Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2023.1120000*

# Testing the Robustness of JAYA Optimization on 3D Surface Alignment of Range Images: A Revised Computational Study

**JOSÉ SANTAMARÍA**[1]

[1]Department of Computer Science, University of Jaén, 23071 Jaén, Spain (e-mail: jslopez@ujaen.es)

Corresponding author: José Santamaría (e-mail: jslopez@ujaen.es).

This work has been funded through Action No. 1 of the Operational Plan to Support Research of the University of Jaén in its biannual period 2023-2024.

**ABSTRACT** The further development of image registration (IR) as automated image alignment techniques is a well-known concern in the field of computer vision (CV). These techniques have been implemented in many real-world scenarios: from remote sensing to medical imaging to artificial vision and computer-aided design. There is great interest in applying original optimization algorithms to overcome the challenges associated with early IR methods (e.g. the ICP algorithm). On the other hand, algorithms rooted in evolutionary theories in nature-inspired computational models such as Evolutionary Computing (EC), have gained in importance over the past two decades. In addition, other algorithms with wide applicability fall into this category of methods, e.g. metaheuristics, swarmming, etc. Most of these methods have been widely adopted to address the IR problem, and are used as reliable alternatives for optimization goals. The aim of this paper is to address the following two main research challenges: (i) Bridging the gap in the revision of solutions proposed in recent years from the latter optimization model and from those ones from a new model based on deep learning (DL); and ii) introducing a new non-metaphor-based IR approach, called JAYA, to solve specific problems of aligning 3D surfaces of range images, also known as range image registration (RIR). In fact, as far as is known, this is the first time JAYA has been suggested for the above RIR problem. In particular, a new RIR method using the JAYA algorithm have been introduced and its performance has been accordingly compared against a wide set of methods from the SoTA. More than a dozen Softcomputing-based RIR methods have been included in the experimentation, making it the largest comparative study ever carried out in this category. In particular, range image datasets belonging to the SAMPL repository have been used, which has been widely adopted by many authors in the SoTA.

**INDEX TERMS** Computer Vision, Evolutionary Computation, Metaheuristics, Deep Learning, Image Registration, JAYA.

## I. INTRODUCTION

Image registration (IR) [1] was found to play a central role in the computer vision (CV) field. Specifically, IR represents one of the core resume efforts in the field of image processing. The focus is on the precise alignment of two or more images (taken at different times, from different sensors or perspectives) in a common coordinate system. IR is then routed to determine either a  geometric transformation or messaging (matching feature points) resulting in optimal overlap of the images considered. Its use extends to a wide range of real-world applications, where medical imaging and remote sensing are the most important areas that are widely explored in the State-of-The-Art (SoTA) [2]–[6].

Over time, various algorithms have been developed that address the issue of IR and contribute to extensive research. Basically, the IR challenge can be defined through the task of optimization, where the IR process revolves around identifying the optimal *transformation* that achieves the most appropriate/optimal alignment between the images. The estimation and optimization of the IR transformation is usually handled by an iterative method that systematically does search for the area of potential IR solutions.

In recent years, several fields of science have seen a growing fascination with the application of improved methods for building modern 3D models of real-world objects and scenes captured with the help of remote scanners. In the first contributions to IR, the optimization made using these early methods was negatively affected by factors such as image noise, image discrimination, and variations in the orders of magnitude within the parameters of the IR transformation. This was particularly evident in the iterative closest point (ICP) algorithm, where the process was highly interlocking in local optima [7], [8]. This particular challenge also arises because these IR methods require that the alignment process begin with precise image pre-alignments, usually performed by experts.

Instead, optimization methods that are *approximate* or *heuristic* coming from the Softcomputing paradigm [9], and often referred to as *metaheuristics* [10], [11], offer a compelling option. In particular, nature-inspired computing refers to a class of metaheuristic algorithms that simulate or are inspired by certain natural phenomena by means of the combination of grammar and randomness. *Evolutionary computation* (EC) [12], [13] is a discipline of nature-inspired optimization algorithms, and these methods demonstrate the ability to produce high-quality results for complex optimization challenges, especially when facing CV tasks [14]. The adoption of this optimization model has generated significant interest within the IR community over the past decade due to its successful results when dealing with the drawbacks of the early methods, i.e. ICP. In particular, evolutionary algorithms (EAs), such as genetic algorithms (GAs), have proven successful in treating the problem of IR. A large number of styles in this category have been published, especially in the last two decades, with many widely and comprehensively studied works found [4]–[6], [15].

A notable advantage of this outstanding IR optimization methods is that they can work without the need for an accurate estimate of the initial position of images. However, these methods require careful adjustment of several control parameters (the probability of mutation or crossover operators in GAs) to achieve optimal performance for each specific application addressed. Determining the appropriate values of control parameters is usually a labor-intensive process that is carried out manually. Users often lose time to improve these important parameters. Therefore, the automatic adjustment of these parameters has become one of the topics of great importance. The more skillfully the control parameter values are set, the more efficient will become the IR methods using this family of approximate algorithms [16]. Specifically, some papers have been published proposing these types of advanced optimization strategies, making them the most advanced designs available to date to solve the problem of IR [5]. However, despite the excellent results of the latter, there are still few proposals in the SoTA that deal with this particular type of optimization strategy.

Since the first efforts to address the IR problem through Softcomputing-based optimization approaches, this topic has become a very dynamic field of research that is witnessing a large influx of contributions. Nevertheless, due to new and innovative improvements in recent years, an updated review of SoTA is necessary. The first objective of this research is to provide a brief overview of the most important IR methods that use such smart technologies. In addition, other emerging approaches from the field of deep learning (DL) [17] applied to the IR problem will be explored.

As a secondary primary objective of this study, a new design of an IR method was proposed, which improved capabilities compared to SoTA. Specifically, the JAYA algorithm is presented as a pioneering optimization strategy that eliminates the need for meticulous tuning of control parameters [18], [19]. As far as is known, this is the first application of the JAYA algorithm to IR, especially in the context of 3D surface alignment of range images. Extensive experimental research has been conducted to evaluate the performance of the proposed method in dealing with pair-wise IR instances for 3D modeling of real objects. The well-know SAMPL repository of range images has been considered in this experimentation. In addition, the JAYA-based IR method has been systematically compared with 12 other analog IR algorithms in the SoTA, by means of using several datasets of range images spanning different levels of complexity, in particular 20 and 40 degrees of overlapping between images.

This manuscript is organized as follows. To begin with, Section II provides foundational insights into the IR challenges related to range data. Next, Section III presents a brief overview of several of those more recent applications of Nature-inspired and similar approaches addressing this problem. Moreover, a concise description of those IR methods using DL techniques are also introduced in this section. The proposed new design of the JAYA algorithm for tackling with the IR problem is accordingly introduced in Section IV. Section V conducts an empirical investigation, evaluating the new proposal of IR based on JAYA and comparing its performance against 12 other methods of the SoTA. Finally, Section VI offers conclusive comments on the findings of this work.

## II. BACKGROUND

### A. PROBLEM FORMULATION

In this section, the contextualized IR problem is introduced for the particular scenario of aligning 3D surfaces obtained by laser ranging devices. This imaging modality is known as range imaging ( [20]).

Specifically, range scanner devices possess the capability to capture three-dimensional images, termed range images, from diverse perspectives of the observed object. Each range image captures a portion of the overall geometry of the scanned object, providing the placement of each image in a distinct coordinate system. Consequently, the adoption of a reconstruction technique becomes imperative to seamlessly integrate the images, ensuring the attainment of a comprehensive and dependable representation of the physical object. This conceptual framework is commonly referred to as three-dimensional model reconstruction [20], where IR assumes a pivotal role.

Typically, two methodologies for reconstructing and integrating multiple range images exist [20]. The *cumulative* pair-wise approach entails the iterative application of the IR method, addressing pairs of range images at each step to reconstruct the original model. In contrast, the *global/multiview* IR stage, implement subsequent to the cumulative approach, aims to enhance the accuracy of the reconstruction outcome by simultaneously considering all pre-registered images. Irrespective of the chosen reconstruction approach (cumulative or multiview), the quality of the resulting model hinges significantly on the efficacy of the pair-wise range IR (RIR) algorithm utilized. Then, in the realm of 3D model reconstruction, the careful selection and optimization of the pair-wise RIR algorithm play a pivotal role in determining the ultimate quality of the integrated model.

There is no universally applicable blueprint for a hypothetical (pair-wise) IR method that could address all IR tasks, e.g. RIR, as the unique requirements of each application need to be carefully considered [2]. Nevertheless, IR methods typically involve the incorporation of four essential components (see Figure 1): two input **Images** denoted as Scene $I_s = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n\}$ and Model $I_m = \{\vec{p}\,'1, \vec{p}\,'2, \ldots, \vec{p}\,'m\}$, with $\vec{p}i$ and $\vec{p}_j{}'$ representing image points; a **Registration transformation** $f$, characterized as a parametric function establishing a relationship between the two images; a **Similarity metric function** $F$, utilized to quantify the qualitative closeness or degree of fitting between the transformed scene image, denoted as $f'(I_s)$, and the model image; and an **Optimizer** tasked with identifying the optimal transformation $f$ within the defined solution search space.

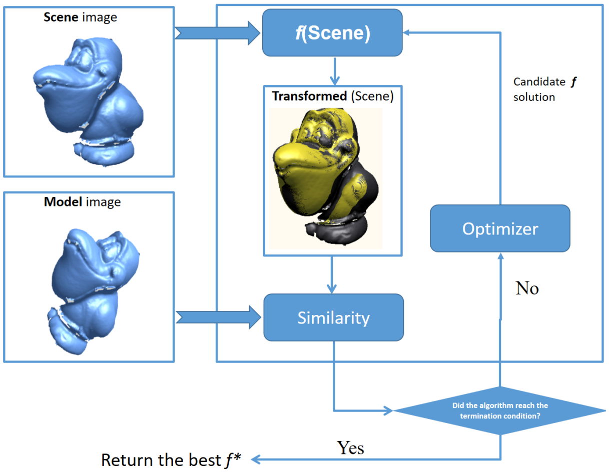

**FIGURE 1. A general description of how the IR optimization process works. Range images from the SAMPL dataset (see Section V) are shown in the diagram.**

Furthermore, in the intricate realm of IR, crafting an effective methodology involves a thoughtful consideration of the specific application requirements. The orchestration of components such as input images, registration transformations, similarity metric functions, and optimizers plays a crucial role in shaping a robust and tailored IR solution.

Similarly, an iterative procedure is commonly pursued until achieving convergence, typically within a tolerance threshold determined by the relevant similarity metric. This element will be addressed in greater depth in the following section. In particular, the ICP algorithm [21], [22] was initially introduced to attain a precise estimation of the rigid pose for pairs of range images through an optimization method grounded in the least squares estimation of the registration transformation $f$. Nevertheless, for the method to converge to a satisfactory alignment, it is essential that the relative rotation and translation of the pair of images remain small.

### B. SOFTCOMPUTING AND RIR

The field of Softcomputing [9] denotes a domain within computer science characterized by the utilization of imprecise solutions to computationally challenging tasks, notably those categorized as NP-complete problems where deriving a global optimal solution is not feasibly achieved within a (preferable) polynomial timeframe. Specifically, Softcomptuing diverges from traditional hard-computing (e.g. the approach of the canonical method for IR, ICP) in its capacity to accommodate imprecision, uncertainty, partial truth, and approximation. The fundamental tenet of Softcomputing lies in leveraging this tolerance for imprecision, uncertainty, partial truth, and approximation to attain manageability, resilience, and minimized solution costs. Key paradigms in Softcomputing encompass Fuzzy systems, Nature-inspired computation, and Artificial neural computing, among others.

Over the past two decades, computational strategies Inspired by Nature such as EC [13], [23], [24], have proven their efficacy in addressing intricate real-world challenges within the realm of CV. Furthermore, general-purpose Metaheuristics and other similar approaches constitute additional optimization algorithms successfully employed in this domain. In particular, several alternative EAs have been introduced in recent years, enhancing the SoTA of this domain by adopting more fitting optimization strategies [13]. These include evolution strategies (ES) [25], scatter search (SS) [26], differential evolution (DE) [27], [28], memetic algorithms (MAs) [29], particle swarm optimization (PSO) [30], estimation distribution algorithms (EDAs) [31], membrane computing (MC) [32], and cellular automata (CA) [33], and differential evolution (DE) [28]. Additionally, there has been a recent influx of EC models inspired by analogous principles, such as the bacterial foraging optimization algorithm (BFOA) [34], artificial bee colony (ABC) [35], harmony search algorithm (HS) [36], bat algorithm (BA) [37], firefly algorithm (FA) [38], and grasshopper optimization algorithm (GOA) [39], among many others.

The initial endeavors to confront the IR challenge using EC trace back to the 1980s [40], when a GA was devised to address the rigid IR of 2D angiography images. Since then, the domain of Softcomputing-based IR has flourished into a highly dynamic field, driven by the favorable outcomes achieved. Numerous well-established EAs have been enlisted to tackle the optimization process of IR. Comprehensive reviews of the application of this family of optimization techniques to IR dealing with 3D modeling and medical imaging scenarios can be found in [4], [6], [15].

As mentioned, the pipeline for reconstructing a 3D model involves executing multiple pair-wise alignments between two consecutive range images to generate the ultimate 3D model of the physical object [41]. Consequently, each pair-wise RIR method strives to identify the Euclidean motion that optimally aligns the *scene* view ($I_s$) with the *model* view ($I_m$). This Euclidean motion is typically characterized by a 3D rigid transformation ($f$) defined by six or seven real-coded parameters, depending on whether Euler or axis-plus-angle representation is employed for rotation, respectively. In this research, it has been considered the rigid transformation as comprising a rotation $R = (\theta,\ Axis_x, Axis_y, Axis_z)$ and a translation $\vec{t} = (t_x, t_y, t_z)$, where $\theta$ and $\vec{Axis}$ denote the angle and axis of rotation, respectively. The transformed points of the *scene* view are identified as

$$f(\vec{p}_i) = R(\vec{p}_i) + \vec{t},\ \ i = \{1, \ldots, n\} \tag{1}$$

where, $n$ represents the quantity of points in the $I_s$ image. Consequently, the evolutionary pair-wise RIR process can be cast as a numerical optimization challenge (where RIR solutions manifest as seven-dimensional real-coded vectors $x = \langle \theta, Axis_x, Axis_y, Axis_z, t_x, t_y, t_z \rangle$) designed to explore the Euclidean transformation $f^*$ that achieves the optimal alignment of both $f(I_s)$ and $I_m$:

$$f^* = \arg\min_{f} \ F(I_s, I_m; f)\ \ s.t.: \ f^*(I_s) \cong I_m \tag{2}$$

in accordance with the optimization of the Similarity metric, $F$. Notably, the median square error (MedSE) is commonly adopted as the $F$ function in 3D modeling [42], owing to its resilience in the face of outliers (e.g., noisy range images obtained during the RIR process). Its formulation is as follows:

$$F(I_s, I_m; f) = MedSE(d_i^2),\ \ \forall i = \{1, \ldots, n\} \tag{3}$$

wherein $MedSE()$ represents the median value derived from all the squared Euclidean distances, $d_i^2 = \|f(\vec{p}_i) - \vec{p}_j'\|^2$ ($j = \{1, \ldots, m\}$), between the transformed scene point, $f(\vec{p}_i)$, and its corresponding nearest point, $\vec{p}_j'$, in the *model* view $I_m$. It is noteworthy that both the $F$ function and either the fitness or the objective function (see Section III) bear identical significance in the optimization process.

Ultimately, to accelerate the calculation of the closest/nearest point for each $f(\vec{p}_i)$ point, sophisticated indexing structures like kd-trees [43] or the grid closest point (GCP) transform [44] are commonly used employed. Specifically, this work makes use of these structures in the experimental section.

## III. BRIEF OVERVIEW

Following the inception of the initial ICP algorithm, numerous additions have been put forth to enhance its capabilities [8], [45]. Since the initial endeavors to address the IR problem using Softcomputing-based solutions, this subject has evolved into a highly active realm of research, marked by a substantial influx of proposals. However, there is a pressing need for an updated examination of the SoTA owing to the emergence of new and innovative enhancements in the past recent years.

Then, this section is devoted to present a brief revision of the most pertinent IR methods within the field. Refer to the subsequent contributions for a more in-depth overview conducted up to the year 2021 [4]–[6], [15]. In addition, new emerging approaches within the field of DL is accordingly introduced below.

### A. EVOLUTIONARY AND METAHEURISTC-BASED APPROACH

In [46], the authors introduced a novel IR approach by hybridizing a self-adaptive ES algorithm with an accelerated PSO algorithm, denoted as ES-APSO. This recent innovation was specifically crafted for the purpose of signature recognition. Notably, their PSO strategy draws inspiration from the Firefly Optimization (FiFO) [47] algorithm, which mimics the flashing behavior of fireflies. Essentially, FiFO adheres to three idealized rules. Firstly, fireflies exhibit unisex attraction, meaning one firefly is drawn to others irrespective of their gender. The second rule posits that attractiveness is proportionate to brightness, with both diminishing as distance increases. The third rule pertains to the brightness of a firefly, determined by the landscape of the objective function (i.e., $F$). ES-APSO underwent extensive testing across a diverse set of experiments using various 2D images representing signatures. The results were subsequently compared against several canonical versions of FiFO. The experimental outcomes underscored the efficacy of ES-APSO in addressing instances of signature recognition. In [48], the authors present an improvement to monomodal IR of medical images, introducing the integration of the Cuckoo Search (CS) method for generating Lévy flights. This modification and optimization are specifically tailored for application in MRIs, with a primary focus on brain cancer detection. The effectiveness of the proposed monomodal IR with CS algorithm was evaluated through a comparative analysis with conventional monomodal IR approaches. In [49], the authors present a novel technique for IR under geometric perturbations, encompassing rotations, translations, and non-uniform scaling. The input images, whether monochrome or colored, undergo preprocessing using a noise-resistant edge detector to yield binarized versions. Their method leverages the efficiency of computations conducted in reduced representations by adopting a memetic approach ( [50]), yielding rapid and promising initial solutions. As stated by the authors, the proposed method amalgamates bio-inspired and EC techniques with clustered search, implementing a procedure specifically designed to tackle the premature convergence issue across various scaled representations.

### B. DEEP LEARNING-BASED APPROACH

As mentioned earlier, the primary constraint of ICP lies in its reliance on a nearest neighbor-based correspondence strategy, which is greatly influenced by the initialization. Present DL-based approaches [17] employ two categories of solvers, contingent on whether they depend on explicit correspondences [51].

Hence, in this scenario, DL is implemented to enhance both the extraction and matching of features. On the one hand, DCP [52] integrates DGCNN [53] for embedding point clouds and incorporates an attention-based module for matching features. This is followed by the application of a differentiable SVD solver within an end-to-end architecture for IR. On the other hand, DGR [54] utilizes fully convolutional geometric features for feature extraction and employs a 6-dimensional segmentation for predicting correspondences. In line with the previous ones, DeepGMR [55] learns to identify pose-invariant correspondences between Gaussian mixture models that approximate the shape. Subsequently, it computes the transformation based on the model parameters, enabling the suppression of noise in the point cloud. The primary limitation of DL-based IR methods based on correspondence lies in the robustness of the SVD solver. This enforces feature matching to be nearly free of outliers, a challenge that proves difficult to overcome.

In summary, while this emergent IR approach, i.e. DL-based IR, shows encouraging performance, its resilience relies significantly on the utilization of proficient differentiable solvers, coupled with the application of correspondences defined heuristically. Ultimately, techniques centered around this approach frequently find themselves ensnared in local minima.

## IV. JAYA-BASED RIR PROPOSAL

It is widely acknowledged that conventional or classical optimization methods come with certain constraints when addressing intricate optimization challenges. These limitations are predominantly linked to the inherent search mechanisms embedded in these traditional approaches. To mitigate some of the shortcomings associated with classical optimization procedures, researchers have introduced Softcomputing-based optimization techniques, e.g. EAs and Metaheuristics, alternatively known as advanced optimization techniques, primarily stemming from artificial intelligence research. These paradigm of algorithms exhibit problem- and model-independent characteristics, with the majority being both efficient and adaptable. The field of research dedicated to these techniques remains highly dynamic, witnessing a continuous influx of novel EC and Metaheuristic-based enhanced algorithms [11], [13], [56].

In the last few years, Dr. Rao proposed a novel optimization algorithm called JAYA [18], devoid of any metaphorical foundation. The algorithm constantly strives to get closer to success, aiming for the optimal solution, while actively avoiding failure, which involves avoiding convergence with respect to the least favorable outcome. Its overall goal is the triumphant achievement of the best solution, which inspires its nomenclature: JAYA, a Sanskrit term meaning triumph or victory. Designed specifically for global optimization challenges, this algorithm is versatile in addressing both continuous and discrete optimization problems, covering single, multiple or numerous objectives.

Following the introduction of the JAYA algorithm, various adaptations and modifications have been introduced to enhance its applicability in the field [19]. The utilization of the JAYA algorithm and its derivatives across diverse engineering and scientific domains is also relevant. Researchers may observe that, aside from its simplicity and efficacy, the JAYA algorithm operates without the necessity for algorithm-specific (i.e. control) parameters, thereby alleviating the drawbacks associated with numerous advanced optimization algorithms that require meticulous tuning of such parameters. As emphasized in this work, improper adjustment of these algorithm-specific parameters can lead to near-optimal results or confinement to local optima.

A handful of methods incorporating a self-tuning methodology of the control parameters have been contributed to the SoTA [57]–[59], and they are acknowledged as the most advanced RIR methods currently available. Next subsections are devoted to introduce the specif design of an enhanced RIR method using a fully self-tuned approach.

### A. RIR OPTIMIZATION BY JAYA

Unlike the dual phases under consideration (i.e., the teacher phase and the learner phase) in the TLBO algorithm [60], JAYA operates with a single phase (i.e. teacher phase). Its concept is straightforward, demonstrating superior performance when compared to alternative optimization algorithms. This algorithm proves effective in attaining global solutions for both continuous and discrete optimization problems, requiring minimal computational effort while maintaining high consistency.

JAYA is easy to develop and eliminates the need for adjusting algorithm-specific parameters. $P$ initial solutions are randomly generated within the specified upper and lower bounds of the process variables. Subsequently, each variable in every solution undergoes a stochastic update as defined by Eq. 4. Consider $F$ as the objective function to minimize (or maximize), and let there be $d$ design variables. The objective function value for the best solution is denoted as $F_{best}$, while the objective function value for the worst solution is denoted as $F_{worst}$ (see pseudo-code in Figure 3 for the JAYA-based RIR method).

$$\begin{aligned} P(i+1,j,k) &= P(i,j,k) \\ &+r(i,j,1)(P(i,j,b) - |P(i,j,k)|) \\ &-r(i,j,2)(P(i,j,w) - |P(i,j,k)|) \end{aligned} \tag{4}$$

Here, $b$ and $w$ denote the indices of the best and worst solutions within the current population $P$ (see Figure 2). The variables $i, j, k$ represent the indices of the iteration, design variable, and candidate solution, respectively. $P(i,j,k)$ denotes the $j$-th variable of the $k$-th candidate solution in the $i$-

th iteration. The numbers $r(i,j,1)$ and $r(i,j,2)$ are randomly generated within the range of $[0,1]$, serving as scaling factors to ensure effective diversification. JAYA's primary objective is to enhance the objective function, i.e. $F$, for each candidate solution in the population.

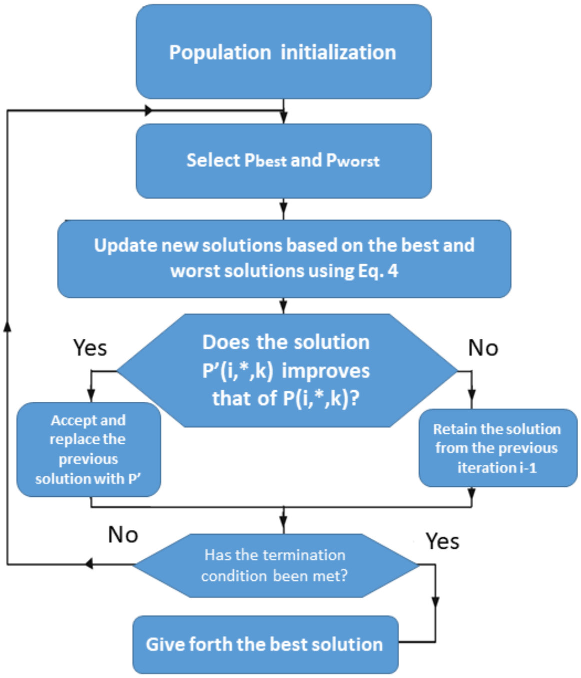
Population initialization
Select Pbest and Pworst
Update new solutions based on the best and worst solutions using Eq. 4
Yes
Does the solution P'(i,*,k) improves that of P(i,*,k)?
No
Accept and replace the previous solution with P'
Retain the solution from the previous iteration i-1
No
Has the termination condition been met?
Yes
Give forth the best solution

**FIGURE 2.** A general flowchart of JAYA.

Consequently, the algorithm seeks to shift the $F$ value of each solution towards the best solution, $F_{best}$, by updating variable values. Following the updates, the new solutions are compared to their corresponding old solutions, and only superior solutions (those with improved $F$ values) are retained for the next generation. Thus, with each generation, a solution moves closer to the best solution, while simultaneously distancing itself from the worst solution, $F_{worst}$. This achieves a balanced trade-off between intensification and diversification in the search process. The algorithm consistently endeavors to move closer to success (i.e., reaching the best solution) and avoids failure (i.e., moving away from the worst solution). Its goal is triumphant attainment of the best solution, denoted as $F_{best}$.

Unlike prior contributions in RIR that employed a self-tuned approach [57]–[59], the JAYA-based RIR introduced in this work incorporates self-adaptation for the population size. Consequently, this novel proposal can be characterized as the first fully self-tuned RIR method in the field. The subsequent section is dedicated to introducing this specific adaptation strategy.

### B. SELF-TUNING THE POPULATION SIZE

The relevant aspect is that the JAYA algorithm, being self-adaptive, automatically tunes the population size [61]. Consequently, users are relieved from this task. Assuming the initial population is random and equals $(10*d)$, where $d$ represents the number of design variables, the formulation for the new population is as follows:

```
Begin JAYA-Optimizer

$i \leftarrow 1$; $i$ stands for number of iterations
$m \leftarrow (10*d)$; $m$ stands for $P$ size

Initialize($P, m$)

Do
    $b \leftarrow$ GetBestSolution($P, m$)
    $w \leftarrow$ GetWorstSolution($P, m$)

    For ($k = 1$ to $m$) Do
        $P'(i,*,k) \leftarrow P(i,*,k)$
        $+r(i,*,1)(P(i,*,b) - |P(i,*,k)|)$
        $-r(i,*,2)(P(i,*,w) - |P(i,*,k)|)$

        If (F($P'(i,*,k)$) is better than F($P(i,*,k)$)) Then
            $P(i,*,k) \leftarrow P'(i,*,k)$
        End-If
    End-For

    $i \leftarrow i+1$
While Not(reach-termination)
Return $P_b$

End JAYA-Optimizer
```

**FIGURE 3.** The pseudo-code of the JAYA-based RIR method.

$$m_{new} = round(m_{old} + r * m_{old}) \tag{5}$$

where, $r$ denotes a randomly generated value within the range of $[-0.5, 0.5]$. The population size may either decrease or increase based on the negative or positive nature of the random value $r$ (see pseudo-code in Figure4 for the fully self-tuned JAYA-based RIR method). For instance, if the current population size $m_{old}$ is 70 (i.e. $(10*7)$, being 7 the number of design variables for the RIR problem) and the randomly generated number $r$ is $-0.25$, the ensuing population size for the subsequent iteration $m_{new}$ will be 53 (see Eq. 5), which means a decrease of the population size.

Additionally, elitism is incorporated when the population size of the new population is higher than the population size of the old population ($m_{new} > m_{old}$). In this scenario, the entire existing old population is carried over to the new one, and the best optimal solutions from the current population are allocated to the remaining $m_{new} - m_{old}$ solutions. Conversely, if the population size of the new population is smaller than that of the old population ($m_{new} < m_{old}$), only the best population from the old one is transferred to the new population. If the population size decreases and falls below the count of design variables $d$, it is adjusted to be equal to the number of design variables (i.e., $m_{new} = d$ when $m_{new} < d$). This strategy promotes a suitable trade-off between intensification and

```
Begin Self-adaptive JAYA-Optimizer

$i \leftarrow 1$; $i$ stands for number of iterations
$\{m_{old}, m_{new}\} \leftarrow (10 * d)$; $m$ stands for $P$ size

Initialize($P, m_{old}$)

Do
    $b \leftarrow$ GetBestSolution($P, m_{old}$)
    $w \leftarrow$ GetWorstSolution($P, m_{old}$)

    For ($k = 1$ to $m_{old}$) Do
        $P'(i,*,k) \leftarrow P(i,*,k)$
        $+r(i,*,1)(P(i,*,b) - |P(i,*,k)|)$
        $-r(i,*,2)(P(i,*,w) - |P(i,*,k)|)$

        If (F($P'(i,*,k)$) is better than F($P(i,*,k)$)) Then
            $P(i,*,k) \leftarrow P'(i,*,k)$
        End-If
    End-For

    $m_{new} \leftarrow round(m_{old} + r * m_{old})$

    $P \leftarrow Elitism(P, m_{new}, m_{old})$

    $m_{old} \leftarrow m_{new}$
    $i \leftarrow i + 1$
While Not(reach-termination)

Return $P_b$

End Self-adaptive JAYA-Optimizer
```

**FIGURE 4. The pseudo-code of the fully self-tuned JAYA-based RIR method.**

diversification of the search, and ensures that the solutions do not get trapped in local optima.

### C. ADAPTIVE LOCAL-SEARCH

Crossover-based local-search (XLS) [62] is a singular class of optimization methods that are especially attractive for real-coding problems. Indeed, they consider crossover operators that have a self-adaptive nature. Such operators can generate trial solutions according to the distribution of the parents solutions in the population without any adaptive parameters. In particular, in [63] the author shown that the unimodal normal distribution crossover (UNDX) operator can provide self-adaptive behavior on a number of real-coded problems, and it has been applied successfully to real and challenging optimization problems. The fundamental idea of XLS is to induce an LS in the vicinity of specific solutions involved to improve the exploitation capabilities of the host global search algorithm.

In the specific design of the fully self-adaptive JAYA-based RIR method (see pseudo-code in Figure 4), the proposed UNDX-based XLS approach is designed as follows: the best solution is considered to be improved, called family father (i.e. $P_b$), only in case it was updated in the current i-*th* iteration. $T$ solutions in the current population $P$ are randomly selected to pair with the $P_b$ solution to create new trial solutions in its neighborhood through the UNDX operator ( [63]). Finally, a choice is made to replace the father $P_b$ solution with a new created one (from the $T$ solutions generated by the UNDX operator) in case it is better than the former ( [64]). Additionally, this specific design for the UNDX-based XLS considers that the three input solutions $X_1, X_2,$ *and* $X_3$ that will be fed to the UNDX operator (see [63]) refer to the best $(P_b)$ , worst $(P_w)$ and a randomly chosen solution from the i-*th* current iteration of $P$, respectively.

Finally, to make the XLS fully adaptive, the value of $T$ is dynamically adjusted equivalently as done in Eq. 5. Specifically, Eq. 6 shows the value of $T$ for the i-*th* iteration in which it is decided to apply the UNDX-based XLS mechanism.

$$T = round(\tfrac{1}{3} \cdot m_{new}) \tag{6}$$

where $m_{new}$ is the number of solutions in the i-*th* population; in case the value of $T$ is less than 3, then $T = 3$ new trial solutions are generated within the UNDX-based XLS strategy.

## V. COMPUTATIONAL STUDY

The purpose of this section is to present a series of experiments in order to study the results obtained through the proposed fully self-tuned JAYA RIR method. In addition, its performance will be compared against to that obtained by those methods proposed to date in the SoTA, also making use of similar optimization approaches: SaEvO [58],MA [50], DE [65], GA [43], PSO [66], GA [45], ABC [67], StEvO [57],BBO [67], GA [44], HS [67], GA [68].

The proposed JAYA-based RIR algorithm was implemented in C++ and compiled with the GNU/g++ toolkit. All the tested methods have been adapted by using the same representation of the rigid transformation $(f)$ and objective function (see Eq.(3) in Section II-B).

In order to perform a fair comparison among the RIR methods, it has been considered for all tested methods the same stop criterion previously used in the experimental sections of [4], [58], [67], i.e. the run (CPU) time. Specifically, the 20 seconds established in those contributions have been maintained in this work. Moreover, both the same control parameters for the tested methods and the running hardware have been maintained in this work, i.e. using an Intel Pentium IV 2.6 MHz processor and 2GB RAM. Note that the proposed JAYA-based RIR method does not need any adjustment of the control parameters.

### A. RANGE IMAGE DATASETS

In order to ease the comparison with the results reported in other contributions in the field [4], [58], [67], the experiments correspond to a number of pair-wise RIR problem instances using different range datasets obtained from the well-known public repository of the *Signal Analysis and Machine Perception Lab* (SAMPL). Figure 5 shows the six range image datasets considered, named as in previous contributions [43]:

"Bird", "Frog", "Tele", "Angel", "Buddha", and "Lobster". The datasets range from eight to fifteen thousand points size.

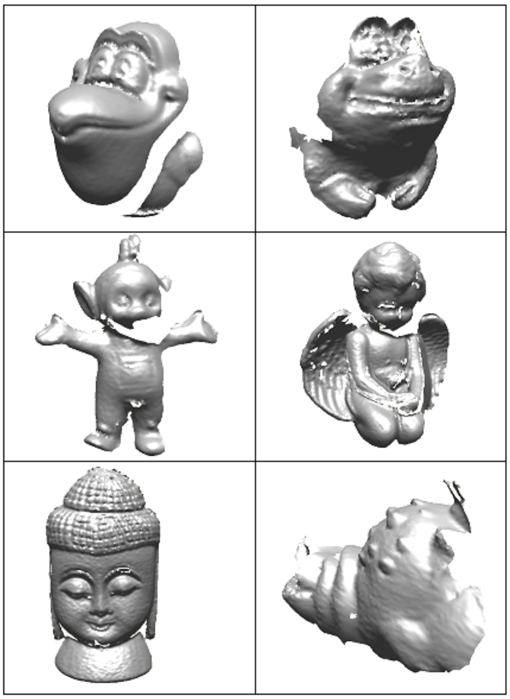

**FIGURE 5. Range image dataset from the SAMPL repository. From left to right and from top to bottom: "Bird", "Frog", "Tele", "Angel", "Buddha", and "Lobster" images.**

Concerning the RIR problem scenarios, 20 and 40 overlapping degrees of the turn table have been commonly used in the literature for testing RIR proposals [3], [4]. As stated, the lower the rotation degree of the turn table is, the more overlapped the adjacent images and the easier the RIR problem will be (see Section II-A). Then, that scenario considering 40 degree overlap is considered the most complex and challenging one.

Finally, it has been used both the GCP structure (see Section II-B) and the subsampled version (5K randomly sampled version) of each range image in order to speed up the computation of the objective function (see Eq.(3) in Section II-B).

### B. EXPERIMENTAL SETUP

To avoid execution dependency, 30 different runs were performed on each of the 13 RIR algorithms before the alignment tested, taking into account image overlap of 20 and 40 degrees, respectively (see section V-A).

All tested algorithms start with an initial set of random solutions. Then, in each run, a randomly generated rigid transformation is applied using an uniform distributed scene image $f_r(I_s)$ and using the RIR method to find the optimal transformation $f^*$ between the transformed escene image $f_r(I_s)$ and the model image $I_m$. Each rigid transformation is randomly generated as follows: each of the 3 rotation axis parameters is in the range $[-1, 1]$, the angle of rotation is in the range $[0°, 360°]$, and the range of the 3 translation parameters is $[-40mm, 40mm]$.

### C. ANALYSIS OF RESULTS

Tables 1 and 2 show statistical results of the minimized objective function $F$ (see Eq.(3)) corresponding to the 30 runs carried out by each of the 13 RIR methods when facing the 2 RIR problem scenarios, i.e. 20 and 40 degrees overlap. Specifically, each column of these tables refer to the range dataset, the RIR method, and the minimum, maximum, mean, and standard deviation values. Furthermore, the last two columns, i.e., $Q^A$ and $Q^B$, rank each method based on its quartile position with respect to the minimum and mean values, respectively. Then, the degree of accuracy $Q^A$ and robustness $Q^B$ of each method can be better assessed.

In particular, a subset of methods with the best performance achieved in the 20 degree scenario and 4 range datasets have been selected (see Table 1), i.e. the 7 methods with the best rating according to $Q^A$ and $Q^B$. Table 2 shows the performance achieved by these 7 RIR methods addressing the most complex scenario with 40 degrees and 6 range datasets.

Figures 6 and 7 graphically show the performance offered by each of the 13 analyzed methods depending on the position they occupy according to accuracy and robustness, $Q^A$ and $Q^B$, respectively. Figure 8 shows the degree of robustness of the 7 best RIR methods facing the 10 problem instances designed in this experimentation, i.e. 4 and 6 for the scenarios with 20 and 40 degrees of overlapping, respectively. Specifically, each bar in the graph shows the number of times in which the considered method has reached the best mean value among the ten RIR problem instances.

As can be seen from the results in Table 1 for the 20 degree scenario, the RIR methods that offer adaptive capabilities of the control parameters are those that offer the best results: JAYA, StEvO, and SaEvO. Also, the memetic MA proposal and the developments based on ABC and HS achieve very competitive performances, although lower than the adaptive ones. It can be seen how the methods with the best performance in all cases are JAYA, StEvO, MA, and SaEvO. The common denominator in the design of the majority of these best algorithms is that a specific mechanism to exploit exploitative capabilities is incorporated, i.e. a local search. Therefore, this design aspect is of special importance in obtaining a better performance. This same behavior is also reflected in the results shown in Table 2 corresponding to the scenario with 40 degrees of overlapping.

Finally, if the results are analyzed globally taking into account the ten RIR problem instances addressed in the two scenarios considering variable complexity, it can be highlighted that the proposed design based on JAYA is the RIR method that offers the best performance in terms of robustness, this being one of the main objectives to be analyzed in this work. Specifically, JAYA behaves as the most robust method in 50 percent of the cases, compared to 30 and 20 percent for StEvO and SaEvO, respectively.

**TABLE 1.** Pre-alignment RIR results of the 20 degree and 4 range datasets problem scenario. The unit length is squared millimeters. The best minimum and mean values are underlined. Also, accuracy and robustness are shown according to quartile positions, $Q^A$ and $Q^B$, respectively.

| Dataset | RIR method | Min. | Max. | Mean | S.D. | $Q^A$ | $Q^B$ |
|---|---|---|---|---|---|---|---|
| *Angel* $_{20^o}$ | SaEvO [58] | **0.2447** | 0.9441 | 0.3499 | 0.2119 | 1 | 1 |
| | MA [50] | 0.2448 | 0.9453 | 0.3185 | 0.1461 | 1 | 1 |
| | DE [65] | 0.2493 | 0.9462 | 0.6732 | 0.2209 | 2 | 3 |
| | GA [43] | 0.2495 | 0.9555 | 0.4179 | 0.2560 | 2 | 2 |
| | PSO [66] | 0.4801 | 0.9814 | 0.9108 | 0.1013 | 4 | 4 |
| | GA [45] | 0.4731 | 0.9733 | 0.8601 | 0.1236 | 3 | 4 |
| | ABC [67] | 0.2470 | 0.5289 | 0.3319 | 0.1007 | 2 | 2 |
| | StEvO [57] | 0.2448 | 0.5268 | **0.2947** | 0.0886 | 1 | 1 |
| | BBO [67] | 0.2576 | 0.9554 | 0.4975 | 0.2692 | 3 | 3 |
| | GA [44] | 0.2553 | 0.9531 | 0.5818 | 0.2792 | 3 | 3 |
| | HS [67] | 0.2494 | 0.9535 | 0.4412 | 0.2724 | 2 | 2 |
| | GA [68] | 0.9092 | 0.9805 | 0.9651 | 0.0151 | 4 | 4 |
| | JAYA | 0.2463 | 0.3638 | 0.3076 | 0.0298 | 4 | 1 |
| *Bird* $_{20^o}$ | SaEvO [58] | **0.1124** | 0.5998 | 0.1812 | 0.1572 | 1 | 1 |
| | MA [50] | 0.1132 | 0.8881 | 0.2075 | 0.2015 | 2 | 2 |
| | DE [65] | 0.1245 | 0.8429 | 0.4793 | 0.2157 | 4 | 3 |
| | GA [43] | 0.1152 | 0.9178 | 0.3506 | 0.3112 | 2 | 2 |
| | PSO [66] | 0.4337 | 0.9613 | 0.8638 | 0.1180 | 4 | 4 |
| | GA [45] | 0.3125 | 0.9361 | 0.7153 | 0.1678 | 4 | 4 |
| | ABC [67] | 0.1167 | 0.9009 | 0.3107 | 0.2451 | 2 | 2 |
| | StEvO [57] | 0.1125 | 0.5977 | 0.1814 | 0.1569 | 1 | 1 |
| | BBO [67] | 0.1263 | 0.9301 | 0.4347 | 0.2759 | 3 | 2 |
| | GA [44] | 0.1199 | 0.9180 | 0.4465 | 0.2725 | 3 | 3 |
| | HS [67] | 0.1170 | 0.9188 | 0.4671 | 0.3603 | 2 | 3 |
| | GA [68] | 0.8577 | 0.9545 | 0.9335 | 0.0214 | 4 | 4 |
| | JAYA | 0.1128 | 0.6543 | **0.1782** | 0.1268 | 1 | 1 |
| *Frog* $_{20^o}$ | SaEvO [58] | **0.1189** | 0.5296 | 0.1789 | 0.1337 | 1 | 1 |
| | MA [50] | 0.1194 | 0.8120 | 0.2029 | 0.1756 | 1 | 2 |
| | DE [65] | 0.1322 | 0.7345 | 0.4374 | 0.1615 | 3 | 3 |
| | GA [43] | 0.1249 | 0.8555 | 0.4329 | 0.2415 | 2 | 2 |
| | PSO [66] | 0.3633 | 0.9407 | 0.7792 | 0.1448 | 4 | 4 |
| | GA [45] | 0.4342 | 0.9044 | 0.6972 | 0.1353 | 4 | 4 |
| | ABC [67] | 0.1226 | 0.7733 | 0.2437 | 0.1798 | 2 | 2 |
| | StEvO [57] | 0.1193 | 0.5308 | 0.1792 | 0.1337 | 1 | 1 |
| | BBO [67] | 0.1649 | 0.8690 | 0.5396 | 0.2023 | 3 | 3 |
| | GA [44] | 0.1234 | 0.8311 | 0.5119 | 0.2162 | 2 | 3 |
| | HS [67] | 0.1260 | 0.8751 | 0.3476 | 0.2749 | 3 | 2 |
| | GA [68] | 0.8147 | 0.9331 | 0.8895 | 0.0347 | 4 | 4 |
| | JAYA | 0.1195 | 0.5765 | **0.1776** | 0.0987 | 2 | 1 |
| *Tele* $_{20^o}$ | SaEvO [58] | 0.0735 | 0.1071 | **0.0780** | 0.0080 | 1 | 1 |
| | MA [50] | 0.0736 | 0.7867 | 0.1639 | 0.2192 | 2 | 2 |
| | DE [65] | 0.0755 | 0.6578 | 0.3193 | 0.1819 | 2 | 3 |
| | GA [43] | 0.0750 | 0.9234 | 0.3728 | 0.3366 | 2 | 3 |
| | PSO [66] | 0.2510 | 0.9319 | 0.7801 | 0.1622 | 4 | 4 |
| | GA [45] | 0.3374 | 0.9161 | 0.7477 | 0.1472 | 4 | 4 |
| | ABC [67] | 0.0752 | 0.8691 | 0.1501 | 0.1817 | 2 | 1 |
| | StEvO [57] | 0.0735 | 0.8647 | 0.1044 | 0.1414 | 1 | 1 |
| | BBO [67] | 0.0829 | 0.8699 | 0.251 | 0.2292 | 3 | 2 |
| | GA [44] | 0.0791 | 0.8958 | 0.3159 | 0.2531 | 3 | 3 |
| | HS [67] | 0.0754 | 0.8721 | 0.2350 | 0.2963 | 2 | 2 |
| | GA [68] | 0.7509 | 0.9519 | 0.9132 | 0.0438 | 2 | 4 |
| | JAYA | **0.0721** | 0.5209 | 0.2267 | 0.1318 | 1 | 2 |

**TABLE 2.** Pre-alignment RIR results of the 40 degree and 6 range datasets problem scenario. The unit length is squared millimeters. The best minimum and mean values are underlined. Also, accuracy and robustness are shown according to quartile positions, $Q^A$ and $Q^B$, respectively.

| Dataset | RIR method | Min. | Max. | Mean | S.D. | $Q^A$ | $Q^B$ |
|---|---|---|---|---|---|---|---|
| *Angel* $_{40^o}$ | SaEvO [58] | **0.3493** | 0.9440 | **0.4983** | 0.2175 | 1 | 1 |
| | MA [50] | 0.3498 | 0.9539 | 0.5271 | 0.2467 | 2 | 2 |
| | GA [43] | 0.3527 | 0.9711 | 0.6790 | 0.2640 | 3 | 4 |
| | ABC [67] | 0.3542 | 0.9098 | 0.5265 | 0.2210 | 3 | 2 |
| | StEvO [57] | 0.3493 | 0.9436 | 0.4990 | 0.2175 | 1 | 1 |
| | HS [67] | 0.3553 | 0.9567 | 0.6460 | 0.2665 | 4 | 3 |
| | JAYA | 0.3503 | 0.9509 | 0.5207 | 0.2001 | 2 | 2 |
| *Bird* $_{40^o}$ | SaEvO [58] | **0.2028** | 0.9269 | 0.4451 | 0.3052 | 1 | 1 |
| | MA [50] | 0.2052 | 0.9373 | 0.4626 | 0.3175 | 2 | 2 |
| | GA [43] | 0.2159 | 0.9425 | 0.5795 | 0.3158 | 3 | 3 |
| | ABC [67] | 0.2124 | 0.9308 | 0.5072 | 0.2829 | 3 | 3 |
| | StEvO [57] | 0.2041 | 0.9168 | **0.3741** | 0.2655 | 1 | 1 |
| | HS [67] | 0.2165 | 0.9430 | 0.6151 | 0.3058 | 4 | 4 |
| | JAYA | 0.2057 | 0.9308 | 0.4412 | 0.2596 | 2 | 1 |
| *Buddha* $_{40^o}$ | SaEvO [58] | 0.3990 | 0.9032 | 0.6120 | 0.1224 | 2 | 2 |
| | MA [50] | 0.3978 | 0.7524 | 0.6300 | 0.1020 | 1 | 2 |
| | GA [43] | 0.5075 | 0.9506 | 0.7146 | 0.1126 | 3 | 3 |
| | ABC [67] | 0.4473 | 0.9446 | 0.6690 | 0.1220 | 3 | 4 |
| | StEvO [57] | 0.3996 | 0.6873 | 0.5730 | 0.1103 | 2 | 1 |
| | HS [67] | 0.5526 | 0.9285 | 0.7147 | 0.1044 | 4 | 4 |
| | JAYA | **0.3948** | 0.7318 | **0.5516** | 0.1097 | 1 | 1 |
| *Frog* $_{40^o}$ | SaEvO [58] | 0.2536 | 0.7725 | 0.3991 | 0.1963 | 2 | 1 |
| | MA [50] | 0.2548 | 0.7812 | 0.4700 | 0.2271 | 2 | 2 |
| | GA [43] | 0.2735 | 0.9474 | 0.6923 | 0.1750 | 3 | 3 |
| | ABC [67] | 0.2717 | 0.8410 | 0.5512 | 0.2015 | 3 | 3 |
| | StEvO [57] | **0.2517** | 0.7717 | 0.3941 | 0.1856 | 1 | 1 |
| | HS [67] | 0.4026 | 0.9005 | 0.7403 | 0.1161 | 4 | 4 |
| | JAYA | 0.2521 | 0.8149 | **0.3870** | 0.1667 | 1 | 1 |
| *Lobster* $_{40^o}$ | SaEvO [58] | 0.2505 | 0.7582 | 0.3787 | 0.1894 | 1 | 1 |
| | MA [50] | **0.2490** | 0.8056 | 0.4369 | 0.2231 | 1 | 2 |
| | GA [43] | 0.2665 | 0.9201 | 0.5727 | 0.2089 | 3 | 3 |
| | ABC [67] | 0.2745 | 0.8220 | 0.6249 | 0.1530 | 4 | 4 |
| | StEvO [57] | 0.2522 | 0.8013 | 0.3816 | 0.1916 | 2 | 2 |
| | HS [67] | 0.2665 | 0.9257 | 0.5890 | 0.1964 | 3 | 3 |
| | JAYA | 0.2562 | 0.7004 | **0.3744** | 0.1533 | 2 | 1 |
| *Tele* $_{40^o}$ | SaEvO [58] | **0.1050** | 0.8062 | 0.1911 | 0.1667 | 1 | 1 |
| | MA [50] | 0.1062 | 0.8354 | 0.2217 | 0.2116 | 2 | 2 |
| | GA [43] | 0.1077 | 0.8950 | 0.5354 | 0.2929 | 3 | 4 |
| | ABC [67] | 0.1082 | 0.8607 | 0.2700 | 0.2222 | 3 | 4 |
| | StEvO [57] | 0.1054 | 0.4708 | **0.1682** | 0.1226 | 1 | 1 |
| | HS [67] | 0.1095 | 0.9222 | 0.4129 | 0.3072 | 4 | 3 |
| | JAYA | 0.1057 | 0.728 | 0.2512 | 0.1915 | 2 | 2 |

## VI. CONCLUSIONS

Image registration has demonstrated to be a very active research area in the last decade, especially when considering new optimization alternatives as those from the soft-computing field. In contrast to traditional image registration methods such as the ICP algorithm, this emerging paradigm of optimization approaches (i.e. metaheuristics, evolutionary algorithms, swarm intelligence, etc) provide a more interesting view-point due to they do not require a good initial estimation of starting solutions and they avoid to be trapped

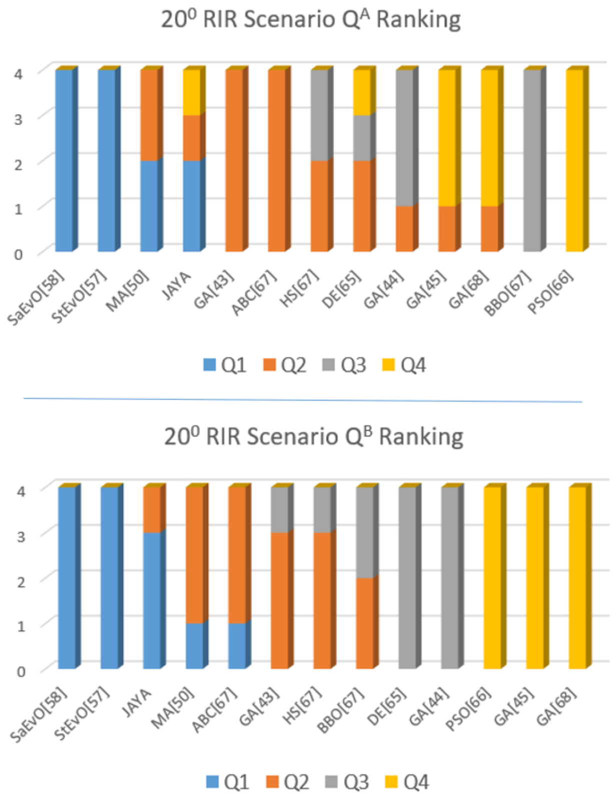

**FIGURE 6. Statistical results of the tested RIR methods when facing the 20 degree overlap RIR scenario. RIR methods are sorted from left (best) to right (worst) according to their best performance considering the quartile-based performance scores $Q^A$ and $Q^B$ in Table 1.**

in local optima. However one of the main shortcomings of this promising algorithms is that they must be carefully tuned in order to achieve their best performance for each problem addressed.

In this work, a brief updated review of the state-of-the-art methods originating from both the previous paradigm and others of recent appearance, such as the one based on deep learning tools, has been carried out. Next, the progress in the registration problem has been delved into using the optimization techniques that have offered the best performance to date, i.e. those based on the softcomputing paradigm. Due to the promising results achieved by methods of this family with self-adaptive capabilities of the control parameters, this work has taken a step further in the proposal of a novel method based on the JAYA algorithm, which does not require the adjustment of no control parameters. Then, it can be said that, as far as is known, this is the only image registration method that does not require any prior adjustment.

An extensive experimentation has been designed for the adequate comparison of the proposed method based on the JAYA algorithm against 12 RIR methods within the same category, all of them from the state of the art. Likewise, 10 range image registration problem instances that make up two scenarios of increasing complexity have been designed. In particular, range image datasets from one of the most used

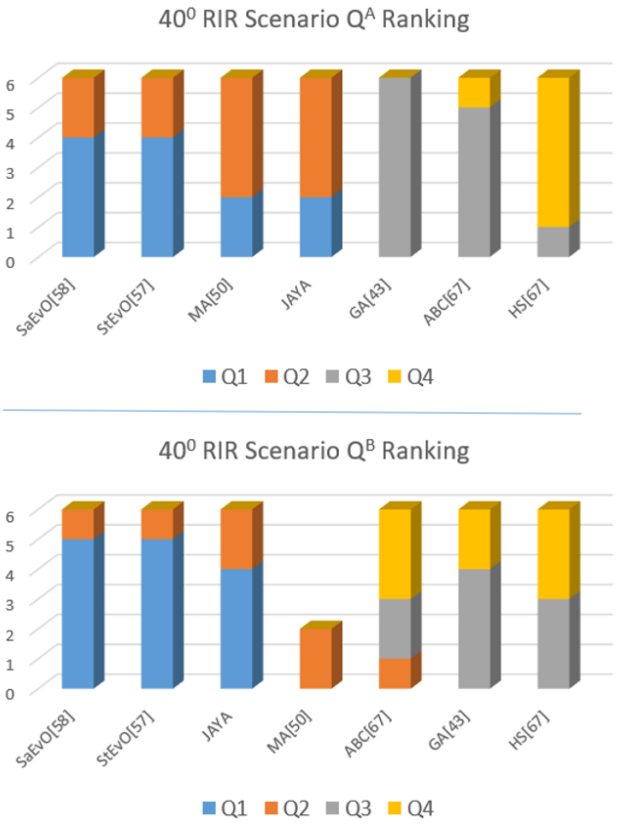

**FIGURE 7. Statistical results of the best seven tested RIR methods when facing the 40 degree overlap RIR scenario. RIR methods are sorted from left (best) to right (worst) according to their best performance considering the quartile-based performance scores $Q^A$ and $Q^B$ in Table 2.**

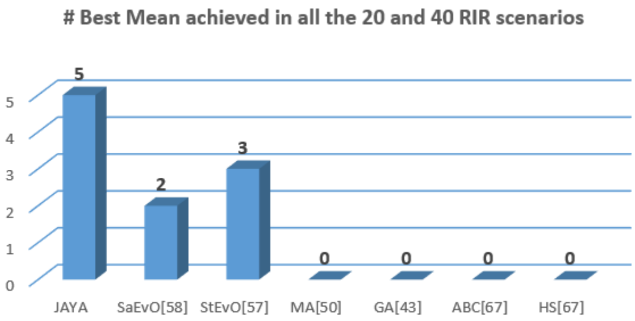

**FIGURE 8. Ranking of the seven best tested RIR methods when facing all the RIR scenarios, i.e. 20 and 40 degree overlap. From left to right are shown the best and worst methods, respectively, according to the number of times each method achieved the best mean value in Tables 1 and 2.**

repositories in the literature have been used to address this type of problems, i.e. the SAMPL repository. From the results obtained, it can be concluded that the methods that perform an adaptive adjustment of the control parameters are the ones that offer the best performance. In particular, the proposal based on the JAYA algorithm is the method with the greatest robustness within this last category, which is one of the main objectives to be analyzed in this work.

## REFERENCES

[1] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D points sets," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 698–700, 1987.

[2] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image Vision Comput.*, vol. 21, pp. 977–1000, 2003.

[3] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vision Comput.*, vol. 25, no. 5, pp. 578–596, 2007.

[4] J. Santamaría, O. Cordón, and S. Damas, "A comparative study of state-of-the-art evolutionary image registration methods for 3D modeling," *Comput. Vis. Image Underst.*, vol. 115, pp. 1340–1354, 2011.

[5] J. Santamaría, M. Rivero-Cejudo, M. Martos-Fernández, and F. Roca, "An overview on the latest nature-inspired and metaheuristics-based image registration algorithms," *Appl. Sci.*, vol. 10, p. 1920, 2020.

[6] C. Cocianu, C. Uscatu, and A. Stan, "Evolutionary image registration: A review," *Sensors*, vol. 23, no. 2, p. 967, 2023.

[7] Y. Liu, "Improving ICP with easy implementation for free form surface matching," *Pattern Recogn.*, vol. 37, no. 2, pp. 211–226, 2004.

[8] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Third International Conference on 3D Digital Imaging and Modeling (3DIM'01)*, Quebec, Canada, 2001, pp. 145–152.

[9] L. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, vol. 11, no. 6, pp. 48–56, 1994.

[10] F. Glover and G. A. Kochenberger, Eds., *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.

[11] M. Gendreau and J. Potvin, Eds., *Handbook of Metaheuristics*, ser. International Series in Operations Research and Management Science. Springer, March 2019, no. 978-3-319-91086-4.

[12] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. IOP Publishing Ltd and Oxford University Press, 1997.

[13] W. Banzhaf, P. Machado, and M. Zhang, *Handbook of Evolutionary Machine Learning*, ser. Genetic and Evolutionary Computation. Springer Nature Singapore, 2023.

[14] Y. Bi, X. Ying, B. Xue, P. Mesejo, S. Cagnoni, and M. Zhang, "A survey on evolutionary computation for computer vision and image analysis: Past, present, and future trends," *IEEE T. Evolut. Comput.*, vol. 27, no. 1, pp. 5–25, 2023.

[15] S. Damas, O. Cordón, and J. Santamaría, "Medical Image Registration Using Evolutionary Computation: A Survey," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 26–42, 2011.

[16] A. E. Eiben and S. K. Smith, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm Evolut. Comput.*, vol. 1, pp. 19–31, 2011.

[17] V. Balas, S. Roy, D. Sharma, and P. Samui, Eds., *Handbook of Deep Learning Applications*. Springer, 2019.

[18] R. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, pp. 19–34, 2016.

[19] L. Silva, Y. S. Lúcio, L. Coelho, V. Mariani, and R. Rao, "A comprehensive review on jaya optimization algorithm," *Artif. Intell. Rev.*, vol. 56, no. 5, pp. 4329–4361, 2023.

[20] R. J. Campbell and P. J. Flynn, "A survey of free-form object representation and recognition techniques," *Comput. Vis. Image Underst.*, vol. 81, no. 2, pp. 166–210, 2001.

[21] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239–256, 1992.

[22] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vision Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[23] K. De Jong, *Evolutionary Computation*. The MIT Press, 2002.

[24] F. Glover, M. Laguna, and R. Martí, "Scatter search," in *Advances in Evolutionary Computation: Theory and Applications*, A. Ghosh and S. Tsutsui, Eds. New York: Springer-Verlag, 2003, pp. 519–537.

[25] H. Schwefel, *Evolution and Optimum Seeking: The Sixth Generation*. New York, NY, USA: John Wiley & Sons, Inc., 1993.

[26] R. Marti, "Multi-start methods," in *Handbook of Metaheuristics*, F. Glover and G. A. Kochenberger, Eds. Kluwer Academic Publishers, 2003, pp. 355–368.

[27] K. Price, "An introduction to differential evolution." in *New ideas in optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. Cambridge, UK: McGraw-Hill, 1999, pp. 79–108.

[28] R. Storn, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces." *J. Global Optim.*, pp. 341–359, 1997.

[29] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.

[30] M. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: A review," *Evolut. Comput.*, vol. 25, no. 1, pp. 1–54, 2017.

[31] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengotxea, Eds., *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Springer Verlag, 2006.

[32] D. Diaz-Pernil, M. Gutierrez-Naranjo, and H. Peng, "Membrane computing and image processing: a short survey," *J. Membr. Comput.*, vol. 1, pp. 58–73, 2019.

[33] K. Bhattacharjee, N. Naskar, S. Roy, and S. Das, "A survey of cellular automata: Types, dynamics, non-uniformity and applications," *Nat. Comput.*, vol. 19, pp. 433–461, 06 2020.

[34] G. Chen, H. Tang, B. Niu, and C. Lee, "A survey of bacterial foraging optimization," *Neurocomputing*, vol. 452, pp. 728–746, 01 2021.

[35] E. Kaya, B. Gorkemli, B. Akay, and D. Karaboga, "A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems," *Eng. Appl. Artif. Intell.*, vol. 115, p. 105311, 10 2022.

[36] X. Gao, G. Vaiyapuri, H. Xu, X. Wang, and K. Zenger, "Harmony search method: Theory and applications," *Comput. Intell. Neurosci.*, vol. 2015, pp. 258 491:1–258 491:10, 2015.

[37] M. Shehab, M. Abu-Hashem, M. Shambour, A. Alsalibi, O. Alomari, J. Gupta, A. Alsoud, B. Abuhaija, and L. Abualigah, "A comprehensive review of bat inspired algorithm: Variants, applications, and hybridization," *Arch. Comput. Methods Eng.*, vol. 30, pp. 765 – 797, 2022.

[38] X. Yang, *Nature-Inspired Metaheuristic Algorithms*, 07 2010.

[39] Y. Meraihi, A. Gabis, S. Mirjalili, and A. Cherif, "Grasshopper optimization algorithm: Theory, variants, and applications," *IEEE Access*, vol. 9, pp. 50 001–50 024, 2021.

[40] J. Fitzpatrick, J. Grefenstette, and D. Gucht, "Image registration by genetic search," in *IEEE Southeast Conference*, Louisville, EEUU, 1984, pp. 460–464.

[41] F. Bernardini and H. Rushmeier, "The 3D model acquisition pipeline," *Computer Graphics Forum*, vol. 21, no. 2, pp. 149–172, 2002.

[42] M. Rodrigues, R. Fisher, and Y. Liu, "Special issue on registration and fusion of range images," *Comput. Vis. Image Underst.*, vol. 87, no. 1-3, pp. 1–7, 2002.

[43] L. Silva, O. R. P. Bellon, and K. L. Boyer, "Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 762–776, 2005.

[44] S. M. Yamany, M. N. Ahmed, and A. A. Farag, "A new genetic-based technique for matching 3D curves and surfaces," *Pattern Recogn.*, vol. 32, pp. 1817–1820, 1999.

[45] E. Lomonosov, D. Chetverikov, and A. Ekart, "Pre-registration of arbitrarily oriented 3D surfaces using a genetic algorithm," *Pattern Recogn. Lett.*, vol. 27, no. 11, pp. 1201–1208, 2006.

[46] C. Cocianu and A. Stan, "New evolutionary-based techniques for image registration," vol. 9, no. 1, p. 176, 2019.

[47] X. Yang, *Nature-Inspired Optimization Algorithms*, 2014.

[48] M. Roslan, N. Ali, N. Radzi, and M. Amin, "Enhanced monomodal image registration process with cuckoo search algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 864, no. 1, p. 012051, 2020.

[49] C. Cocianu and C. Uscatu, "Multi-scale memetic image registration," *Electronics*, 2022.

[50] J. Santamaría, O. Cordón, S. Damas, J. García-Torres, and A. Quirin, "Performance evaluation of memetic approaches in 3D reconstruction of forensic objects," *Soft Comput.*, vol. 13, no. 8-9, pp. 883–904, 2009.

[51] Z. Zhang, Y. Dai, and J. Sun, "Deep learning based point cloud registration: an overview," *Virtual Real. Intell. Hardw.*, vol. 2, no. 3, pp. 222–246, 2020.

[52] Y. Wang and J. Solomon, "Deep closest point: Learning representations for point cloud registration," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3522–3531.

[53] Y. Wang, Y. Sun, Z. Liu, S. Sarma, M. M. Bronstein, and J. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 146:1–146:12, 2019.

[54] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 2511–2520.

[55] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "Deepgmr: Learning latent gaussian mixture models for registration," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12350. Springer, 2020, pp. 733–750.

[56] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search," in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds. Kluwer Academic Publishers, 2003, pp. 321–353.

[57] J. Santamaría, S. Damas, J. M. Garcia-Torres, and O. Cordón, "Self-adaptive evolutionary image registration using differential evolution and artificial immune systems," *Pattern Recognition Letters*, Dec 1 2012 2012.

[58] J. Santamaría, S. Damas, O. Cordón, and A. Escamez, "Self-adaptive evolution: Toward new parameter free image registration methods," *IEEE T. Evolut. Comput.*, vol. 17, no. 4, pp. 545–557, Aug 2013 2013.

[59] I. D. Falco, U. Scafuri, E. Tarantino, A. D. Cioppa, K. Yetongnon, A. Dipanda, R. DePietro, and L. Gallo, "Fast range image registration by an asynchronous adaptive distributed differential evolution," in *2016 12th International Conference on Signal-Image Technology and Internet-based Systems (SITIS)*, Naples, Italy, 2016, pp. 643–651.

[60] R. Rao, "Review of applications of tlbo algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems," *Decision Science Letters*, vol. 5, pp. 1–30, 01 2016.

[61] R. Rao and K. More, "Design optimization and analysis of selected thermal devices using self-adaptive jaya algorithm," *Energy Conversion and Management*, vol. 140, pp. 24–35, 05 2017.

[62] H. G. Beyer and K. Deb, "On self-adaptive features in real-parameter evolutionary algorithms," *IEEE T. Evolut. Comput.*, vol. 5, no. 3, pp. 250–270, June 2001.

[63] H. Kita, "A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms," *Evolut. Comput.*, vol. 9, pp. 223–41, 02 2001.

[64] N. Noman and H. Iba, "Enhancing Differential Evolution Performance with Local Search for High Dimensional Function Optimization," in *Genetic and Evolutionary Computation Conference (GECCO'05), ACM*, 2005, pp. 967–974.

[65] I. de Falco, A. Della Cioppa, D. Maisto, and E. Tarantino, "Differential Evolution as a viable tool for satellite image registration," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1453–1462, 2008.

[66] M. P. Wachowiak, R. Smolikova, Y. Zheng, J. M. Zurada, and A. S. El-Maghraby, "An approach to multimodal biomedical image registration utilizing particle swarm optimization," *IEEE T. Evolut. Comput.*, vol. 8, no. 3, pp. 289–301, 2004.

[67] J. Garcia-Torres, S. Damas, O. Cordon, and J. Santamaria, "A case study of innovative population-based algorithms in 3d modeling: Artificial bee colony, biogeography-based optimization, harmony search," *Expert Systems with Applications*, vol. 41, p. 1750–1762, 03 2014.

[68] C. K. Chow, H. T. Tsui, and T. Lee, "Surface registration using a dynamic genetic algorithm," *Pattern Recogn.*, vol. 37, pp. 105–117, 2004.

**DR. JOSÉ SANTAMARÍA** was born in Barcelona and is residing in the province of Jaén, Spain. Since 2006, he holds a PhD in Computer Science. His research interest is focused on Computer Vision and Deep Learning. His scientific production and lecturer activities began in 2003 and 2005, respectively. He has held different positions as researcher and teaching staff at the universities of Cádiz, Granada, and Jaén. He is Associate Professor at the department of computer science of the University of Jaén since 2016. He is in charge of the "Applied Computational Engineering" Research Group. During more than 20 years of research experience, he has successfully supervised several master degree and PhD students. Moreover, he has participated as principal researcher and team member in more than 10 research projects funded at regional, national and European level (e.g. FP7 Programme).

Since 2003, Dr. Santamaría has published more than 80 scientific contributions, with more than forty papers being accepted by international journals indexed in the JCR database, most of them ranked first quartile (Q1). According to the Scopus platform, his scientific production has accumulated more than 3000 citations (H-index and i10-index greater than 20). Also, several of his contributions are considered highly cited papers, with more than 1K citations in some cases. Moreover, he is co-inventor of a patent with international coverage (US20120182294A1). Additionally, he has been awarded with the "IFSA Award for Outstanding Applications of Fuzzy Technology". He actively performs professional tasks as reviewer, guest editor, and editorial board member for several renowned international journals, e.g. Computational Intelligence, Computer Methods and Programs in Biomedicine, among others. He is an exper evaluator for the European Commission. The research group he leads has managed to forge several international collaborations with Indian and Australian academic and research institutions: The Pandit Deendayal Energy University and The Queensland University of Technology, respectively.