematical Formulation

To solve the camera positioning problem, the approach of relating images according to the homography matrix between images pairs to project them onto a reference plane was used [17]. The homography matrix that represents the pairs between the  $i$ -th and  $j$ -th images is given by Equation (2).

$$H_{ij} = K_i R_i R_j^T K_j^{-1} \quad (2)$$

where

$$K_i = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

represents the intrinsic matrix,  $f_x, f_y$  are the focal length and  $c_x, c_y$  the optical centers of the camera. The rotation matrix is given by:  $R_i = R_{\theta_i} R_{\phi_i}$  where  $\theta$  and  $\phi$  are the pan and tilt angles, respectively.

Each feature is projected onto all the images it corresponds to, and the sum of the squared distances of images is minimized in relation to the camera parameters. For each image, the parameters representing pan and tilt can be optimized [41], as well as the values of camera's focal length and optical center. Equation (4) represents the projection residue of  $k$ -th feature in one image corresponding to the  $m$ -th feature in another image.

$$r_{ij}^k = u_i^k - p_{ij}^k \quad (4)$$

where  $u_i^k$  represents the  $k$ -th feature in the  $i$ -th image,  $r_{ij}^k$  is the residue of the projection of the  $k$ -th feature of the  $j$ -th image in  $i$ -th image, and  $p_{ij}^k$  is the projection of the image  $j$  to image  $i$  of the corresponding point  $u_j^m$ , represented by Equation (5b).

$$p_{ij}^k = K_i R_i R_j^T K_j^{-1} u_j^m \quad (5a)$$

$$p_{ij}^k = H_{ij} u_j^m \quad (5b)$$

Therefore, each camera is defined by rotation  $(\phi_i, \theta_i)$ , focal length  $(f_i)$  and optical centers  $(c_i)$ . Thus, the cost function is given by the error represented by the squared sum of the residual errors of all images, presented by Equation (6).

$$C(\{(\phi_i, \theta_i, f_i, c_i)\}_{i=1}^N) = \sum_{i=1}^N \sum_{j \in I(i)} \sum_{k \in F(i,j)} (r_{ij}^k)^2 \quad (6)$$

$N$  is the number of images,  $I(i)$  is the set of images corresponding to the image  $i$  and  $F(i, j)$  is the set of features between images  $i$  and  $j$ . The problem consists in minimizing the Equation (6). The parameters values will be found using optimization methods based on bio-inspired algorithm composed of a set of constraints related to the minimum and maximum boundary values of the parameters. Thus, the optimization problem mathematical model can be described according to Equations (7) and (8).

$$\text{Minimize} \quad \text{fitness} = C(\{(\phi_i, \theta_i, f_i, c_i)\}_{i=1}^N) \quad (7)$$

Subject to:

$$\begin{aligned} \phi_i^{min} &\leq \phi_i \leq \phi_i^{max} \\ \theta_i^{min} &\leq \theta_i \leq \theta_i^{max} \\ f_i^{min} &\leq f_i \leq f_i^{max} \\ c_i^{min} &\leq c_i \leq c_i^{max} \end{aligned} \quad (8)$$

As originally conceived, this problem is unrestricted. This represent a problem to population based algorithms as the search space is too wide. In the optimization process, the values obtained during image capture by the robotic system were taken as references for the parameters to be adjusted. In order to address the fact that data provided by the acquisition device might have errors that can cause misalignment, the bounds described in Equation (8) is used to accommodate the sensors' uncertainty and errors. Thus, these bounds that constrain the parameters between a minimum and maximum value, for this solution, were established empirically. Through tests, the limits for the the rotation angles were set in a range of +5 and −5 degrees of the original acquisition values and focal length and optical centers limited by +5 and −5 pixels from calibration values.

The feasibility of the solutions in metaheuristic methods can be assured in one of two ways: objective function cost penalization for unfeasible generated solutions or, restricting unfeasible generated solutions that surpasses a bound to its value. In this particular study, the latter form is used in order to maintain the population of solutions in a limited search space thus, ensuring the feasibility of the solutions.

#### 2.4.2. Bat Algorithm

As proposed by Yang [42], Bat Algorithm (BA) is a metaheuristic optimization algorithm based on the echolocation behavior of bats, in which bats emits loud pulses in varying frequencies to detect and seek preys or to avoid hitting objects. In that matter, a mathematical analogy was developed to implement it as an optimization problem solving algorithm. To control exploration and exploitation process, the frequency ( $f$ ) and loudness amplitude ( $A$ ) parameters are modified during the iterative process according to (9) and (10). The  $\alpha$  parameter represents the decrease rate of amplitude and  $\beta$  is a random vector in the interval [0, 1].

$$fr_i = fr_{min} + (fr_{max} - fr_{min})\beta \quad (9)$$

$$A_i^{t+1} = \alpha A_i^t \quad (10)$$

The positions ( $X_i^t$ ) and velocities ( $V_i^t$ ) of the bats are described from Equations (11)–(13).

$$V_i^{t+1} = V_i^t + (X_i^t - X_*^t)fr_i \quad (11)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (12)$$

For the local exploitation stage, a new bat is generated according to a solution selected among the current best solutions, using a random walk as described in Equation (13).

$$X_i^{t+1} = X_*^t + \epsilon \cdot \text{mean}(A_i^t) \quad (13)$$

Also, the pulse emission rate increases once a bat is getting closer to a prey according to (14) in which the  $\lambda$  represents the increase rate.

$$r_i^{t+1} = 1 - e^{-\lambda t} \quad (14)$$

The Bat Algorithm pseudocode is given in Algorithm 1.

**Algorithm 1** Bat Algorithm (BA) Pseudocode.

---

```

Initialize parameters:  $\alpha, \lambda$ 
Initialize bats:  $X_i (i = 1, 2, 3, \dots, n)$ 
Evaluate bats:  $F_{ob}(X_i)$ 
Update best bat:  $X_*^t$ 
while stop criteria isn't met do
  for  $i \leftarrow 1, n$  do
     $fr_i = fr_{min} + (fr_{max} - fr_{min})\beta, \beta \in [0, 1]$ 
     $V_i^{t+1} = V_i^t + (X_i^t - X_*^t)fr_i$ 
     $X_i^{t+1} = V_i^{t+1} + X_i^t$ 
    if  $r_i > rand, rand \in [0, 1],$  then
       $X_i^{t+1} = X_*^t + \varepsilon \cdot mean(A_i^t), \varepsilon \in [-1, 1]$ 
    end if
    Verify bounds ( $X_i^{t+1}$ )
    Evaluate bat  $f(X_i^{t+1})$ 
    if  $rand < A_i^t \ \& \ f(X_i^{t+1}) \leq f(X_i^t), rand \in [0, 1],$  then
       $r_i^{t+1} = 1 - exp(-\lambda t)$ 
       $A_i^{t+1} = \alpha A_i^t$ 
    end if
    Update best bat  $X_*^t$ 
  end for
end while

```

---

## 2.4.3. Grey Wolf Optimizer

First published in 2014 [43], the Grey Wolf Optimizer (GWO), is a bio-inspired optimization metaheuristic, based on the hunting behavior of grey wolves. The algorithm takes into account their social interaction and hierarchy, as well as their approaching pattern when hunting. The wolves represent possible solutions for the problem and are divided as follows:  $\alpha$  the best search agent,  $\beta$  the second best agent,  $\delta$  the third best agent and  $\omega$  the rest of the population. The searching process described by a movement of the wolves according to the  $\alpha, \beta$  and  $\delta$  positions as described by Equations (15)–(17).

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha^t - \vec{X}_i^t| \quad \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta^t - \vec{X}_i^t| \quad \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta^t - \vec{X}_i^t| \quad (15)$$

$$\vec{X}_1 = \vec{X}_\alpha^t - \vec{A}_1 \cdot \vec{D}_\alpha \quad \vec{X}_2 = \vec{X}_\beta^t - \vec{A}_2 \cdot \vec{D}_\beta \quad \vec{X}_3 = \vec{X}_\delta^t - \vec{A}_3 \cdot \vec{D}_\delta \quad (16)$$

$$\vec{X}_i^{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (17)$$

The GWO uses a searching parameter  $a$  to control the exploitation and exploration process, which is decreased linearly from 2 to 0 over the course of the iterations, and is used to limit the fluctuation range of the parameter  $\vec{A}$ . Note that when the process reaches half of the max iteration value, the wolves can only move between their position and the position of the prey as  $\vec{A}$  is limited by the interval  $[-1, 1]$  and the exploitation stage of the algorithm starts.

The Grey Wolf Optimizer pseudocode is given in Algorithm 2.

**Algorithm 2** Grey Wolf Optimizer (GWO) Pseudocode.

---

```

Initialize wolfs:  $X_i (i = 1, 2, 3, \dots, n)$ 
Evaluate wolfs:  $F_{ob}(X_i)$ 
Hierarchy definition:  $X_\alpha, X_\beta, X_\delta$ 
while Stop criteria isn't met, do
    Update search parameter:  $a^t$ 
    for  $i \leftarrow 1, n$  do
         $\vec{A} = a^t \cdot \vec{r}_1 - a^t, \vec{r}_1 \in [0, 1]$ 
         $\vec{C} = 2 \cdot \vec{r}_2, \vec{r}_2 \in [0, 1]$ 
         $\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha^t - \vec{X}_i^t|$ 
         $\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta^t - \vec{X}_i^t|$ 
         $\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta^t - \vec{X}_i^t|$ 
         $\vec{X}_1 = \vec{X}_\alpha^t - \vec{A}_1 \cdot \vec{D}_\alpha$ 
         $\vec{X}_2 = \vec{X}_\beta^t - \vec{A}_2 \cdot \vec{D}_\beta$ 
         $\vec{X}_3 = \vec{X}_\delta^t - \vec{A}_3 \cdot \vec{D}_\delta$ 
         $\vec{X}_i^{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$ 
        Wolf's hierarchy update:  $X_\alpha, X_\beta, X_\delta$ 
    end for
end while

```

---

## 2.4.4. Salp Swarm Algorithm

The Salp Swarm algorithm is a bio-inspired optimization metaheuristic. Firstly published in 2017 [44] and further reviewed in [45], it was inspired by the behavior of salps when navigating and foraging in oceans. As a way to mathematically model the movement of the salp chain, the population is divided into two types: the leader ( $X_{i=1}$ ) and the followers ( $X_{i>1}$ ). The leader is the salp in front of the chain and every other salp is considered as a follower. The best solution obtained so far is modeled as a food source ( $X_*$ ) and the position update of the leader is done according to its current position as described in Equation (18).

$$\begin{cases} X_1^{t+1} = X_* - C_1 [(b_{up} - b_{lo}) \cdot C_2 + b_{lo}] & C_3 \leq 0.5 \\ X_1^{t+1} = X_* + C_1 [(b_{up} - b_{lo}) \cdot C_2 + b_{lo}] & C_3 > 0.5 \end{cases} \quad (18)$$

The position update of the follower salps is given by Equation (19).

$$X_i^{t+1} = \frac{X_i^t + X_{i-1}^t}{2}, \quad i \geq 2 \quad (19)$$

To control the exploration and exploitation in this method a search parameter  $C_1$  is proposed. This parameter is updated in each iteration as given by Equation (20).

$$C_1 = 2 \cdot \exp\left(-\left(\frac{4t}{t_{max}}\right)^2\right) \quad (20)$$

The Salp Swarm Algorithm pseudocode is given in Algorithm 3.



**Algorithm 3** Salp Swarm Algorithm (SSA) Pseudocode.

---

```

Initialize Salps:  $X_i (i = 1, 2, 3, \dots, n)$ 
Evaluate Salps:  $F_{ob}(X_i)$ 
Sort  $X$  by ascending order (minimization)
Update Best Salp  $X_*$ 
for  $t \leftarrow 1, t_{max}$  do
     $C_1 = 2 \cdot \exp\left(-\left(\frac{4t}{t_{max}}\right)^2\right)$ 
    for  $i \leftarrow 1, n$  do
        if  $i < n/2$ , then
             $C_3$  receives a random number between 0 and 1
             $C_2$  receives a random number between 0 and 1
            if  $C_3 < 0.5$ , then
                 $X_i^{t+1} = X_* - C_1 [(b_{up} - b_{lo}) \cdot C_2 + b_{lo}]$ 
            else
                 $X_i^{t+1} = X_* + C_1 [(b_{up} - b_{lo}) \cdot C_2 + b_{lo}]$ 
            end if
        else
             $X_i^{t+1} = \frac{X_i^t + X_{i-1}^t}{2}$ 
        end if
        Verify search bounds  $X_i^{t+1}$ 
        if  $F_{ob}(X_i^{t+1}) < F_{ob}(X_*)$ , then
             $X_* = X_i^{t+1}$ 
        end if
    end for
end for

```

---

## 2.4.5. Arithmetic Optimization Algorithm

The Arithmetic optimization algorithm is a novel metaheuristic method that was proposed in 2021 [46]. It is a population based technique and, as the name implies, its based on using Arithmetic operators. The algorithm uses a parameter called Math Optimizer Accelerated (MOA) to control the exploration and exploitation phases as described in (21). Min and Max denotes the maximum and minimum values and are user specified.

$$MOA(t) = Min \times t \left( \frac{Max - Min}{t_{max}} \right) \quad (21)$$

Another coefficient used in this algorithm is the Math Optimizer probability (MOP) as shown by Equation (22) in which the  $\alpha$  denotes a exploitation sensitivity parameter.

$$MOP(t) = 1 - \frac{t^{1/\alpha}}{t_{max}^{1/\alpha}} \quad (22)$$

While in the exploration phase, the algorithm uses the Division and Multiplication operators according to Equation (23).

$$x_{i,d}^{t+1} = \begin{cases} x_d^* \div (MOP + \epsilon) \times ((X_* - C_1 [(b_{d_{up}} - b_{d_{lo}}) \times \mu + b_{d_{lo}}]) & r_2 < 0.5 \\ x_d^* \times MOP \times ((X_* - C_1 [(b_{d_{up}} - b_{d_{lo}}) \times \mu + b_{d_{lo}}]) & r_2 \geq 0.5 \end{cases} \quad (23)$$

The exploitation phase of the algorithm uses the Addition and Subtraction operators and is described in Equation (24).

$$x_{i,d}^{t+1} = \begin{cases} x_d^* - MOP \times ((X_* - C_1 [(b_{d_{up}} - b_{d_{lo}}) \times \mu + b_{d_{lo}}]) & r_3 < 0.5 \\ x_d^* + MOP \times ((X_* - C_1 [(b_{d_{up}} - b_{d_{lo}}) \times \mu + b_{d_{lo}}]) & r_3 \geq 0.5 \end{cases} \quad (24)$$

The Arithmetic Optimization Algorithm pseudocode is given in Algorithm 4.

---

**Algorithm 4** Arithmetic Optimization Algorithm (AOA) Pseudocode.

---

```

Initialize parameters:  $\alpha, \mu$ 
Initialize solutions:  $X_i (i = 1, 2, 3, \dots, n)$ 
while  $t < t_{max}$  do
  Evaluate Solutions:  $F_{ob}(X_i)$ 
  Update Best solution:  $X_*$ 
  Update MOA value using EQ (21)
  Update MOP value using EQ (22)
  for  $i \leftarrow 1, n$  do
    for  $d \leftarrow 1, D$  do
       $r_1, r_2, r_3$  receives a random number between 0 and 1
      if  $r_1 > MOA$  then
        if  $r_2 > 0.5$  then
          Apply the Division math operator (D “÷”)
          Update the  $i$ th solutions’ positions using the first rule in Equation (23)
        else
          Apply the Multiplication math operator (M “×”)
          Update the  $i$ th solutions’ positions using the second rule in Equation (23)
        end if
      else
        if  $r_3 > 0.5$  then
          Apply the Subtraction math operator (S “−”)
          Update the  $i$ th solutions’ positions using the first rule in Equation (24)
        else
          Apply the Addition math operator (A “+”)
          Update the  $i$ th solutions’ positions using the second rule in Equation (24)
        end if
      end if
    end for
  end for
   $t = t + 1$ 
end while

```

---

#### 2.4.6. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a stochastic optimization method based on population developed in 1995 by James Kennedy and Russell Eberhart [47]. The algorithm is based on the collective behavior of flocks of birds and schools of fish by observing the activities of these individuals in dodging predators and searching for food. The initialization of the PSO algorithm occurs with the random creation of the particles that constitute the population (swarm). Particle determination consists of the description of its velocity ( $V_{(i,d)}$ ) and position ( $X_i$ ) which are expressed through Equations (25) and (26), respectively.

$$V_{(i,d)} = \omega V_{(i,d)} + c_1 \cdot r_p (X_{(i,d)}^{p*} - X_{(i,d)}) + c_2 \cdot r_g (X_{(d)}^{g*} - X_{(i,d)}) \quad (25)$$

$$X_i = X_i + V_i \quad (26)$$

As described by Equation (25), the velocity of each particle is updated according to the direction of the individual’s best found position ( $X_{(i,d)}^{p*}$ ) and in the best individual’s found position ( $X_{(d)}^{g*}$ ).  $\omega$  is a weight parameter that controls the impact of previous particle’s velocity on its current one;  $r_1, r_2$  are random variables uniformly distributed between [0, 1] and  $c_1, c_2$  are positive constants that control the maximum step size.

The Particle Swarm Optimization pseudocode is given in Algorithm 5.

**Algorithm 5** Particle Swarm Optimization (PSO) Pseudocode.

---

```

Initialize particles:  $X_i(i = 1, 2, 3, \dots, n)$ 
Initialize velocity for each particle:  $V_i(U[-(b_{up} - b_{lo}), (b_{up} - b_{lo})])$ 
Evaluate particles:  $F_{ob}(X_i)$ 
Update best position for each particle:  $X^{p*}$ 
Update swarm's best position:  $X^{g*}$ 
while stop criteria isn't met do
  for  $i \leftarrow 1, n$  do
    for  $d \leftarrow 1, D$  do
       $r_p, r_g$  receives a uniform random number between 0 and 1
       $V_{(i,d)} = \omega V_{(i,d)} + c_1 \cdot r_p (X_{(i,d)}^{p*} - X_{(i,d)}) + c_2 \cdot r_g (X_{(d)}^{g*} - X_{(i,d)})$ 
       $X_i = X_i + V_i$ 
      if  $F_{ob}(X_i) < F_{ob}(X^{p*})$  then
        Update particle's best position:  $X^{p*} = X_i$ 
        if  $F_{ob}(X^{p*}) < F_{ob}(X^{g*})$  then
          Update swarm's best position:  $X^{g*} = X^{p*}$ 
        end if
      end if
    end for
  end for
end while

```

---

## 2.4.7. Time Complexity Analysis for the Metaheuristics Algorithms

For the five metaheuristic algorithms focused in this article, the time complexity is dependant on the number of individuals/agents, dimension of the problem and maximum number of iterations. Overall, by analyzing the steps of each algorithm, the computational complexity  $t$  is stated in Equation (27).

$$O(t \cdot d \cdot n + C \cdot n) \quad (27)$$

where  $t$  is the number of iterations,  $d$  shows the number of dimensions,  $n$  indicates the number of search agents and  $C$  is the cost of objective function.

## 2.4.8. Levenberg-Maquardt

The Levenberg-Maquardt (LM) Algorithm, first proposed by Kenneth Levenberg [48] and later by Donald Marquardt [49], is a technique that numerically solves nonlinear function minimization problems iteratively by finding local minima of multivariable functions expressed by the sum of squares. This technique is a combination of the gradient descent and Gauss–Newton methods [50] which is based on a local linearization of the residuals according to (28). A vector function  $f$  maps  $\mathbf{p} \in R^m$  to estimate a measured value  $\hat{x} = f(\mathbf{p})$ ,  $\hat{x} \in R^n$ . It is desired to minimize the squared distance  $\epsilon^T \epsilon$  with  $\epsilon = x - \hat{x}$  for all  $\mathbf{p}$ , with the initial parameter  $\mathbf{p}_0$  and a measured value  $\mathbf{x}$  with the intention of at the end finding  $\mathbf{p}^+$  that satisfies the relation  $f$  locally.

$$f(\mathbf{p} + \delta_{\mathbf{p}}) \approx f(\mathbf{p}) + \mathbf{J}\delta_{\mathbf{p}} \quad (28)$$

$$\mathbf{J} = \frac{\partial f(\mathbf{p})}{\partial \mathbf{p}} \quad (29)$$

Equation (29) defines the Jacobian matrix of  $f$  with respect to  $\mathbf{p}$ . At each iteration a value for  $\delta_{\mathbf{p}}$  is estimated in order to minimize Equation (30).

$$\|x - f(\mathbf{p} + \delta_{\mathbf{p}})\| \approx \|x - f(\mathbf{p}) - \mathbf{J}\delta_{\mathbf{p}}\| = \|\epsilon - \mathbf{J}\delta_{\mathbf{p}}\| \quad (30)$$

The  $\delta_{\mathbf{p}}$  value is the solution of the linear least-squares problem where minimum is found when  $\mathbf{J}\delta_{\mathbf{p}} - \epsilon$  is orthogonal to the column space of  $\mathbf{J}$  [51]. The LM Algorithm actually solves a equation called as augmented normal equations presented in Equation (31).

$$(\mathbf{J}^T\mathbf{J} + \mu\mathbf{I})\delta_{\mathbf{p}} = \mathbf{J}^T\epsilon \quad \mu > 0 \quad (31)$$

where  $\mathbf{I}$  is the identity matrix, and  $\mu$  is known as the damping term, which is adjusted at each iteration to ensure error reduction.

The Levenberg-Maquardt pseudocode is given in Algorithm 6.

---

**Algorithm 6** Levenberg-Maquardt Pseudocode.
 

---

**Input:** A vector function  $f : R^m \rightarrow R^n$  with  $n \geq m$ , a measurement vector  $\mathbf{x} \in R^n$  and an initial parameters estimate  $\mathbf{p}_0 \in R^m$ .

**Output:** A vector  $\mathbf{p}^+ \in R^m$  minimizing  $\|\mathbf{x} - f(\mathbf{p})\|^2$

**Algorithm:**

$k := 0; v := 2; \mathbf{p} := \mathbf{p}_0;$

$\mathbf{A} := \mathbf{J}^T\mathbf{J}; \epsilon_p := \mathbf{x} - f(\mathbf{p}); \mathbf{g} := \mathbf{J}^T\epsilon_p;$

$\text{stop} := (\|\mathbf{g}\|_{\infty} \leq \epsilon_1); \mu := \tau * \max_{i=1, \dots, m} (A_{ii});$

**while** stop criteria isn't met and  $(k < k_{max})$  **do**

$k := k + 1;$

**repeat**

    Solve  $(\mathbf{A} + \mu\mathbf{I})\delta_{\mathbf{p}} = \mathbf{g};$

**if**  $(\|\delta_{\mathbf{p}}\| \leq \epsilon_2(\|\mathbf{p}\| + \epsilon_2))$  **then**

      stop:=true;

**else**

$\mathbf{p}_{new} := \mathbf{p} + \delta_{\mathbf{p}};$

$\rho := (\|\epsilon_p\|^2 - \|\mathbf{x} - f(\mathbf{p}_{new})\|^2) / (\delta_{\mathbf{p}}^T(\mu\delta_{\mathbf{p}} + \mathbf{g}));$

**if**  $\rho > 0$  **then**

        stop:= $(\|\epsilon_p\| - \|\mathbf{x} - f(\mathbf{p}_{new})\| < \epsilon_4\|\epsilon_p\|);$

$\mathbf{P} = \mathbf{P}_{new};$

$\mathbf{A} := \mathbf{J}^T\mathbf{J}; \epsilon_p := \mathbf{x} - f(\mathbf{p}); \mathbf{g} := \mathbf{J}^T\epsilon_p;$

        stop:=(stop) or  $(\|\mathbf{g}\|_{\infty} \leq \epsilon_1);$

$\mu := \mu * \max(\frac{1}{3}, 1 - (2\rho - 1)^3); v := 2;$

**else**

$\mu := \mu * v; v := 2 * v;$

**end if**

**end if**

**until**  $(\rho > 0$  or stop)

  stop:= $(\|\epsilon_p\| \leq \epsilon_3);$

**end while**

$\mathbf{p}^+ := \mathbf{p};$

---

### 2.5. Image Stitching Process

The image stitching process is initialized after adjusting the positioning of the images. First, the spherical projection of the images is performed with the desired final resolution. In this process, the images are mapped on a cartographic plane with a projection formulation that relates pixel and geographic coordinates [52]. The images are projected using the relations  $[u, v]^T = R[\theta, \phi]^T$ , where  $(\theta, \phi)$  are the horizontal and vertical angular directions (longitude and latitude, respectively), and  $(u, v)$  are pixel coordinates [53]. The projection provides a value  $(u, v)$  directly from a value  $(x, y, z)$ . A point  $\mathbf{P} = (x, y, z)$  is the 3D projection of a point  $(x, y)$  from the original image. Thus, the spherical coordinates can be obtained from the cartesian coordinates given by  $\mathbf{P}$ , according to the relations shown in (32) and (33):

$$\theta = \arctan \frac{y}{z} \quad (32)$$

$$\phi = \arccos \frac{z}{\sqrt{x^2 + y^2 + z^2}} \quad (33)$$

In spherical projection, a point can be mapped to a 2D plane simply by setting longitude as horizontal coordinate value and latitude as vertical [54]. After projecting the images there may be lines on the boundaries of the overlapping areas, for example due to the intensity of adjacent pixels differing. Therefore, a correction algorithm is applied to obtain a uniform panorama and remove visible seams. Multi-band Blending technique was used for image blending [55]. This technique uses Laplacian Pyramid with Gaussian Kernel to blend the images while keeping their significant features. The images are reduced to different levels with the Gaussian Kernel. Afterwards, a Laplacian Pyramid is created, calculated based on the subtraction between the Gaussian image and the expansion of the Gaussian pyramid lower level. The Laplacian pyramids are then mixed, resulting in a final panorama obtained by interpolating and adding all of the pyramid's levels [55].

### 2.6. Comparison Method

In order to evaluate the impact of using metaheuristics to improve the optimization problem of minimizing re-projection error in 360° panorama image stitching, the dispersion of final solutions, computational time and convergence curves are taken into consideration. In addition, an Analysis of Variance (ANOVA) and Tukey's Honest Significance (THS) Tests were performed to statistically evaluate the validity of the simulation results. Before each run, a novel group of initial solution is generated. As a way to ensure that the final result of the optimization will achieve a result at least as good as the previously known acquisition parameters, the latter are used as the first individual of the initial population. The rest of the population is generated randomly within the search space according to Equation (34).

$$X_{i,d} = [r]_0^1 \times (b_{d_{up}} - b_{d_{lo}}) + b_{d_{lo}}; \quad (34)$$

Then, all metaheuristic algorithms run with the same initial group to avoid starting-point bias interfering with the comparison. All methods are population based and the population is set to 35. In addition, the stopping criteria for all algorithms is set as 1000 iterations. In that regard, the total number of objective function evaluations per method is equal. This is meant to ensure that the algorithms return a final solution at least as good as the one that was acquired initially. This procedure is illustrated by Figure 3.

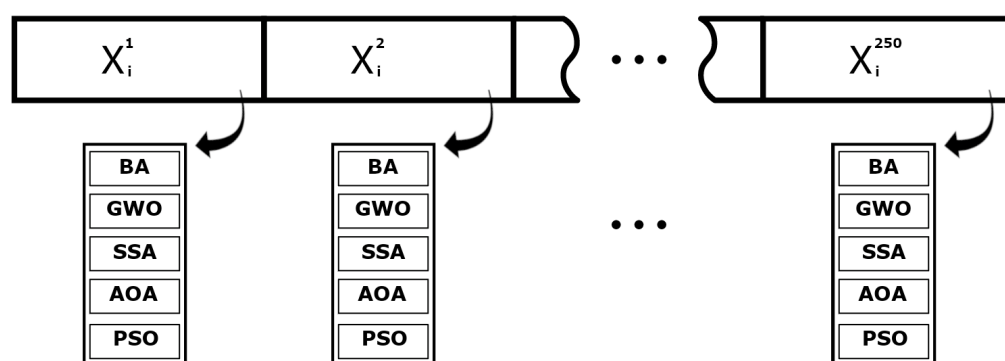


Figure 3. Database generation procedure.

Besides the population size and the maximum number of iterations, the techniques have other parameters and some are determined empirically. These parameters and definitions are presented in Table 1.

**Table 1.** Parameter values for the comparative algorithms.

Algorithm	Parameter	Value
BA	A	1
	r	1
	$\lambda$	0.01
	$\alpha$	0.9995
	$fr_{min}$	0
	$fr_{max}$	100
GWO	Convergence parameter ( $a$ )	Linear reduction from 2 to 0
AOA	$\alpha$	5
	$\mu$	0.49999
SSA	Convergence parameter ( $C_1$ )	According to Equation (20)
PSO	Topology	Global
	Cognitive and social constants	( $C_1, C_2$ ) 2,2
	Inertia weight	Linear reduction 0.9 to 0.2
	Velocity limit	6

### 3. Results and Discussion

This section presents the obtained results for each proposed optimization method, and some discussions about them. The simulations were conducted by using an Intel Core i7-7700HQ CPU 2.80 GHz computer with 16 GB of RAM and Windows 10 64-bit operating system. The proposed method was implemented in C++ programming language.

The proposed approaches are applied to image data from the Arts and Design School at the Federal University of Juiz de Fora, Brazil, from two different points of view, and compared with the Levenberg–Marquardt methodology. Figure 4 presents two points off view generated from the original image acquisition data.

**Figure 4.** Arts and Design School at the Federal University of Juiz de Fora: Two point of View.

#### 3.1. Case Study I

The performance of BA, GWO, AOA, SSA and PSO are compared with respect to: (i) the final value of the objective function; (ii) convergence; (iii) boxplots of the results and (iv) statics indices. In addition, the results are analyzed with respect to that obtained by the classical Levenberg–Marquardt method.

Performing the algorithms described in Section 2, convergence curves and average convergence for the metaheuristic algorithms were obtained as shown in Figures 5 and 6. Figure 5 shows the convergence history of each algorithms for the 250 simulations performed and Figure 6 shows the average curve of all convergence curves from all simulations.

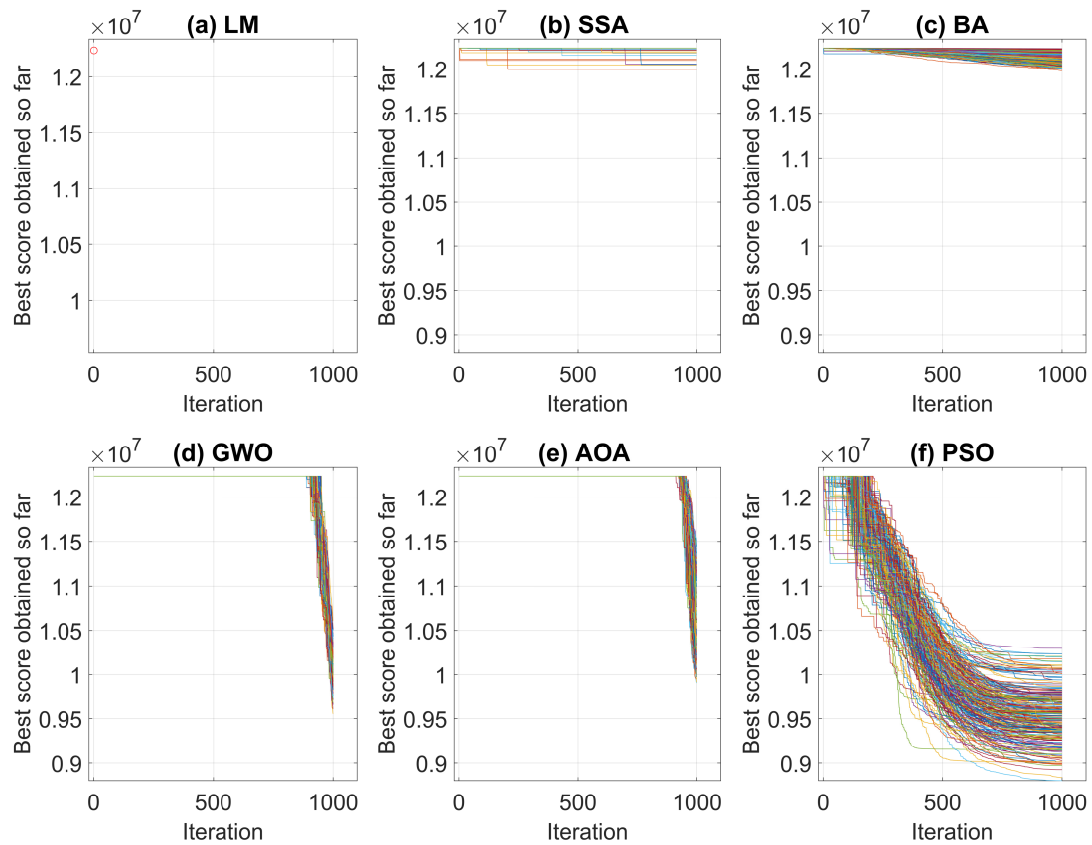


Figure 5. Convergence Curves in 250 simulations for Case Study I.

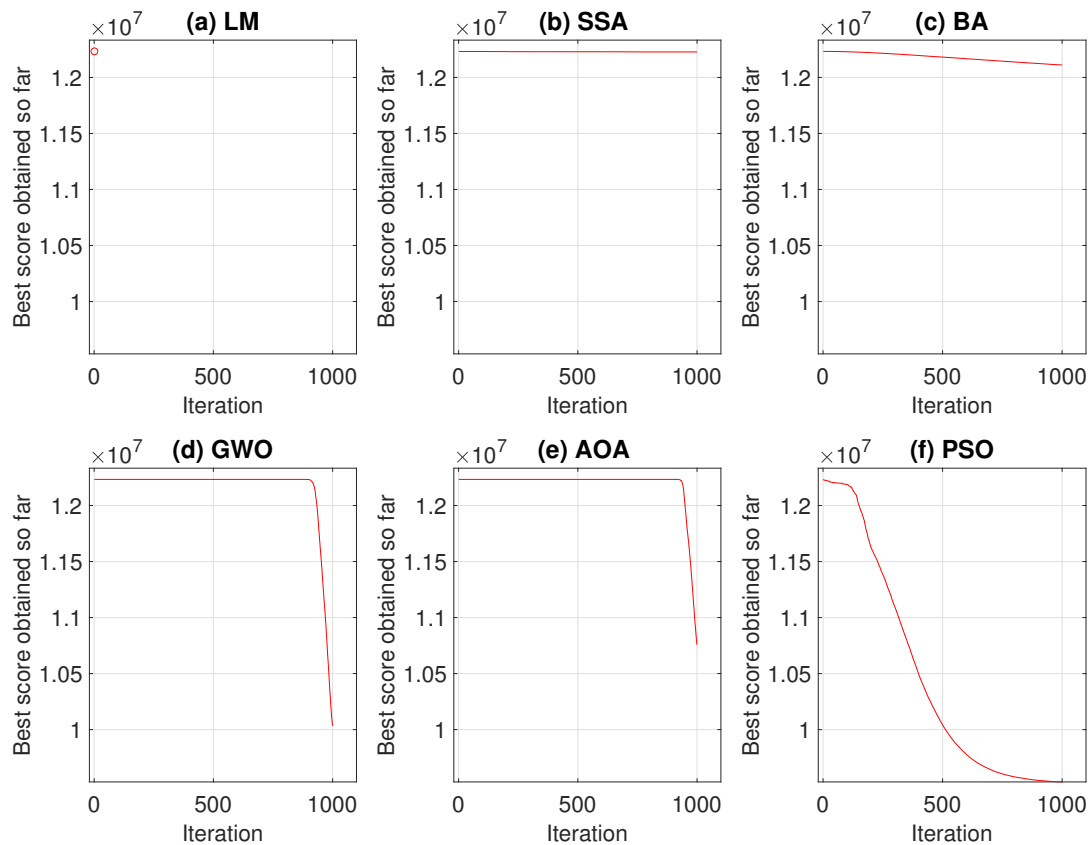


Figure 6. Average Convergence Curve.

Analyzing the responses obtained by the convergence curves presented in Figures 5 and 6, it can be seen that in Levenberg–Marquardt method, the initial point interferes with the final solution, and as the initial guess is equal to the solution acquired by the robot, the method converged to the same solution, i.e., the solution presented is already a local optimum point. Therefore, the presented metaheuristics obtained a better response, since they found new solutions with objective function values of less than the initial one. When comparing the metaheuristics, the techniques convergence verifies that PSO and BA present a better global search stage compared to the others methods. That is, at process beginning, the algorithms perform an exploration of solution region and find good solutions. Meanwhile, GWO and AOA obtained similar convergence characteristics by having their differential in the local search step.

For further analysis, boxplot test is carried out for all the considered algorithms. They are presented in Figure 7, showing the objective function optimal values considering a set of 250 simulations. Table 2 presents statistical indices like median, mean and standard deviation of the fitness value acquired. They can be used to explain the information enclosed in the boxplot figure, and also present the average time (computational effort) in seconds of each algorithm.

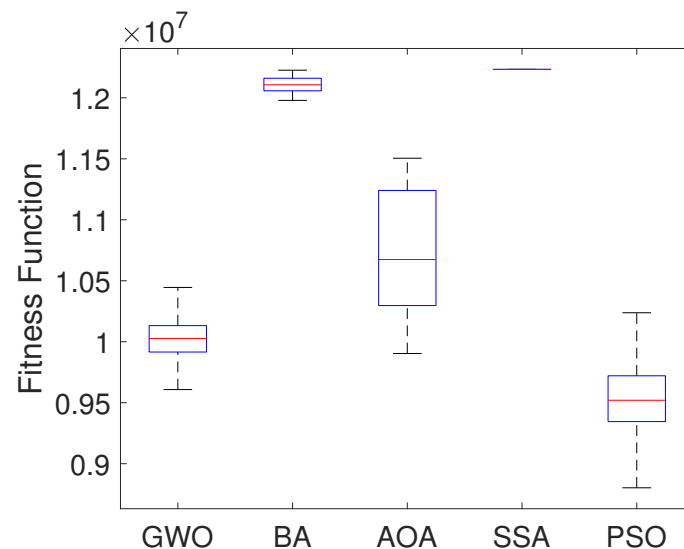


Figure 7. Boxplots of the Optimal Fitness Values from the 250 Simulations.

Table 2. Comparison of Fitness Values Obtained by Metaheuristics and Average Execution Time.

Algorithm	Minimum	Median	Mean	Standard Deviation	Average Time (sec)
GWO	$9.552480 \times 10^6$	$1.0027000 \times 10^7$	$1.0033 \times 10^7$	$1.7718 \times 10^5$	10.6541
BA	$1.1979000 \times 10^7$	$1.2106950 \times 10^7$	$1.2111 \times 10^7$	$6.4233 \times 10^4$	9.1809
AOA	$1.1504700 \times 10^7$	$1.0674300 \times 10^7$	$1.0757 \times 10^7$	$4.9586 \times 10^5$	9.7178
SSA	$1.1994400 \times 10^7$	$1.2233100 \times 10^7$	$1.2228 \times 10^7$	$2.8482 \times 10^4$	9.0267
PSO	$8.802380 \times 10^6$	$9.520155 \times 10^6$	$9.5329 \times 10^6$	$2.9654 \times 10^5$	9.2606

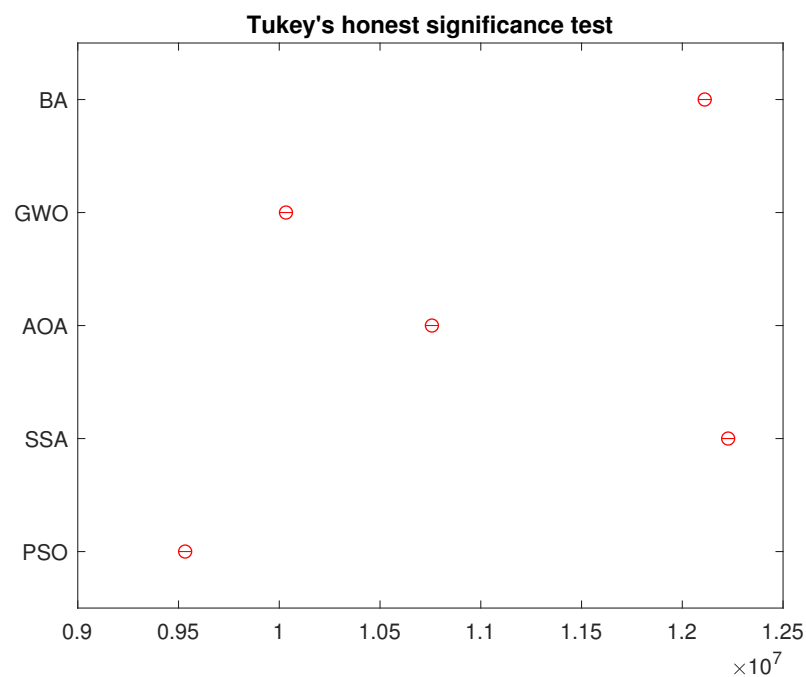
Based on results in Figure 7 and Table 2, it can be noticed that PSO Algorithm outperforms the other metaheuristic algorithms in the optimization by presenting the lowest median, mean and fitness values. On the other hand, SSA showed the worst results. Although the Bat algorithm presents the second highest median value, it presents a smaller dispersion of the data, which indicates stability in the results, while the Arithmetic Optimization Algorithm presents greater variation. Table 2 indicates that the computation time values of all algorithms are similar to each other.



In order to reject the null hypothesis, a variance test analysis is done, with results are shown in the Table 3. For further comparison, a Tukey's honest significance was also done and the results in Figure 8 assures that the groups are all different from each other.

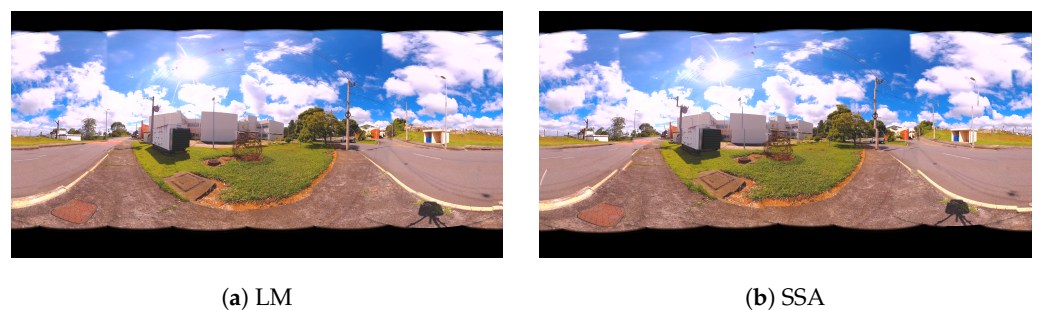
**Table 3.** Anova Table.

Source	SS	df	MS	F	P > F
<b>Treatments</b>	$1.466 \times 10^{15}$	4	$3.665 \times 10^{14}$	49,951.36	0
<b>Error</b>	$9.216 \times 10^{13}$	1245	$7.403 \times 10^{10}$	-	-
<b>Total</b>	$1.558 \times 10^{15}$	1249	-	-	-

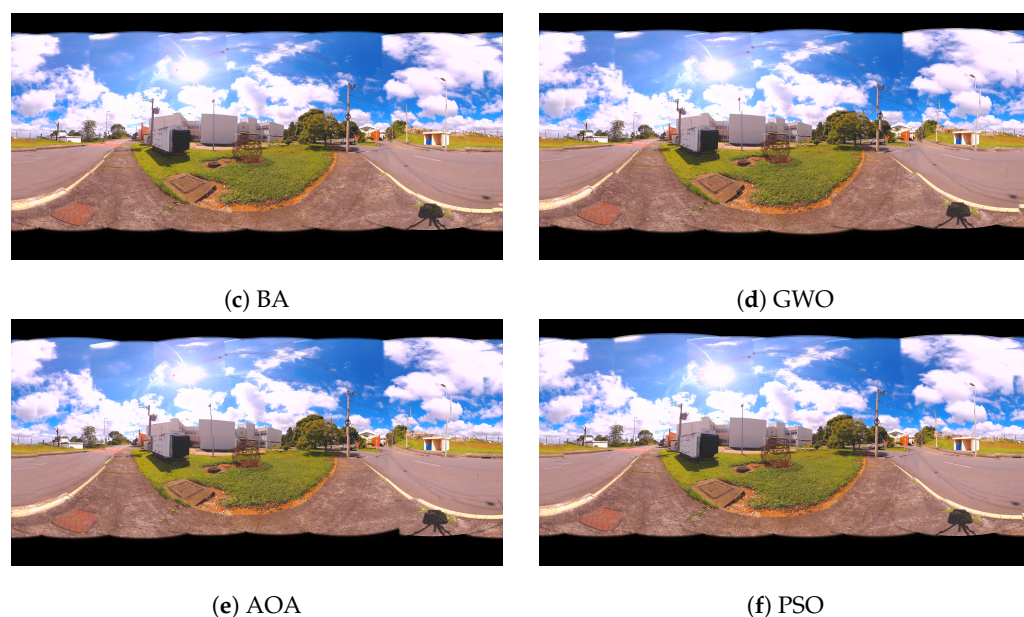


**Figure 8.** Tukey's Honest Significance Test from the 250 Simulations.

Panoramic images are obtained from the values found by the algorithms. In Figure 9, the panoramas generated by the results of LM, AOA, GWO and SSA methods are shown. Figure 10 illustrates the 360 panorama that obtained the best solution found by the PSO Algorithm.



**Figure 9.** Cont.



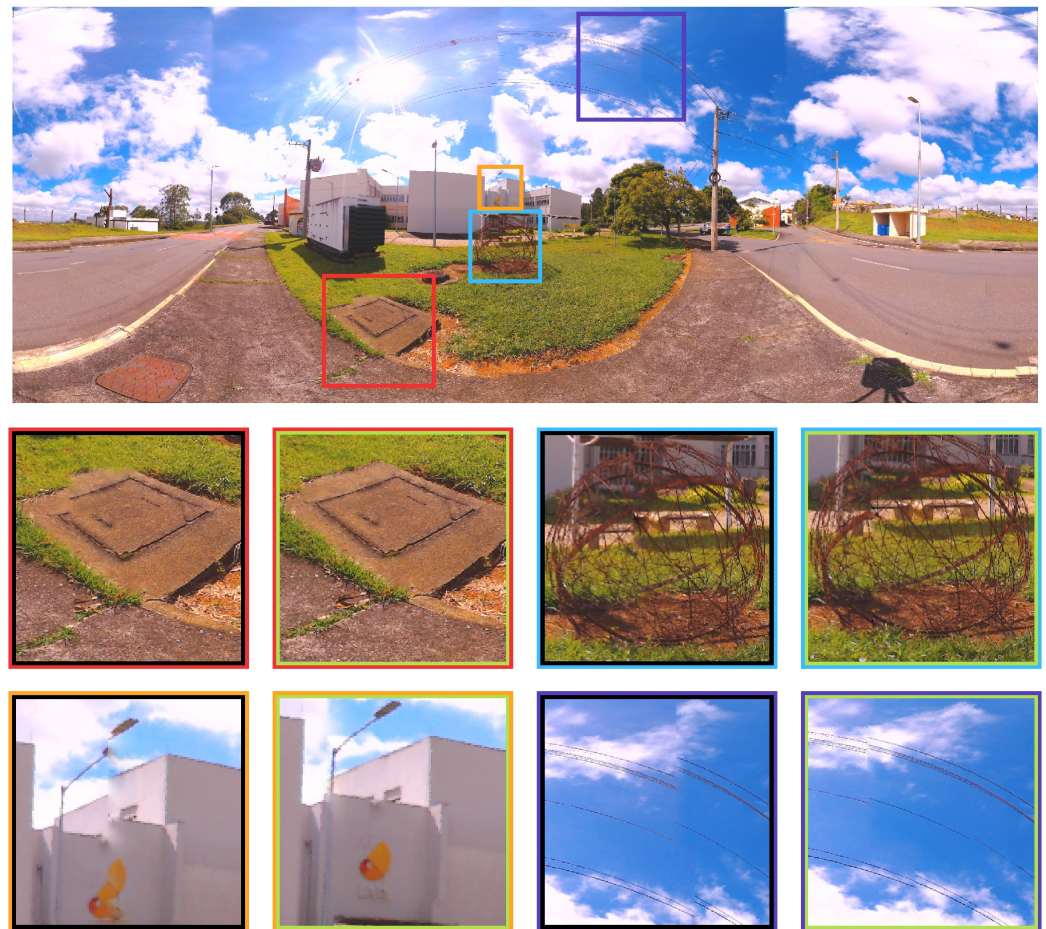
**Figure 9.** 360 Panoramic Images Resulting from the Proposed Metaheuristics for Case Study I: (a) Levenberg–Marquardt (LM), (b) Salp Swarm Algorithm (SSA), (c) Bat Algorithm (BA), (d) Grey Wolf Optimizer (GWO), (e) Arithmetic Optimization Algorithm (AOA) and (f) Particle Swarm Optimization (PSO).



**Figure 10.** 360 Panoramic Image Resulting from the best method for Case Study I: Particle Swarm Optimization (PSO).

For a clearer visualization, some improvements in the objective function can be visualized in the zoomed figures of the artifacts present in the comparison between the LM (inner black square) and PSO (inner green square) optimized images in Figure 11. Four spots were chosen to represent the difference (red, cyan, yellow and dark blue squares). The details are shown below in the same figure.

Even with these improvements, the Panorama image is still not perfect and show some artifacts and misalignment because of the multimodal nature of the Bundle adjustment problem. More image processing procedures can further be used to enhance this panorama's results. Although there is clear statistical improvement of the objective function, the RGB image still can be modified to become a more visually appealing panorama. In that regard, more procedures could further be used to enhance the panorama's results and are still under investigation.



**Figure 11.** Zoomed figures in four spots for Case study I, represented by the red, cyan, yellow and dark blue squares. The differences of each approach for these spots are shown below in the image.

### 3.2. Case Study II

The same analyses of Case Study I, presented in Section 3.1, were performed for Case Study II. Therefore, the results of convergence and average convergence curve of the methods can be seen in Figures 12 and 13.

For this case, Levenberg–Marquardt’s behavior converging to the initial local minimum solution is also notorious. Analyzing the convergence curves of the other methods, it is observed that, the Particle Swarm Optimization and Bat techniques present a better global search in the solution, while Grey Wolf Optimizer and Arithmetic Optimization Algorithm have their prominence in the local search. The Salp Swarm Algorithm in all simulations, on the other hand, kept the fitness value equal to the original, that is, the algorithm did not find better solutions during iterations.

Figure 14 presents the boxplots of all methods considering the minimum values found in the 250 simulations. Table 2 presents the statistical data for better results analysis obtained by the boxplots and the average computational efforts in seconds for each algorithm.

Analyzing the results of the Figure 14 and Table 4, it can be seen that the PSO Algorithm presented the smallest objective function value as well as median and mean, but in relation to the others it presented the largest standard deviation. The Grey Wolf optimizer presented the smallest dispersion of the data. On the other side, Salp Swarm Algorithm performed the worst result, not improving the solution. Table 4 indicates that the computational time values of all algorithms are similar to each other.

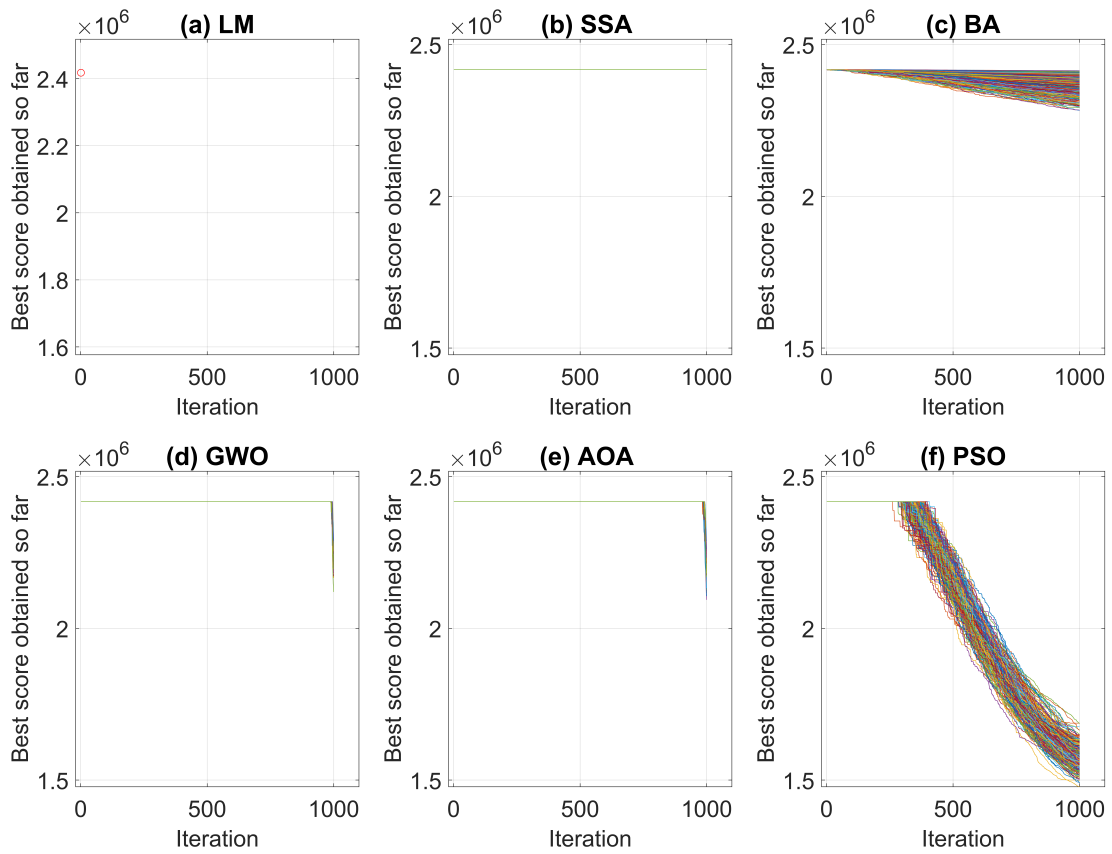


Figure 12. Convergence Curves in 250 simulations for Case Study II.

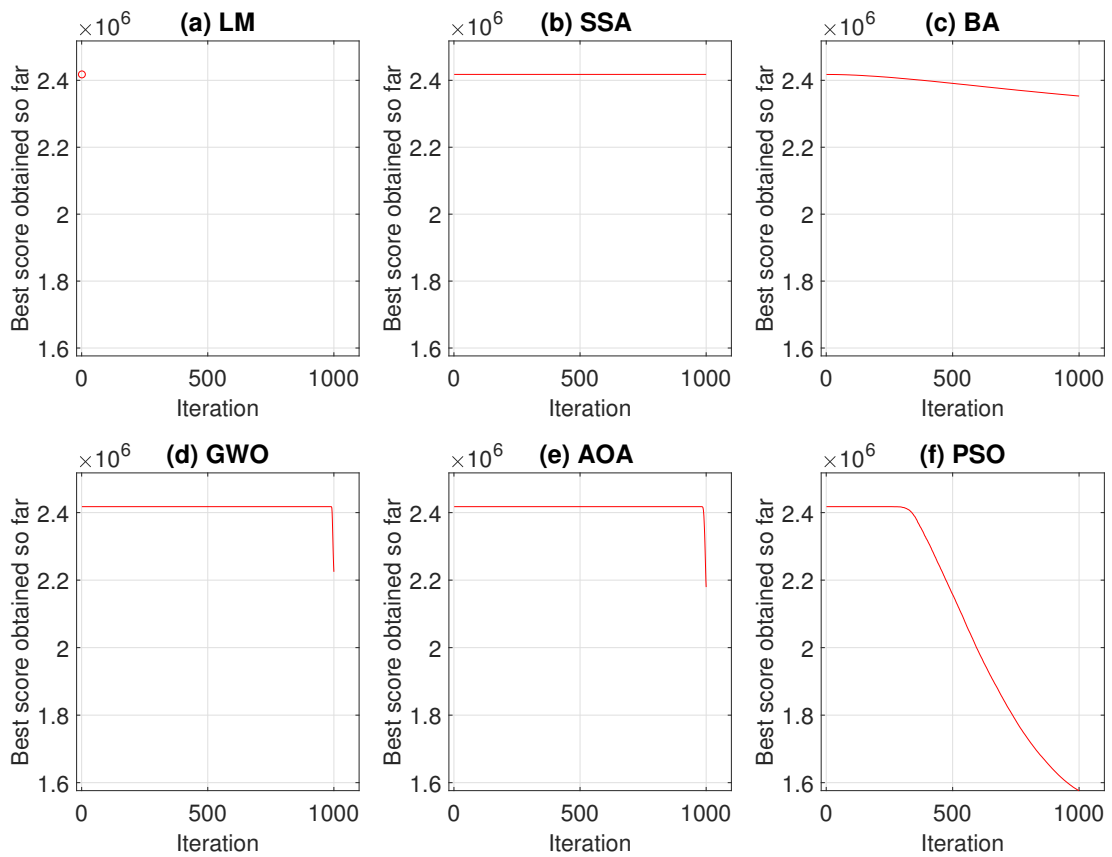


Figure 13. Average Convergence Curve.

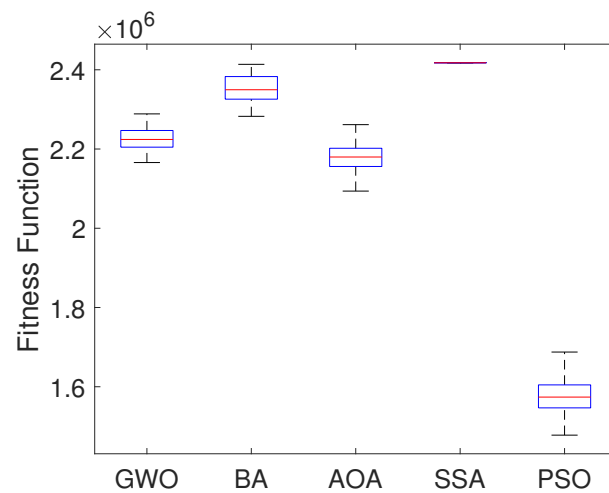


Figure 14. Boxplots of the optimal fitness values from the 250 simulations.

Table 4. Comparison of fitness values obtained by metaheuristics and average execution time.

Algorithm	Minimum	Median	Mean	Standard Deviation	Average Time (s)
GWO	$2.119770 \times 10^6$	$2.224145 \times 10^6$	$2.2247 \times 10^6$	$2.9250 \times 10^4$	15.1934
BA	$2.282500 \times 10^6$	$2.349620 \times 10^6$	$2.3530 \times 10^6$	$3.3987 \times 10^4$	13.7660
AOA	$2.179840 \times 10^6$	$2.179840 \times 10^6$	$2.1794 \times 10^6$	$3.3183 \times 10^4$	14.3220
SSA	$2.417660 \times 10^6$	$2.417660 \times 10^6$	$2.417660 \times 10^6$	0	13.4370
PSO	$1.477820 \times 10^6$	$1.573870 \times 10^6$	$1.5764 \times 10^6$	$4.0016 \times 10^4$	13.6258

In order to reject the null hypothesis, an analysis of variance test is done, and the results are shown in the Table 5. For further comparison, a Tukey's honest significance was also done and the results in Figure 15 assures that the groups are all different from each other.

Table 5. Anova Table.

Source	SS	df	MS	F	P > F
Treatments	$1.120 \times 10^{14}$	4	$2.802 \times 10^{13}$	29,726.08	0
Error	$1.174 \times 10^{12}$	1245	$9.426 \times 10^8$	-	-
Total	$1.132 \times 10^{14}$	1249	-	-	-

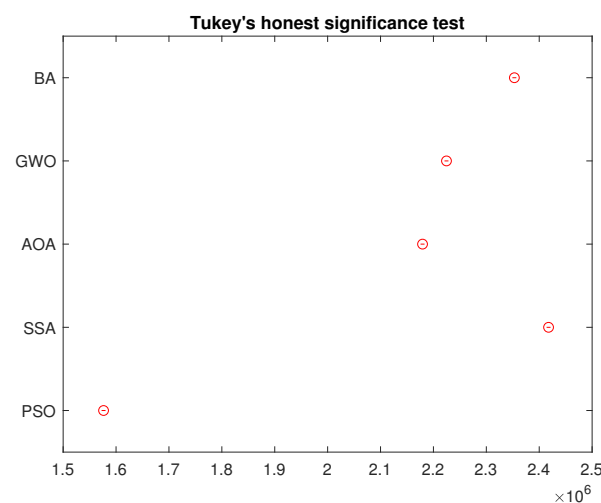
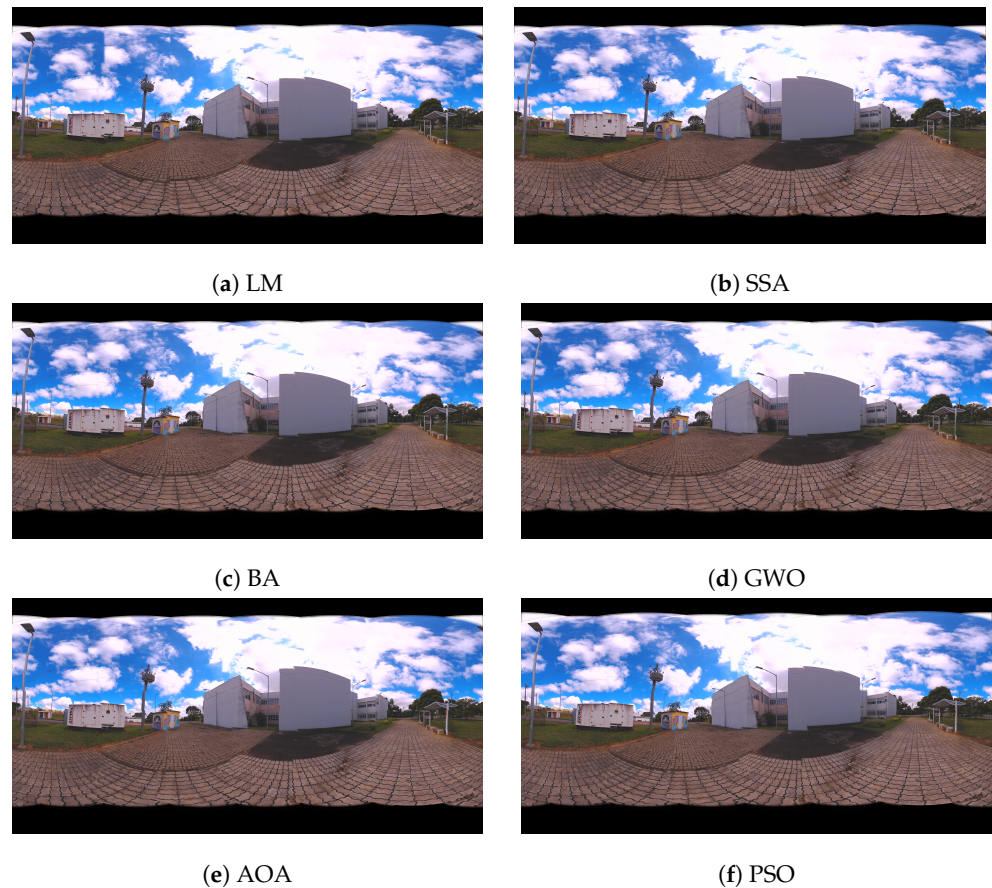


Figure 15. Tukey's Honest Significance Test from the 250 Simulations.

### Panorama Results

The results of optimization algorithms are shown in Figures 16 and 17 by means of the panoramic images generated. Figure 16 presents the panoramic results of LM, SSA, BA, GWO, AOA and PSO methods. Figure 17 illustrates the 360 panorama that obtained the best solution found by PSO Algorithm.



**Figure 16.** 360 Panoramic Images Resulting from the Proposed Metaheuristics for Case Study II: (a) Levenberg–Marquardt (LM), (b) Salp Swarm Algorithm (SSA), (c) Bat Algorithm (BA), (d) Grey Wolf Optimizer (GWO), (e) Arithmetic Optimization Algorithm (AOA) and (f) Particle Swarm Optimization (PSO).



**Figure 17.** 360 Panoramic Image Resulting from the best method for Case Study II : Particle Swarm Optimization (PSO).

For a clearer visualization, some improvements in the objective function can be visualized in the zoomed figures of the artifacts present in the comparison between the LM (inner black square) and PSO (inner green square) optimized images in Figure 18.



**Figure 18.** Case study II - Zoomed figures of four different spots represented by the red, cyan, yellow and dark blue squares show the nuance of each optimization approach.

Although the panorama improved, the image is still not perfect and show some artifacts and misalignment that might be caused, in the same way as in case I, by the multimodal nature of the Bundle adjustment problem. Even though there is statistical improvement of the objective function, the RGB image still has room for improvement. In addition, more procedures could further be used to enhance the panorama's results and are still under investigation.

For a more comprehensive visual comparison, an animated figure is available in the following link: <https://github.com/PvirtualGit/Sensors2021> (accessed on 2 June 2021).

#### 4. Conclusions and Future Work

The proposed research work presented the realization of bundle adjustment optimization through metaheuristics for 360 panoramic image generation. The main idea is to perform camera parameters optimization in order to achieve better alignment between images, comparing bio-inspired algorithms to the classical, derivative based, Levenberg–Marquardt used in the literature. The process was formulated as a problem to minimize the re-projection error, in order to find the best transformation that aligns the images to be stitched together.

The proposed techniques were evaluated in real scenarios and show good statistical results, proving an improvement in the solutions over the classical method, i.e., the metaheuristics obtained an improvement in the bundle adjustment problem. Although the

panorama images are still not perfect, the objective of this study was achieved as the bio-inspired algorithms obtained better results than those found by Levenberg–Marquardt.

For future works, the modeling of hybrid algorithms can be explored in order to aggregate the main advantages of each chosen technique. Also, the image processing might be improved using the gain compensation described in [17] and a weighted average of the matches cost in the objective function, both of which are under evaluation by the time this article was written.

**Author Contributions:** Conceptualization, methodology and writing M.J.R.A., T.d.R.A. and I.C.S.J.; funding acquisition, L.M.H.; review and editing I.C.S.J., L.M.H. and V.F.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by TBE and EDP over supervision of ANEEL under grant number PD-02651-0013/2017.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No available data.

**Acknowledgments:** This research was supported by TBE and EDP under supervision of ANEEL—The Brazilian Regulatory Agency of Electricity. The authors would like to thank the following Brazilian Federal Agencies UFJF, CNPq and INESC Brazil.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Szeliski, R. Image alignment and stitching. In *Handbook of Mathematical Models in Computer Vision*; Springer: Cham, Switzerland, 2006; pp. 1–15.
2. Melo, A.G.; Pinto, M.F.; Honorio, L.M.; Dias, F.M.; Masson, J.E. 3D Correspondence and Point Projection Method for Structures Deformation Analysis. *IEEE Access* **2020**, *8*, 177823–177836. [[CrossRef](#)]
3. Pang, S.; Sun, M.; Hu, X.; Zhang, Z. SGM-based seamline determination for urban orthophoto mosaicking. *ISPRS J. Photogramm. Remote Sens.* **2016**, *112*, 1–12. [[CrossRef](#)]
4. Attard, L.; Debono, C.J.; Valentino, G.; Di Castro, M. Tunnel inspection using photogrammetric techniques and image processing: A review. *ISPRS J. Photogramm. Remote Sens.* **2018**, *144*, 180–188. [[CrossRef](#)]
5. Biundini, I.Z.; Melo, A.G.; Pinto, M.F.; Marins, G.M.; Marcato, A.L.; Honorio, L.M. Coverage Path Planning Optimization for Slopes and Dams Inspection. In Proceedings of the Iberian Robotics Conference 2019, Porto, Portugal, 20–22 November 2019; pp. 513–523.
6. Pinto, M.F.; Marcato, A.L.; Melo, A.G.; Honório, L.M.; Urdiales, C. A framework for analyzing fog-cloud computing cooperation applied to information processing of UAVs. In *Wireless Communications and Mobile Computing*; Hindawi: London, UK, 2019; pp. 273–292.
7. Xue, W.; Zhang, Z.; Chen, S. Ghost Elimination via Multi-Component Collaboration for Unmanned Aerial Vehicle Remote Sensing Image Stitching. *Remote Sens.* **2021**, *13*, 1388. [[CrossRef](#)]
8. Li, X.; Feng, R.; Guan, X.; Shen, H.; Zhang, L. Remote sensing image mosaicking: Achievements and challenges. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 8–22. [[CrossRef](#)]
9. Hoang, V.D.; Tran, D.P.; Nhu, N.G.; Pham, V.H.; others. Deep feature extraction for panoramic image stitching. In Proceedings of the Asian Conference on Intelligent Information and Database Systems 2020, Phuket, Thailand, 23–26 March 2020; pp. 141–151.
10. Wang, M.; Niu, S.; Yang, X. A novel panoramic image stitching algorithm based on ORB. In Proceedings of the 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 13–17 May 2017; pp. 818–821.
11. Ghosh, D.; Kaabouch, N. A survey on image mosaicking techniques. *J. Vis. Commun. Image Represent.* **2016**, *34*, 1–11. [[CrossRef](#)]
12. Mills, A.; Dudek, G. Image stitching with dynamic elements. *Image Vis. Comput.* **2009**, *27*, 1593–1602. [[CrossRef](#)]
13. Li, J.; Deng, B.; Tang, R.; Wang, Z.; Yan, Y. Local-adaptive image alignment based on triangular facet approximation. *IEEE Trans. Image Process.* **2019**, *29*, 2356–2369. [[CrossRef](#)]
14. Chen, J.; Xu, Q.; Luo, L.; Wang, Y.; Wang, S. A robust method for automatic panoramic UAV image mosaic. *Sensors* **2019**, *19*, 1898. [[CrossRef](#)]
15. Agarwala, A.; Dontcheva, M.; Agrawala, M.; Drucker, S.; Colburn, A.; Curless, B.; Salesin, D.; Cohen, M. Interactive digital photomontage. In *ACM SIGGRAPH 2004 Papers*; ACM: Los Angeles, CA, USA, 2004; pp. 294–302.
16. Eden, A.; Uyttendaele, M.; Szeliski, R. Seamless image stitching of scenes with large motions and exposure differences. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; pp. 2498–2505.



17. Brown, M.; Lowe, D.G. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vis.* **2007**, *74*, 59–73. [[CrossRef](#)]
18. Pérez, P.; Gangnet, M.; Blake, A. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*; ACM: Los Angeles, CA, USA, 2003; pp. 313–318.
19. Triggs, B.; McLauchlan, P.F.; Hartley, R.I.; Fitzgibbon, A.W. Bundle adjustment—A modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms 1999*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 298–372.
20. Honório, L.; da Silva, A.L.; Barbosa, D.; Delboni, L. Solving optimal power flow problems using a probabilistic  $\alpha$ -constrained evolutionary approach. *IET Gener. Transm. Distrib.* **2010**, *4*, 674–682. [[CrossRef](#)]
21. Poubel, R.; De Oliveira, E.; Manso, L.; Honório, L.; Oliveira, L. Tree searching heuristic algorithm for multi-stage transmission planning considering security constraints via genetic algorithm. *Electr. Power Syst. Res.* **2017**, *142*, 290–297. [[CrossRef](#)]
22. De Mello Honório, L.; da Silva, A.M.L.; Barbosa, D.A. A gradient-based artificial immune system applied to optimal power flow problems. In *Proceedings of the International Conference on Artificial Immune Systems 2007*, Santos, Brazil, 26–29 August 2007; pp. 1–12.
23. Oliveira, E.J.; Oliveira, L.W.; Pereira, J.; Honório, L.M.; Junior, I.C.S.; Marcato, A. An optimal power flow based on safety barrier interior point method. *Int. J. Electr. Power Energy Syst.* **2015**, *64*, 977–985. [[CrossRef](#)]
24. Da Silva, A.L.; Rezende, L.; Honório, L.; Manso, L. Performance comparison of metaheuristics to solve the multi-stage transmission expansion planning problem. *IET Gener. Transm. Distrib.* **2011**, *5*, 360–367. [[CrossRef](#)]
25. Honório, L.M.; Costa, E.B.; Oliveira, E.J.; de Almeida Fernandes, D.; Moreira, A.P.G. Persistently-exciting signal generation for optimal parameter estimation of constrained nonlinear dynamical systems. *ISA Trans.* **2018**, *77*, 231–241. [[CrossRef](#)]
26. Rezende, H.; Silva, M.; Santos, M.; Honório, L.; Silva, L.; Vidal, V.; Ribeiro, J.; Cerqueira, A.; Pancoti, A.; Regina, B. Signal estimation for UAV control loop identification using artificial immune systems. In *Proceedings of the 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania, 10–12 October 2018; pp. 579–584.
27. Min, Z.; Jiguo, Z.; Xusheng, X. Panorama stitching based on sift algorithm and Levenberg–Marquardt optimization. *Phys. Procedia* **2012**, *33*, 811–818. [[CrossRef](#)]
28. Brown, M.; Lowe, D.G. Recognising panoramas. In *Proceedings of the ICCV, Nice, France, 14–17 October 2003*; p. 1218.
29. Allawi, Z.T.; Ibraheem, I.K.; Humaidi, A.J. Fine-tuning meta-heuristic algorithm for global optimization. *Processes* **2019**, *7*, 657. [[CrossRef](#)]
30. Silva, M.F.; Ribeiro, A.C.; Santos, M.F.; Carmo, M.J.; Honório, L.M.; Oliveira, E.J.; Vidal, V.F. Design of angular PID controllers for quadcopters built with low cost equipment. In *Proceedings of the 2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania, 13–15 October 2016; pp. 216–221.
31. Santos, M.F.; Honório, L.M.; Costa, E.B.; Oliveira, E.J.; Visconti, J.P. Active fault-tolerant control applied to a hexacopter under propulsion system failures. In *Proceedings of the 2015 19th International Conference on System Theory, Control and Computing (ICSTCC)*, Cheile Gradistei, Romania, 14–16 October 2015; pp. 447–453.
32. Santos, M.; Pereira, V.S.; Ribeiro, A.C.; Silva, M.F.; Carmo, M.J.; Vidal, V.F.; Honório, L.M.; Cerqueira, A.S.; Oliveira, E.J. Simulation and comparison between a linear and nonlinear technique applied to altitude control in quadcopters. In *Proceedings of the 2017 18th International Carpathian Control Conference (ICCC)*, Sinaia, Romania, 28–31 May 2017; pp. 234–239.
33. Aguiar, M.; Alves, T.; Pinto, L.; Honório, L.; Junior, I.; Vidal, V. Base Images for 360 Degrees Panorama with Camera Poses. 2021. Available online: <https://data.mendeley.com/datasets/89bhydjkzv/1> (accessed on 16 July 2021).
34. Ma, J.; Jiang, X.; Fan, A.; Jiang, J.; Yan, J. Image matching from handcrafted to deep features: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 23–79. [[CrossRef](#)]
35. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
36. Li, Y.; Wang, Y.; Huang, W.; Zhang, Z. Automatic image stitching using SIFT. In *Proceedings of the 2008 International Conference on Audio, Language and Image Processing*, Shanghai, China, 7–9 July 2008; pp. 568–571.
37. Adel, E.; Elmogy, M.; Elbakry, H. Image stitching based on feature extraction techniques: A survey. *Int. J. Comput. Appl.* **2014**, *99*, 1–8. [[CrossRef](#)]
38. Shaikh, T.S.; Patankar, A.B. Multiple feature extraction techniques in image stitching. *Int. J. Comput. Appl.* **2015**, *123*, 15.
39. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
40. Bandaru, S.; Deb, K. Metaheuristic techniques. *Decis. Sci. Theory Pract.* **2016**, *220*, 693–750.
41. Yong, H.; Huang, J.; Xiang, W.; Hua, X.; Zhang, L. Panoramic background image generation for PTZ cameras. *IEEE Trans. Image Process.* **2019**, *28*, 3162–3176. [[CrossRef](#)]
42. Yang, X.S. A new metaheuristic bat-inspired algorithm. In *Proceedings of the Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Granada, Spain, 12–14 May 2010; pp. 65–74.
43. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
44. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [[CrossRef](#)]
45. Faris, H.; Mirjalili, S.; Aljarah, I.; Mafarja, M.; Heidari, A.A. Salp swarm algorithm: Theory, literature review, and application in extreme learning machines. *Nat. Inspired Optim.* **2020**, 185–199.

46. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [[CrossRef](#)]
47. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
48. Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* **1944**, *2*, 164–168. [[CrossRef](#)]
49. Marquardt, D.W. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441. [[CrossRef](#)]
50. Transtrum, M.K.; Sethna, J.P. Improvements to the Levenberg–Marquardt algorithm for nonlinear least-squares minimization. *arXiv* **2012**, arXiv:1201.5885.
51. Lourakis, M.; Argyros, A.A. Is Levenberg–Marquardt the most efficient optimization algorithm for implementing bundle adjustment? In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, 17–21 October 2005; pp. 1526–1531.
52. Liu, S.; Zhao, L.; Li, J. Panorama measurement based on spherical projective geometry. *IJDTA* **2016**, *9*, 81–92. [[CrossRef](#)]
53. Barazzetti, L.; Previtali, M.; Scaioni, M. Procedures for condition mapping using 360 images. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 34. [[CrossRef](#)]
54. Wang, Y.; Li, Y.; Yang, D.; Chen, Z. A fast intra prediction algorithm for 360-degree equirectangular panoramic video. In Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, 10–13 December 2017; pp. 1–4.
55. Burt, P.J.; Adelson, E.H. A multiresolution spline with application to image mosaics. *ACM Trans. Graph. (TOG)* **1983**, *2*, 217–236. [[CrossRef](#)]