Expert Systems with Applications 41 (2014) 1750–1762

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A case study of innovative population-based algorithms in 3D modeling: Artificial bee colony, biogeography-based optimization, harmony search [☆]

José M. García-Torres [a,*], Sergio Damas [b], Oscar Cordón [a,b,c], José Santamaría [d]

[a] Dept. of Computer Science and Artificial Intelligence, University of Granada, Spain
[b] European Centre for Soft Computing, Mieres, Spain
[c] Centro de Investigación en Tecnologías de La Información y de las Comunicaciones (CITIC), University of Granada, Spain
[d] Dept. of Computer Science, University of Jaén, Spain

ARTICLE INFO

Keywords:
Population-based metaheuristics
ABC
BBO
HS
Image registration
3D modeling

ABSTRACT

Deterministic or analytical methods for computing the global optima of a functional have been extensively applied in a wide range of engineering applications. Nevertheless, it is wellknown they usually lack of effectiveness when dealing with complex nonlinear optimization problems. In particular, such a shortcomings have been addressed by using approximate approaches, named metaheuristics. Among them all, those methods using a population-based scheme, e.g. the evolutionary algorithms, have been the most successful optimization strategies. Recently, innovative population-based algorithms such as ABC, BBO, and HS have arisen as promising optimization methods due to they provide a good tradeoff between design and performance when compared to other more elaborated methods. In this work, we aim to first introduce the particular design of these three cutting edge algorithms, and additionally analyse their performance when tackling a challenging real-world optimization problem. In particular, our case study of numerical optimization tackles a computer vision problem named 3D range image registration for 3D modeling tasks. Computational experiments have been conducted comparing the performance of ABC, HS, and BBO against other contributions in the state-of-the-art of 3D image registration.

1. Introduction

Optimization problems are often complex situations to cope with in several areas of knowledge such as engineering. The objective function may have many local optima and in many cases finding the best solution (named global optimum) is so time-consuming that goes beyond the admissible in practical applications. Those problems cannot be handled by classical methods (e.g. gradient-based algorithms) which are likely to compute local optima. Thus, there remains a need for efficient and effective numerical optimization methods for tackling challenging real-world engineering problems. In the last few decades, approximate algorithms, named metaheuristics (MHs) (Glover & Kochenberger, 2003; Luke, 2009), have demonstrated their good performance in these kinds of problems, where the guarantee of finding the optimal solution is relaxed in order to obtain high quality solutions in a much more reduced time interval.

There are different kinds of MHs. Among them all, population-based techniques work on a population of solutions based on analogies with natural phenomena. This approach has been applied to a large amount of engineering optimization problems and it has being proved to be effective in solving well-known challenging problems. Within population-based techniques, we can find classical techniques such as genetic algorithms (GAs) (Goldberg, 1989; Michalewicz, 1996), particle swarm optimization (PSO) (Clerc, 2006; Kennedy & Eberhart, 2001) and ant colony optimization (ACO) (Dorigo & Di Caro, 1999; Dorigo & Stützle, 2004; Farhaana et al., 2012; Zhou & Wang, 2012). Recently, many new population-based approaches have been arised: artificial bee colony (ABC) (Karaboga & Basturk, 2007a, 2007b), differential evolution (DE) (Price, 1999; Storn, 1997), harmony search (HS) (Geem, Kim, & Loganathan, 2001), cats swarming (CS) (Chu & Tsai, 2007), and biography-based optimization (BBO) (Simon, 2008), among others.

Testing these new population-based approaches and carrying out a comparison with other state of the art methods may serve the field to analyse both their shortcomings and goggness in

[☆] This work is partially supported by both the Spanish Ministerio de Educación y Ciencia (Ref. TIN2009-07727) including EDRF fundings and the University of Jaén (Ref. R1/12/2010/61) including fundings from *Caja Rural de Jaén*.

[*] Corresponding author. Tel.: +34 660057583.

E-mail addresses: jmgt@correo.ugr.es (J.M. García-Torres), sergio.damas@softcomputing.es (S. Damas), oscar.cordon@softcomputing.es (O. Cordón), jslopez@ujaen.es (J. Santamaría).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.08.074

performance, thus allowing the field to propose more advanced variants, e.g. addressing hybrid strategies. In this work, we aim at analysing the performance of three cutting edge algorithms: ABC, BBO, and HS. To do so, we have considered a case study of numerical optimization within the field of computer vision, known as the image registration (IR) problem (Zitová & Flusser, 2003). In particular, we addressed several IR problem instances for 3D modeling by using laser range scanners (Santamaría, Cordón, & Damas, 2011). Moreover, these results were compared to those provided by other state-of-the-art population-based algorithms in the field of IR.

The structure of this contribution is as follows. In Section II, a description of the optimization design of the population-based algorithms ABC, BBO, and HS. Then, in Section III, we are introducing the basics of the real-world problem tackled which is known as range image registration (RIR) (Bernardini & Rushmeier, 2002). A revision of IR methods using population-based approaches will also be followed in this section. the mos relevant conclusions will be shown in section IV. Finally, Section V collects some concluding remarks as well as possible works for the near future.

2. Recent advancements on population-based algorithms

2.1. Background

Numerical optimization problems are encountered in many domains, e.g. science, engineering, management, and business. Formally, they may be defined as a couple $P = (X, F)$ where X is named as search space and it represents the set of feasible problem solutions $X = \{x_1, \ldots, x_{|X|}\}$. Each of the latter d-dimensional vectors, $x \in X$, consists of a set of design variables, $x = (x^1, \ldots, x^d)$, each one ranging to a particular continuous domain $x^j \in [LB^j, UB^j]$ $(1 < j < d)$. A function $F : X \rightarrow \mathrm{IR}^+$, so-called the objective function which assigns a real value to every solution $x \in X$ indicating its quality.

The main challenge in solving an optimization problem is to find the global optimal solution. However, computing optimal solutions in many real-world applications would be so time-consuming that go beyond the admissible, basically due to the high dimension of the factible solutions space. In the last decades, MHs have emerged as a new kind of approximate search and optimization algorithms (Glover & Kochenberger, 2003; Luke, 2009). They combine basic heuristic methods in order to explore efficient and effectively the search space which will provide acceptable solutions in a reasonable time.

One of the main advantages of MHs is that they make use of a general purpose optimization framework requiring relatively few modifications to be applied to a specific problem. The MHs family include methods as Simulated Annealing (SA), tabu search (TS), multi-start local search (MS), iterated local search (ILS), variable neighborhood search (VNS), and greedy randomized adaptative search procedures (GRASP). These are usually termed as trajectory-based MHs. On the other hand, population-based MHs, e.g. evolutionary algorithms (EAs) (Bäck, Fogel, & Michalewicz, 1997; Fogel, 2005), consider populations of candidate problem solutions instead.

In particular, GAs are probably the most extended population-based algorithm in the literature to face real-world optimization problems. GAs are theoretical and empirically found to provide global near-optimal solutions for several problems of complex optimization. The search space represented in GAs is a collection of individuals (problem solutions) or chromosomes conforming a population, each of them operating simultaneously on several points of the search space. An initial set/population of solutions is randomly generated. Then, a pool of parents is randomly selected for reproduction on the basis of the *fitness function*,[1] which measures how good is each candidate solution and guides the search space exploration strategy.

The reproduction procedure, which is based on crossover and mutation operators is iteratively performed at every generation (iteration) in order to generate the offspring population. Crossover operators systematically/randomly mix parts (block of genes) of two individuals of the previous population, and additionally every new combined individual is subjected to random changes by using mutation operators. The next generation is produced using a replacement mechanism which selects individuals from the pool composed of the parents and the new offspring generated.

Regardless the approach (trajectory-based vs. population-based), MHs constitute a very interesting choice to achieve a good quality solution in a reasonable time. Specifically, optimization algorithms, already based on the evolution of populations of solutions, have obtained a remarkable success. The next section will be devoted to introduce the description of three innovative population-based MHs recently proposed in the literature, namely ABC, BBO, and HS.

2.2. Artificial bee colony

Swarm intelligence has become an interesting research nowadays (Bonabeau, Dorigo, & Theraulaz, 1999; Cui, Zeng, & Sun, 2006) been applied to solve optimization problems (Zhou & Wang, 2012). The ABC algorithm is a new swarming variant inspired in the population-based approach proposed by Karaboga and Basturk (2007a, 2007b) which is based on the intelligent behavior of honeybee swarms.

In the ABC algorithm, the colony of artificial bees is divided into the three following categories (Karaboga & Basturk, 2008):

- **Employed:** They take nectar from the food source to the hive and share information with onlookers about their location.
- **Onlookers:** Those specialized bees tend to select a food source (the most profitable one) according to their quality, which is given by shared information provided by employed bees in the hive.
- **Scouts:** They are employed bees whose food source has been abandoned. They start to search a new food source randomly.

The algorithm works just by including a common area in the hive so-called the *dancing area*, where bees share and exchange information about food sources. Bees identify the quality of food source by means of the duration of dancing which is determined by the nectar contained in the food source being exploited and its distance to the hive. From the optimization viewpoint, each food source represents a possible solution to the problem, where scouts perform exploration and employed and onlooker bees are focused on the exploitation of search space.

Specifically, the ABC algorithm is sketched in Fig. 1. It starts by associating all the employed bees to randomly generated food sources. Each food source x_i $(i = 1, 2, \ldots, N)$ is a *d-dimensional* vector where d is the number of optimization parameters. Next, each iteration is done as follows. Every employed bee determines a food source within the neighborhood of its current source when using the expression

$$v_i^j = x_i^j + \Phi_i^j (x_i^j - x_k^j) \tag{1}$$

where k different to $i \in 1, 2, \ldots N$ and $j \in 1, 2, \ldots, d$ are randomly chosen indexes, and Φ_i^j is a random number between $[-1, 1]$ which

[1] The fitness or objective function is one of the most important components of heuristic methods whose design affects dramatically to the performance of the implemented method.

```
Begin ABC

Initialize population /*food source positions*/
Evaluate population /*calculate the nectar amounts*/
best food ← Best(population)
While (Not reached stop criterion) Do
    /*Produce new solutions $v_i$ for employed bees. Greedy-selection($x_i$,$v_i$)*/
    For $i \leftarrow 1$ to $N$ Do
        $k \leftarrow (Rand(1, N) \neq i)$
        $j \leftarrow Rand(1, d)$
        $v_{ij} \leftarrow x_{ij} + \Phi_{ij}(x_{ij} - x_{kj})$
        If ($F(v_i)$ is better than $F(x_i)$) Then
            Move bee to $v_i$
            $trials(i) \leftarrow 0$
        Else
            $trials(i) \leftarrow trials(i) + 1$
        End-If
    End-For
    /*Calculate $p_i$ values for the solutions $x_i$*/
    For $i \leftarrow 1$ to $N$ Do
        $p_i \leftarrow \frac{F(x_i)}{\sum_{j=1}^{N} F(x_j)}$
    End-For
    /*Produce new solutions $v_i$ for onlookers depending on $p_i$. Greedy-selection($x_i$,$v_i$)*/
    For $i \leftarrow 1$ to $N$ Do
        $p \leftarrow U[0, 1]$
        If ( $p < p_i$ ) Then
            $k \leftarrow (Rand(1, N) \neq i)$
            $j \leftarrow Rand(1, d)$
            $v_i^j \leftarrow x_i^j + \Phi_i^j(x_i^j - x_k^j)$
            If ($F(v_i)$ is better than $F(x_i)$) Then
                Move Bee to $v_i$
                $trials(i) \leftarrow 0$
            Else
                $trials(i) \leftarrow trials(i) + 1$
            End-If
        End-If
    End-For
    /* Scouts moves for searching new food sources */
    For $i \leftarrow 1$ to $N$ Do
        If ( $trials(i) > \#maxtrials$ ) Then
            For $j \leftarrow 1$ to $d$ Do
                $x_i^d \leftarrow LB^j + U[0, 1] * (UB^j - LB^j)$
            End-For
        End-If
    End-For
    best food ← Best(population)
End-While
Return best food source;

End ABC
```

Fig. 1. Pseudo-code of the ABC algorithm.

controls the production of a neighbor food source around x_i^j. When applying a greedy selection process, if the nectar amount of new food (objective function value) is better than the current one for the employed bee, the bee is moving to this new food, leaving its current one. Then, nectar information is shared with onlookers and the onlooker bees will select a food source according to a probability p_i determined using the expression:

$$p_i = \frac{F(x_i)}{\sum_{j=1}^{N} F(x_j)} \tag{2}$$

where $F(x_i)$ is the objective function value of the solution represented by the food source i and N is the total number of food sources. Next, a food source which is not improved in a predetermined number of trials is abandoned and replaced by scouts with a new food source using:

$$x_i^j = LB^j + U[0, 1](UB^j - LB^j) \tag{3}$$

where LB^j and UB^j are the lower and upper bounds for the variable j. Finally, the best solution found so far is stored. ABC assumes that only one employed bee is needed for every food source making the number of food sources is equal to the number of employed bees.

In the last few years, ABC has been applied to solve several optimization problems such as multi-dimensional numeric optimization functions (Karaboga, 2009; Karaboga & Akay, 2009; Karaboga & Basturk, 2007b), numerical benchmark functions which are used in engineering problems with high dimensionality (Karaboga & Basturk, 2008; Alvarado-Iniesta, García-Alcaraz, Rodríguez-Borbón, & Maldonado, 2013), digital filters (Karaboga, 2009) and to the leaf-constrained minimum spanning tree (LCMST) problem (Singh, 2009), among others. Further information can be found in the ABC's homepage (Karaboga, 2013).

2.3. Biography based optimization

Biography based optimization (BBO) (Simon, 2008) is a new population-based algorithm aimed to tackle optimization problems. This innovative approach is based on mathematical models of biography describing natural ways of distributing species, i.e., how species migrate, how they arise and become extinct. Next, we review the biography model components and their correspondence with the BBO algorithm from an optimization viewpoint (Simon, 2008):

- **Habitat (H):** It represents a solution inside the search space of a d-dimensional numerical optimization problem.
- **Habitat Suitability Index (HSI):** In a biography model, geographical areas, that is well suited as residences for biological species are said to have a high HSI. Thus, HSI corresponds to the objective function from an optimization viewpoint.
- **Suitability Index Variables (SIV):** The computation of the HSI value is influenced by additional factors such as rainfall, diversity of vegetation, land area, and temperature. All of them are called the SIVs.
- **Ecosystem:** It refers to a group of N habitats. From a population-based optimization viewpoint, it corresponds to the population of solutions. The size N is usually constant but the use of variable-sized ecosystems is being studied (Simon, 2008).
- **Immigration Rate:** The control parameter λ is devoted to control habitat immigration. The maximum immigration rate in a single habitat (I) is reached when species are not in the habitat. As the number of species increases, it will become crowded and immigration will decrease due to fewer species would be able to survive in that crowd.
- **Emigration Rate:** In this case, the μ parameter controls habitat emigration. If there are no species, emigration is null. As the number of species increases, species are capable to leave their habitat in order to explore other residences. Maximum emigration rate (E) is reached in a single habitat when containing the maximum number of species it can support.
- **Migration model:** According to different mathematical models of biogeography, various migration curves can be applied (Ma, 2010; MacArthur & Wilson, 1967). In Fig. 2, a linear model is illustrated where $S0$ is the equilibrium number of species (Simon, 2008), which denotes that the immigration and emigration rates are equal. I is the maximum possible immigration rate, E is the maximum possible emigration rate, and S_{max} is the largest possible number of species that the habitat can support.

The BBO algorithm has two main operators: *habitat modification* (Ω) and *mutation* (M). The former is an $H^n \longrightarrow H$ probabilistic operator that adjusts an habitat H_i based on the ecosystem H^n according to its immigration rate λ_i and to the emigration rate μ_j taking H_j as the source of the modification. The second one is an $H \longrightarrow H$, a probabilistic operator that randomly modifies the SIV factors of an habitat H_i according to a probability p_i based on both the λ_i and μ_i parameters (Simon, 2008). Both operators let BBO implement elitism for the best b habitats (b is a user-selected elitism parameter) setting $\lambda_i = 0$ in the habitat modification operator and $p_i = 0$ in the mutation operator.

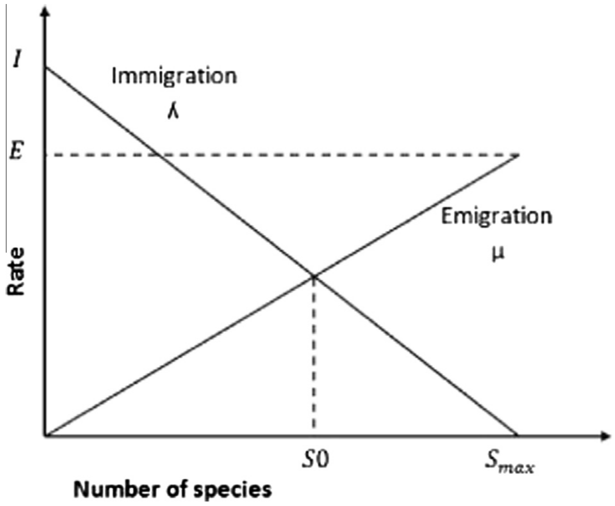

Fig. 2. Linear model of migration (Simon, 2008; MacArthur & Wilson, 1967).

As it is shown in Fig. 3, BBO starts initializing the search with a set of random habitats and the HSI is computed from each of them. Next, the habitat modification (Ω) and mutation (M) operators are subsequently applied on each non-elite habitat ($N - b$ worst habitats) while the stopping criterion is not satisfied.

Recently, BBO has been applied in several optimization problems, for instance: the Traveling Salesman Problem (TSP) (Song, Liu, & Wang, 2010), the optimization of a set of standard multi-variable benchmark functions (Simon, 2008), the tuning of neuro-fuzzy system parameters for diagnosis of cardiac diseases (Ovreiu & Simon, 2010), a sensor selection problem for aircraft engine health estimation (Simon, 2008), economic emission load dispatch problems (Bhattacharya & Chattopadhyay, 2011) and the power flow problem in analysis of circuits (Rarick, Simon, Villaseca, & Vyakaranam, 2009). Further information can be found in the BBO's homepage (Simon, 2013).

2.4. Harmony search

The last population-based algorithm, which was analyzed in this study, is based on analogies with the natural phenomena of the musical process. It is focused on searching the perfect state of harmony, an analogy to the optimization process, and it is called the harmony search (HS) algorithm (Geem et al., 2001). As the optimization procedure tries to find a global solution which is determined by an objective function, a musical performance (e.g., rhythms played on a set of instruments in jazz) seeks to find pleasing harmony determined by an aesthetic standard. The possible values of each variable may be identified in the pitches of the different instruments in HS, where each iteration would be a practice and the global optimum can be considered as finding a Fantastic Harmony.

Furthermore, while each variable of the solution vector is taking value within a range of possible values, the same occurs to HS when each player sounds with a musical note within a candidate range, and all together build an harmony. If the harmony is good, the experience is stored in each player's memory and, next time, te occurrence to achive a more harmonius chord is increased. Basically, HS consists of the following components (Lee & Geem, 2005):

- **Initialization:** The harmony memory (HM) matrix is filled with *HMS* random vectors which are sorted according to the values of the objective function $F(x)$:

```
Begin BBO

Ecosystem initialization;
best habitat ← Best(Ecosystem);
While (Not reached stop criterion) Do
    For $i \leftarrow 1$ to $N$ Do
        Compute HSI for habitat $H_i$
    End-For
    /*Apply Habitat Modification $\Omega(\lambda, \mu)$ over non-elite habitats inside ecosystem*/
    For $i \leftarrow 1$ to $N$ Do
        Select $H_i$ with probability based on the immigration rate $\lambda_i$
        If $H_i$ is selected Then
            For $j \leftarrow 1$ to $N$ Do
                Select $H_j$ with probability based on $\mu_j$
                If $H_j$ is selected Then
                    Randomly select a SIV factor from $H_j$
                    Replace a random SIV factor in $H_i$ with the one selected from $H_j$
                End-If
            End-For
        End-If
    End-For
    /*Apply Mutation $M(\lambda, \mu)$ over non-elite habitats inside ecosystem*/
    For $i \leftarrow 1$ to $N$ Do
        Select a SIV factor in $H_i$ with probability based on the emigration rate;
        If $H_i(SIV)$ is selected Then
            Replace $H_i(SIV)$ with a randomly generated SIV
        End-If
    End-For
    For $i \leftarrow 1$ to $N$ Do
        Recompute HSI for habitat $H_i$
        If ($HSI(H_i)$ is better than $HSI(besthabitat)$) Then
            best habitat ← $H_i$;
        End-If
    End-For
End-While
Return best habitat

End BBO
```

Fig. 3. Pseudo-code of the BBO algorithm.

$$HM = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & \ldots & x_1^d \\ x_2^1 & x_2^2 & x_2^3 & \ldots & x_2^d \\ x_3^1 & x_3^2 & x_3^3 & \ldots & x_3^d \\ \vdots & \ldots & & & \\ x_{HMS}^1 & x_{HMS}^2 & x_{HMS}^3 & \ldots & x_{HMS}^d \end{bmatrix} \tag{4}$$

where HMS is the harmony memory size. The parameters *harmony memory considering rate* (HMCR) and the *pitch adjusting rate* (PAR) are also initialized. HMCR refers to the probability of choosing random notes. On the other hand, PAR mimics the adjustment of each instrument by shifting to neighboring values (Lee & Geem, 2005) according to:

$$x^{j'} = x^j \pm r \times bw, \quad 1 \leqslant j \leqslant d \tag{5}$$

where $r \sim U[0,1]$ and bw is an arbitrary distance bandwidth for the continuous design variable.

- **Harmony Improvisation:** A new harmony vector, $x_i = < x_i^1, x_i^2, \ldots, x_i^d >$ is generated considering HCMR and PAR. For each instrument $x_i^j (1 \leqslant j \leqslant d)$, the value is selected as follows:

$$x_i^j \leftarrow \left\{ \begin{array}{ll} x_i^j \in HM, & \text{if } U[0,1] <= HMCR \\ x_i^j \in [LB^j, UB^j], & \text{if } U[0,1] > HMCR \end{array} \right\} \tag{6}$$

where LB^j and UB^j are the lower and upper bounds of each variable.

- **Update of HM:** If the quality of the new harmony x_i' is better than the current best in HM, the new harmony is included in HM by replacing the worst one in HM.

As the previous methods, HS has been recently applied in optimization problems as the TSP (Geem et al., 2001), the minimization of multi-variable functions (Geem et al., 2001; Omran & Mahdavi, 2008), the design of a pipeline network for water supply (Geem et al., 2001), various engineering optimization problems (Lee & Geem, 2005), the addressing of sudoku puzzles (Geem, 2007), the spread spectrum radar polyphase codes design problem (Gil-López et al., 2012) and the sum-of-ratios optimization problem applied in economy and engineering (Jaberipour & Khorram, 2010). Further information can be found in the HS's homepage (Geem, 2013).

3. Range image registration in 3D modeling: a case study

IR (Zitová & Flusser, 2003) is a fundamental task in computer vision and computer graphics fields used to find either a spatial *transformation* (e.g., rotation, translation, etc.) or a correspondence (matching of similar image features) among two or more images acquired under different conditions: with different times, by using different sensors, by taking different viewpoints, or with a combination of them. IR aims to achieve the best possible overlapping, by transforming those independent images into a common one. Over the years, IR has been applied to tackle with many real-world problems ranging from remote sensing to medical imaging, artificial vision, and computer-aided design (CAD).

The optimization process applied by traditional IR methods like the classical iterative closest point (ICP) algorithm (Besl & McKay, 1992; Chen & Medioni, 1992) is likely to provide incorrect registration transformation estimations due to the fact that these methods are usually prone to be trapped in local minima (Liu, 2004; Masuda, 2002; Rusinkiewicz & Levoy, 2001; Sharp, Lee, & Wehe, 2002) since they assume a rough prealignment of the images. (See Fig. 4).

In the last few years, there has been an increasing interest in adopting MHs as the optimization technique for IR methods (Santamaría et al., 2011). In particular, evolutionary algorithms (EAs) (Bäck, 1996; Bäck et al., 1997) have been successfully applied to tackle with IR problems without requiring a good initial estimation of the image alignment.

The first attempts to face the IR problem using EC can be found in the eighties (Fitzpatrick, Grefenstette, & Gucht, 1984), where a GA (Michalewicz, 1996) was developed for tackling rigid IR of 2D angiographic images. Since then, evolutionary IR has become a very active area due to the successful results and several well-known EAs which have been considered to tackle with the IR optimization process (Damas, Cordón, & Santamaría, 2011; Santamaría et al., 2011) where a two-stage IR approach is usually considered: a first coarser stage, named as prealignment, and a refinement step usually applying ICP-based IR algorithms.

3.1. Problem definition

The key idea of the IR process is to achieve the transformation that places different 2D/3D images in a common coordinate system. There is not an universal design for a hypothetical IR method that could be applicable to all registration tasks due to the particular application items that must be taken into account in every real world application (Zitová & Flusser, 2003). However, IR methods usually require the four following components:

- Two input **Images** named 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 $\vec{p}'_j$ being image points.
- A **Registration transformation** f, relating the two images. Typically, it is a parametric function.

  $$f : \Re^2/\Re^3 \longrightarrow \Re^2/\Re^3 \qquad (7)$$

  based on geometric transformations such as translation, rotation, scale, etc.
- A **Similarity metric function** F. It aims to measure a qualitative value of closeness or degree of fitting between the transformed scene image, noted by $f'(I_s)$, and the model image.

  $$F(I_e, I_m; f) = \mid f(I_e) - I_m \mid \qquad (8)$$

- An **Optimizer**. It is a method that seeks the optimal solution $f*$ which optimizes F

  $$f^* = \arg\ \min/\max_f\ F(I_e, I_m; f) \quad s.t.$$

  $$f^*(I_e) = I_m \qquad (9)$$

  An iterative process is often followed (see Fig. 5) which usually finishes when convergence is achieved, i.e., when the similarity metric is bellow a given tolerance threshold.

```
Begin HS

Initialize HM;
best harmony ← Best(HM);
While ($Not\ reached\ stop\ criterion$) Do
    For $i \leftarrow 1$ to $d$ Do
        $j \in Rand(1, HMS)$
        If ($U[0,1] < HMCR$) Then /*memory consideration*/
            $x^{i'} = x^i_j$
            If ($U[0,1] < PAR$) Then /*pitch adjustment*/
                $x^{i'} = x^{i'} \pm r \times bw$
        Else /*random selection*/
                $x^{i'} \leftarrow LB^j + r \times (UB^j - LB^j)$
        End-If
    End-For
    If ($F(x')$ is better than worst in $HM$) Then
        $HM \leftarrow (HM \setminus worst) \cup x'$
    End-If
    If ($F(x')$ is better than best harmony) Then
        best harmony ← $x'$
    End-If
End-While
Return best harmony

End HS
```

Fig. 4. Pseudo-code of the HS algorithm.

3.2. Range image registration

In order to provide a more specific description of the problem, we focused our attention on the particular application we are considering in our experiments: the pair-wise IR of range images (i.e. RIR) for 3D modeling of real objects (Campbell & Flynn, 2001; Godin, Hebert, Masuda, & Taubin, 2009; Rodrigues, Fisher, & Liu, 2002; Salvi, Matabosch, Fofi, & Forest, 2007).

Range scanners are able to capture 3D images, named range images, of the surface of the sensed object. Every range image is acquired from a particular viewpoint and it partially models the geometry of the scanned object. Thus, it is mandatory to consider a reconstruction technique to perform the accurate integration of the images in order to achieve a complete and reliable model of the physical object. This framework is usually called 3D modeling (see Fig. 6) and it is based on applying IR techniques to achieve the integration of the range images (Salvi et al., 2007; Campbell & Flynn, 2001; Rodrigues et al., 2002; Godin et al., 2009).

The 3D model reconstruction procedure involves several pair-wise alignments of two adjacent range images in order to obtain the final 3D model of the physical object. Therefore, every pair-wise IR method aims to find the Euclidean motion that brings the *scene* view (I_s) into the best possible alignment with the *model* view (I_m). An Euclidean motion is usually considered based on a 3D rigid transformation (f) determined by seven real-coded parameters, that is: a rotation $R = (\theta, Axis_x, Axis_y, Axis_z)$, with θ and $\vec{Axis}$ being the angle and axis of rotation, and a translation $\vec{t} = (t_x, t_y, t_z)$, respectively. Then, the transformed points of the *scene* view are denoted by

$$f(\vec{p}_i) = R(\vec{p}_i) + \vec{t}, \quad i = 1 \cdots N_{I_s} \qquad (10)$$

Hence, the pair-wise IR task can be formulated as an optimization problem developed to search for the Euclidean transformation f^* achieving the best alignment of both images according to the considered *Similarity metric F*:

$$f^* = \arg\ \min_f\ F(I_s, I_m; f) \quad s.t. : f^*(I_s) \cong I_m \qquad (11)$$

The median square error (MedSE) is usually considered the Similarity metric in 3D modeling (Rodrigues et al., 2002; Santamaría, Cordón, Damas, García-Torres, & Quirin, 2009):

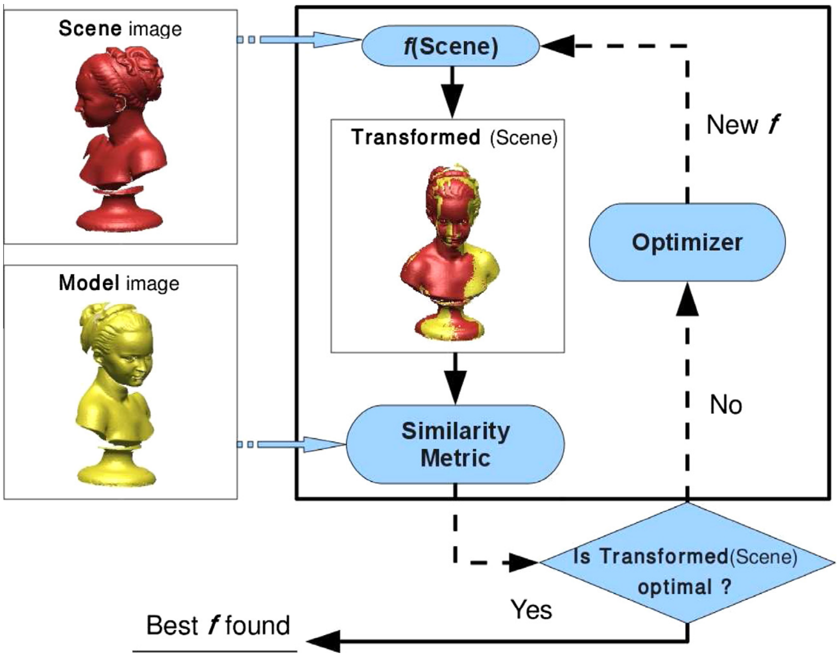

Fig. 5. The IR optimization process

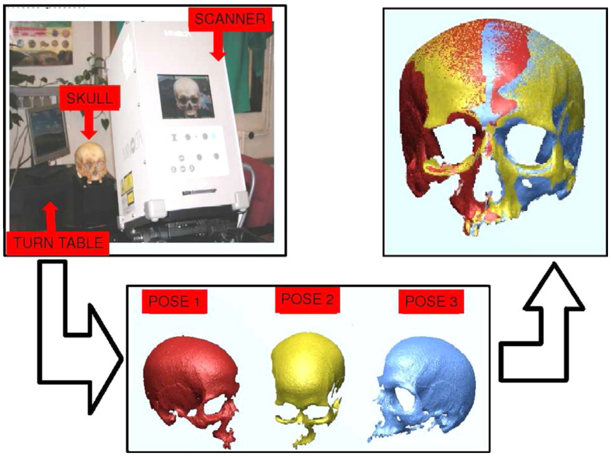

Fig. 6. The 3D modeling procedure of forensic objects

$$F(I_s, I_m; f) = MedSE(d_i), \quad \forall i \in \{1, \ldots, N_{I_s}\} \tag{12}$$

where *MedSE*() corresponds to the median d_i value. We define $d_i = \|f(\vec{p}_i) - \vec{q}_{cl}\|^2$ as the squared Euclidean distance between the transformed scene point, $f(\vec{p}_i)$, and its corresponding closest point, $\vec{q}_{cl}$, in the *model* view I_m. In order to speed up the computation of the closest point q_{cl} of I_m, indexing structures as kd-trees (Zhang, 1994) or the grid closest point (GCP) transform proposed in (Yamany, Ahmed, & Farag, 1999) are often used.

4. Experiments and analysis of results

This section is aimed at presenting a number of experiments to be studied for the performance of the three analyzed population-based algorithms according to accuracy and robustness. Specifically, we considered a benchmark suite of several problem instances of the well-known IR problem by using range images. The obtained results will be benchmarked against those obtained by four state-of-the-art evolutionary IR methods, i.e.:

- Santamaria et al.'s proposal (StEvO) (Santamaría, Damas, García-Torres, & Cordón, 2012), a recent contribution based on a automatic or self-adaptive tuning of the EA control parameters.
- de Falco et al.'s method (deFalco08) (de Falco, Della Cioppa, Maisto, & Tarantino, 2008), which makes use of a basic implementation of the differential evolution (DE) algorithm (Storn, 1997).
- Silva et al.'s contribution (Silva05) (Silva, Bellon, & Boyer, 2005), in which a steady-state variant of GAs (Goldberg, 1989) is developed.
- Yamany et al.'s proposal (Yamany99) (Yamany et al., 1999), where the authors considered a binary representation of the transformation parameters and a canonical implementation of GAs.

Both these three analyzed and the latter four algorithms were implemented in C++ and compiled with the GNU/g++ tool. We adapted all the tested methods by using the same representation of the rigid transformation (*f*) and objective function (see Eq. (12) in Section 3.2) in order to accomplish a fair comparison. The similarity metric considered will depend on the particular real-world application being faced.

4.1. Range image datasets and problem scenarios

In order to ease the comparison with the reported results in other contributions in the field (Salvi et al., 2007; Silva et al., 2005), our results 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, *http://sampl.ece.ohio-state.edu/data/3DDB/RID/index.htm*). Specifically, Fig. 7 shows the six range datasets considered, named as in previous contributions (Silva et al., 2005): "Frog", "Bird", "Tele", "Lobster", "Angel", and "Buddha". Each dataset ranges from 8 K to 15 K of size.

The second group of datasets is the specific ones, acquired using a Konica-Minolta© VI-910 laser scanner, named "Skull" and "Tooth" (see Fig. 8). Each dataset ranges from 30 K to 70 K of size.

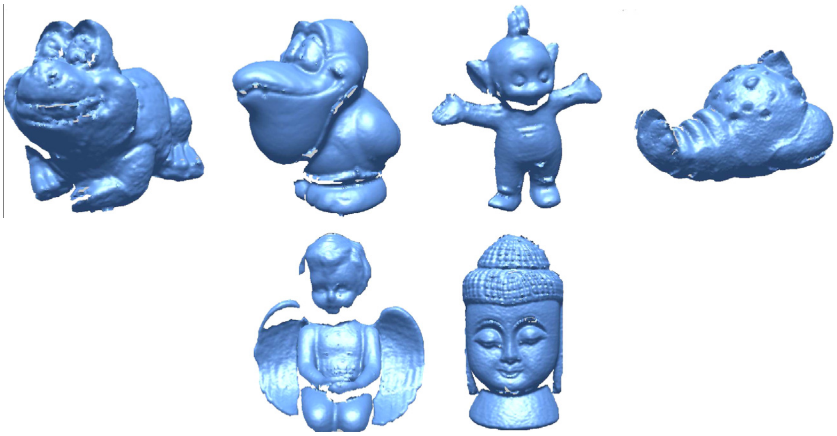

Fig. 7. Range image datasets available at the SAMPL repository. From left to right: “Frog”, “Bird”, “Tele”, “Lobster”, “Angel”, and “Buddha” images.

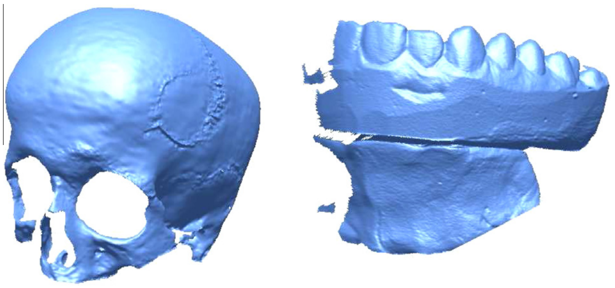

Fig. 8. Specific range image datasets. From left to right: “Skull” and “Tooth” images.

Table 1
Original size of the range images for the considered datasets.

Dataset	Original size (number of points)
Frog	9584
Bird	9051
Tele	6810
Lobster	11683
Angel	14009
Buddha	14801
Skull	76794
Tooth	59033

Table 2
Best performing values of the control parameters of the ABC, BBO, and HS algorithms tackling the RIR problem.

ABC		BBO		HS	
		PopSize	100		
		Elitism size	2		
Colony size (CS)	20	*Habitat modification prob.*	1	*PopSize*	50
Limit	150	*Immigration prob. bounds*	[0,1]	*HMCR*	0.9
n_o, n_e	50% of CS	*Step size*	1	*PAR*	0.2
n_s	1	*Maximum immigration rate*	1	*Step size*	0.1
		Maximum emigration rate	1		
		Mutation prob.	0.05		

Table 3
RIR results of the 20 degrees of overlapping problem scenario.

Dataset	Algorithm	Minimum	Maximum	Mean	Std. dev.
Angel	ABC	**0.2470**	0.5289	**0.3319**	0.1007
	BBO	0.2576	0.9554	0.4975	0.2692
	HS	0.2494	0.9535	0.4412	0.2724
	StEvO	**0.2448**	0.5268	**0.2947**	0.0886
	DeFalco08	0.2493	0.9462	0.6732	0.2209
	Silva05	0.2495	0.9555	0.4179	0.2560
	Yamany99	0.2553	0.9531	0.5818	0.2792
Bird	ABC	**0.1167**	0.9009	**0.3107**	0.2451
	BBO	0.1263	0.9301	0.4347	0.2759
	HS	0.1170	0.9188	0.4671	0.3603
	StEvO	**0.1125**	0.5977	**0.1814**	0.1569
	DeFalco08	0.1245	0.8429	0.4793	0.2157
	Silva05	0.1152	0.9178	0.3506	0.3112
	Yamany99	0.1199	0.9180	0.4465	0.2725
Frog	ABC	**0.1226**	0.7733	**0.2437**	0.1798
	BBO	0.1649	0.8690	0.5396	0.2023
	HS	0.1260	0.8751	0.3476	0.2749
	StEvO	**0.1193**	0.5308	**0.1792**	0.1337
	DeFalco08	0.1322	0.7345	0.4374	0.1615
	Silva05	0.1249	0.8555	0.4329	0.2415
	Yamany99	0.1234	0.8311	0.5119	0.2162
Tele	ABC	**0.0752**	0.8691	**0.1501**	0.1817
	BBO	0.0829	0.8699	0.251	0.2292
	HS	0.0754	0.8721	0.2350	0.2963
	StEvO	**0.0735**	0.8647	**0.1044**	0.1414
	DeFalco08	0.0755	0.6578	0.3193	0.1819
	Silva05	0.0750	0.9234	0.3728	0.3366
	Yamany99	0.0791	0.8958	0.3159	0.2531

Overall, eight different datasets of range images have thus been considered.

Besides, we have defined several pair-wise RIR problem scenarios by using different overlapping degrees between pairs of adjacent images. Specifically, four and six RIR problem instances were considered using pairs of range images of the SAMPL’s datasets at 20 and 40 rotation degrees of misalignment between the adjacent views, and another two more complex instances of pairs of images at 45 and 60 degrees were also considered using the “Skull” and the “Tooth” specific datasets, respectively. Then, we designed four different RIR problem scenarios regarding the rotation degree (20, 40, 45, and 60) from which twelve different problem instances have been generated. Each of those scenarios will be faced by every RIR algorithm.

Moreover, we used a subsampled version of each range image (original sizes are shown in Table 1) in order to speed-up the computation of the objective function (see Eq. (12) in Section 3.2). In particular, five thousand points have been randomly chosen by using a uniform distribution. Feature-based approaches (Gal & Cohen-Or, 2006; Zitová & Flusser, 2003) can also be adopted to achieve a reduced and a characteristic subset of image points, even providing improved IR results (Santamaría et al., 2011). Nevertheless, the latter usually needs the intervention of expert users to obtain high quality features.

4.2. Parameter settings

All the tested algorithms are run on a PC with an Intel Pentium IV 2.6 MHz processor and 2 GB RAM. We maintained the values of the control parameter of each of the four state-of-the-art algorithms (StEvO, deFalco08, Yamany99, and Silva05) as those used in their original contribution facing the IR problem. Regarding the three analyzed algorithms (ABC, BBO, and HS), we carried out some preliminary experiments to carefully tune their control parameters facing the RIR problem. The parameter values, which are finally selected, are given in Table 2.

In order to avoid execution dependence, thirty different runs have been performed for each of the seven tested RIR algorithms when facing each of the four problem scenarios, i.e. considering 20, 40, 45, and 60 degrees of image overlapping. Moreover, all the tested algorithms start from a initial population of random solutions. Each run concerns applying a rigid transformation, randomly generated using an uniform distribution, to the scene image $f_r(I_s)$. In each case, the RIR method will search for this optimal transformation f^* between the proposed image $f_r(I_s)$ and the model image I_m. Every rigid transformation is randomly generated as follows: each of the three rotation axis parameters will be in the range $[-1, 1]$; the rotation angle will range in $[0^\circ, 360^\circ]$; and the range of three translation parameters is $[-40\text{ mm}, 40\text{ mm}]$.

In order to perform a fair comparison among the methods included in this study, we considered CPU time as the stop criterion. Different time limits were tested and 20 s was determined as a good threshold just allowing all the methods to achieve accurate solutions.

4.3. Analysis of results

Tables 3–5 show statistical results regarding the thirty different runs carried out by each of the seven RIR algorithms when facing the four RIR problem scenarios (i.e. 20, 40, 45, and 60 degrees of overlapping). In particular, each column of these tables refer to the range dataset, the algorithm, and the minimum, maximum,

Table 4
RIR results of the 40 degrees of overlapping problem scenario.

Dataset	Algorithm	Minimum	Maximum	Mean	Std. dev.
Angel	ABC	**0.3542**	0.9098	**0.5265**	0.2210
	BBO	0.3690	0.9674	0.6975	0.2435
	HS	0.3553	0.9567	0.6460	0.2665
	StEvO	**0.3493**	0.9436	**0.4990**	0.2175
	DeFalco08	0.3694	0.9599	0.7954	0.1499
	Silva05	0.3527	0.9711	0.6790	0.2640
	Yamany99	0.3623	0.9687	0.7776	0.2057
Bird	ABC	**0.2124**	0.9308	**0.5072**	0.2829
	BBO	0.2426	0.9419	0.7633	0.2189
	HS	0.2165	0.9430	0.6151	0.3058
	StEvO	**0.2041**	0.9168	**0.3741**	0.2655
	DeFalco08	0.2955	0.9350	0.7358	0.1852
	Silva05	0.2159	0.9425	0.5795	0.3158
	Yamany99	0.2776	0.9407	0.7547	0.2070
Frog	ABC	**0.2717**	0.8410	**0.5512**	0.2015
	BBO	0.4794	0.9191	0.7691	0.0823
	HS	0.4026	0.9005	0.7403	0.1161
	StEvO	**0.2517**	0.7717	**0.3941**	0.1856
	DeFalco08	0.3997	0.8000	0.6937	0.0876
	Silva05	0.2735	0.9474	0.6923	0.1750
	Yamany99	0.2809	0.8964	0.7490	0.1220
Tele	ABC	**0.1082**	0.8607	**0.2700**	0.2222
	BBO	0.1198	0.8837	0.4648	0.2300
	HS	0.1095	0.9222	0.4129	0.3072
	StEvO	**0.1054**	0.4708	**0.1682**	0.1226
	DeFalco08	0.1240	0.7722	0.4785	0.1520
	Silva05	0.1077	0.8950	0.5354	0.2929
	Yamany99	0.1104	0.9230	0.4689	0.2686
Buddha	ABC	**0.4473**	0.9446	**0.6690**	0.1220
	BBO	0.6258	0.9481	0.7798	0.0978
	HS	0.5526	0.9285	0.7147	0.1044
	StEvO	**0.3996**	0.6873	**0.5730**	0.1103
	DeFalco08	0.6705	0.9335	0.8105	0.0812
	Silva05	0.5075	0.9506	0.7146	0.1126
	Yamany99	0.6080	0.9259	0.7704	0.0889
Lobster	ABC	0.2745	0.8220	0.6249	0.1530
	BBO	0.3249	0.9406	0.6735	0.1555
	HS	**0.2665**	0.9257	**0.5890**	0.1964
	StEvO	**0.2522**	0.8013	**0.3816**	0.1916
	DeFalco08	0.3392	0.7917	0.6642	0.1020
	Silva05	0.2665	0.9201	0.5727	0.2089
	Yamany99	0.3010	0.8846	0.6530	0.1756

Table 5
RIR results of both the 45 (''Skull'') and the 60 (''Tooth'') degrees of overlapping problem scenarios.

Dataset	Algorithm	Minimum	Maximum	Mean	Std. dev.
Skull	ABC	0.2423	0.8897	0.6526	0.1578
	BBO	0.2401	0.9288	0.6648	0.2215
	HS	**0.2388**	0.9911	**0.6499**	0.2200
	StEvO	**0.2251**	0.7719	**0.3783**	0.2025
	DeFalco08	0.2311	0.8465	0.6677	0.1831
	Silva05	0.2395	0.8451	0.6691	0.1859
	Yamany99	0.2525	0.9416	0.6710	0.2143
Tooth	ABC	0.0433	0.7066	**0.3141**	0.1975
	BBO	0.1105	0.8824	0.5590	0.2057
	HS	**0.0426**	0.7851	0.4202	0.2351
	StEvO	**0.0396**	0.6231	**0.2674**	0.1752
	DeFalco08	0.0454	0.7237	0.4936	0.1922
	Silva05	0.0473	0.7511	0.3958	0.2147
	Yamany99	0.0568	0.8549	0.5488	0.1943

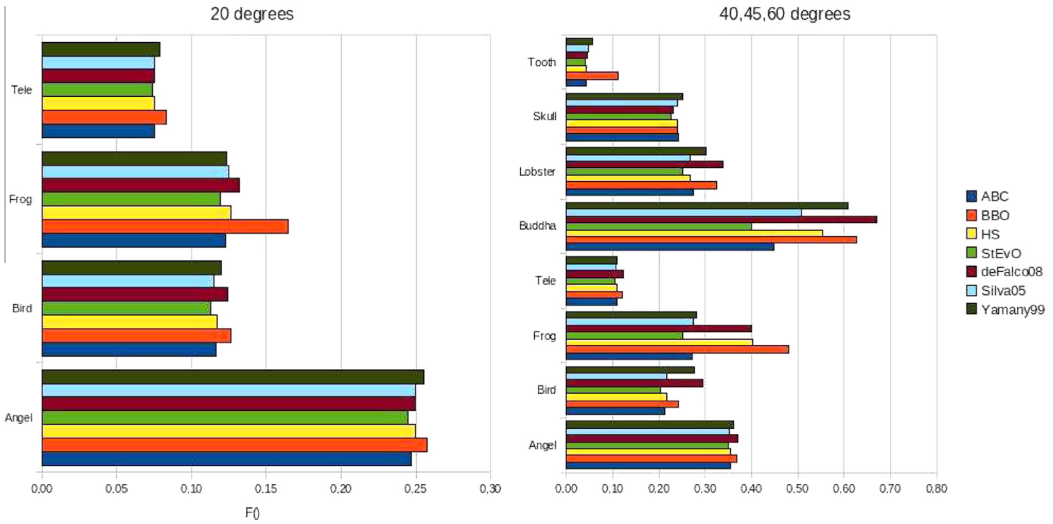

Fig. 9. Bar-graphs comparing the accuracy (according to the minimum value in Tables 3 to 5) of all the tested algorithms on each of the four RIR problem scenarios.

mean and standard deviation values of the *F* function (see Eq. (12)) in those thirty runs. The unit length is always in squared millimeters. Two different algorithm families are distinguished, the one of the three new methods analyzed (ABC, BBO and HS) and that of the four state-of-the-art evolutionary RIR methods. The algorithm with the best (lowest) minimum and mean results in each family is highlighted according to the boldface as well as the overall best value for each dataset is underlined.

Besides, Figs. 9–11 summarize and graphically highlight the latter tabulated data, regarding accuracy (according to minimum values of *F*) and robustness (according to mean and median values of *F*), respectively.

In view of the latter data, we can see how ABC is the most accurate and robust algorithm out of the three new RIR methods which are being analyzed, i.e. BBO and HS. Specifically, ABC achieves the most accurate results in all the instances of the 20 degrees RIR problem scenario and within five out of the six instances of the 40 degrees RIR problem scenario. Nevertheless, it does not behave in that way in the two instances of the 45 and 60 degrees RIR problem scenarios where HS offers a slightly better performance in some cases. In addition, ABC obtains improved outcomes regarding robustness in all the instances of the first scenario, five out of the six instances of the second scenario, and one out of the two instances of the third and fourth scenarios. This fact can also be viewed in Fig. 11, where median values for the thirty different runs are analyzed and ABC achieves the best values in three instances of the 20 degrees problem scenario and five instances of the 40, 45 and 60 degrees scenarios. Overall, it is proven that the ABC algorithm achieves a more suitable trade-off between exploration/exploitation search phases.

Regarding to BBO and HS, we can see how the latter result obtains more accurate outcomes than the former one in all of the twelve addressed RIR problem instances. According to robustness, HS also achieves a better performance than BBO in all the scenarios except in two of the four instances of the 20 degrees RIR problem scenario, i.e. the ''Bird'' and the ''Frog'' instances. Thus, the three analyzed algorithms can be ranked in a decreasing order according to both accuracy and robustness as follows: ABC, HS, and BBO.

When benchmarking ABC results with those of the baseline evolutionary methods from the specific problem being tackled, we can see how the former one obtains a lower performance than the best algorithm in that group, i.e. StEvO, according accuracy and robustness. Nevertheless, we should highlight that ABC achieves outstanding results due to the basic design it proposes compared to the very complex and specific one used by StEvO, which makes use of a more elaborated optimization scheme. However, in comparison to the deFalco08, Silva05, and Yamany99 state-of-the-art evolutionary RIR methods, ABC provided improved outcomes in eleven out of the twelve addressed instances (except in the ''Lobster'' dataset of the 40 degrees problem scenario). Regarding BBO

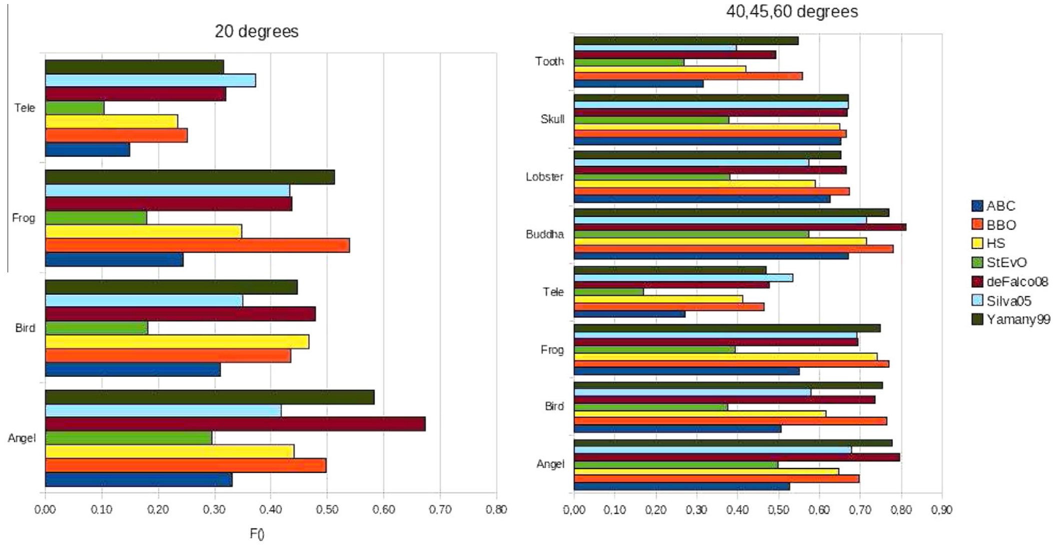

Fig. 10. Bar-graphs comparing the robustness (according to the mean value in Tables 3 to 5) of all the tested algorithms on each of the four RIR problem scenarios.

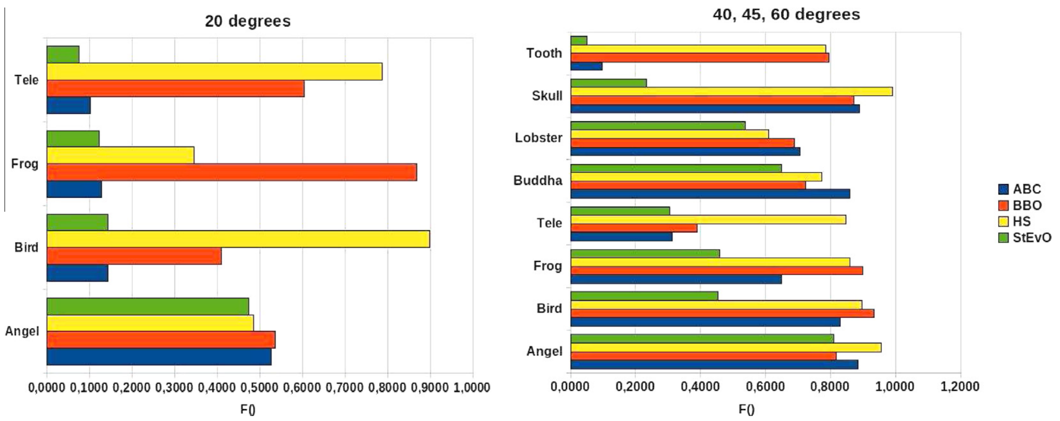

Fig. 11. Bar-graphs comparing the robustness (according to the median value) of ABC, BBO, HS and StEvO on each of the four RIR problem scenarios.

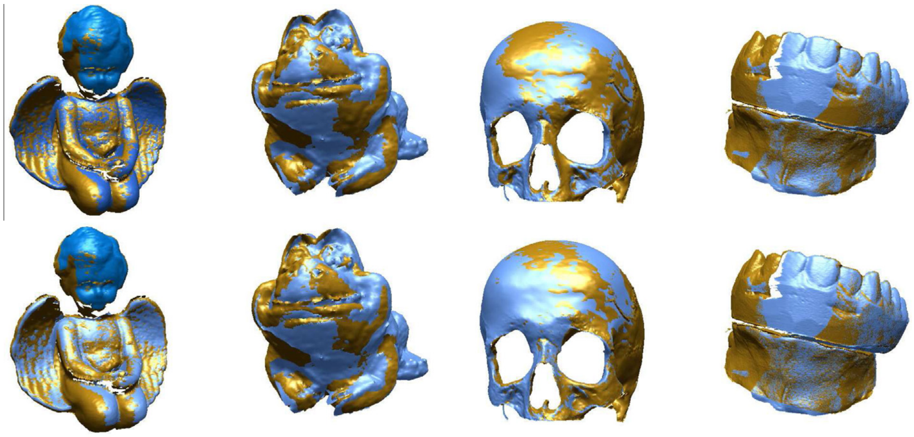

Fig. 12. From left to right: Some of the most accurate RIR results (according to the minimum values in Tables 3–5) obtained by ABC and StEvO shown in the first and the second columns, respectively.

and HS, despite their low performance against ABC and StEvO, the former two algorithms achieve better results than deFalco08, Silva05, and Yamany99 in many of the addressed RIR problem scenarios according to both accuracy and robustness criteria. According to individual accuracy, Fig. 12 shows how ABC, BBO, and HS achieve similar RIR results compared to one of the best evolutionary RIR approaches in the literature, StEvO.

Finally, Table 6 shows the results of the Mann–Whitney U test, also known as Wilcoxon ranksum test, used for a deeper statistical study of the results. This is motivated by those situations in which

Table 6
Mann–Whitney U paired-test results (10% significance level) comparing all the evolutionary RIR methods in the twelve problem instances.

	ABC	BBO	HS	StEvO	DeFalco08	Silva05	Yamany09
ABC	○	+	+	–	+	+	+
BBO	–	○	–	–	=	=	=
HS	–	+	○	–	+	=	+
StEvO	+	+	+	○	+	+	+
DeFalco08	–	=	–	–	○	–	=
Silva05	–	=	=	–	+	○	+
Yamany09	–	=	–	–	=	–	○

simply by using the visual inspection procedure does not provide good insights of the real performance of the algorithms, e.g. the outcomes shown in Fig. 12. Unlike the commonly used t-test, the Wilcoxon test does not assume normality of the samples, which would be unrealistic for the data in our real-world application (Lehmann, 1975). The significant results are comprised by three symbols:'+' meaning the significance is favorable to the method in the row; '–' stands for the significance favorable to the method in the column; and '=' is used when there is not significance favorable (not relevant) to any of the couple of methods which are being compared. As reported above, the three tested algorithms (ABC, BBO, and HS) achieve competitive RIR outcomes compared to the state-of-the-art methods. Specifically, remarkable significant results are drawn by ABC, which is only outperformed by the best method, StEvO.

5. Concluding remarks

In this work we performed a first step in the analysis of some cutting edge population-based optimization algorithms tackling continuous optimization problems. In particular, the ABC, BBO, and HS algorithms have been considered in this study due to their innovative designs. A computational analysis of their performance has been carried out facing IR, a challenging and well-known NP-hard problem within the computer vision field.

A benchmark suite composed by several instances of the pair-wise IR problem using range images has been considered to testing the ABC, BBO, and HS algorithms and benchmarking their results with several state-of-the-art evolutionary IR methods proposed to date as well. Promising results have been obtained by these three recent population-based approaches when tackling complex IR problem instances of range images. Specifically, ABC achieved the best outcomes according to accuracy and robustness regarding BBO and HS, which is motivated by the good trade-off between exploration and exploitation of the search, also promoted by the former algorithm. Regarding the latter two methods, it has been shown how HS improves BBO according to accuracy in all the problem instances addressed, as well as it also outperforms BBO in terms of robustness in most of the problem instances. Then, these algorithms can be ranked in a decreasing order according to their performance, tackling the IR problem as follows: ABC, HS, BBO. This seems to suggest there is a relationship between the design complexity level of these approaches and their performance. As it has been recently proven in the literature, those approaches based on easy-to-implement schemes usually provide competitive outcomes compared to the most elaborated ones, e.g. ABC vs. StEvO, despite the best results of the latter method are due to its specific design considered to this particular real-world application. Moreover, it can be shown how ABC outperforms the performance obtained by several of the state-of-the-art IR algorithms.

Finally, we conclude that there is room for improvement for all the three analyzed algorithms from different points of view. In particular, the synergy between global and local search strategies (Santamaría et al., 2009) will promote the achievement of more accurate results, as it has been proven in memetic algorithms (Moscato & Norman, 1992; Krasnogor & Smith, 2005; Santamaría et al., 2009). Besides, hybridizations between the ABC, BBO, and HS algorithms with other MHs (Glover & Kochenberger, 2003; Luke, 2009) would be another promising alternative to the improvement of any of the former innovative optimization approaches.

Acknowledgments

We want to acknowledge all the team of the Physical Anthropology lab at the University of Granada (headed by Dr. Botella and Dr. Alemán) for their support during the acquisition of the specific range datasets "Skull" and "Tooth". This work is partially supported by both the Spanish Ministerio de Educación y Ciencia (Ref. TIN2009-07727) including EDRF fundings and the University of Jaén (Ref. R1/12/2010/61) including fundings from *Caja Rural de Jaén*.

References

Alvarado-Iniesta, A., García-Alcaraz, J. L., Rodríguez-Borbón, M. I., & Maldonado, A. (2013). Optimization of the material flow in a manufacturing plant by use of artificial bee colony algorithm. *Expert Systems with Applications, 40*(12), 4785–4790.

Bäck, T. (1996). *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press.

Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997). *Handbook of evolutionary computation*. IOP Publishing Ltd and Oxford University Press.

Bernardini, F., & Rushmeier, H. (2002). The 3D model acquisition pipeline. *Computer Graphics Forum, 21*(2), 149–172.

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14*, 239–256.

Bhattacharya, A., & Chattopadhyay, P. (2011). Hybrid differential evolution with biogeography-based optimization algorithm for solution of economic emission load dispatch problems. *Expert Systems with Applications, 38*(11), 14001–14010.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence*. Oxford University Press.

Campbell, R. J., & Flynn, P. J. (2001). A survey of free-form object representation and recognition techniques. *Computer and Vision Image Understanding, 81*(2), 166–210.

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing, 10*(3), 145–155.

Chu, S., & Tsai, P. (2007). Computational intelligence based on the behavior of cats. *International Journal of Innovative Computing, Information and Control, 3*(1), 163–173.

Clerc, M. (2006). *Particle swarm optimization*. ISTE Publishing Company.

Cui, Z., Zeng, J., & Sun, G. (2006). A fast particle swarm optimization. *International Journal of Innovative Computing, Information and Control, 2*, 1365–1380.

Damas, S., Cordón, O., & Santamaría, J. (2011). Medical image registration using evolutionary computation: An experimental survey. *Computational Intelligence Magazine, IEEE, 6*(4), 26–42.

de Falco, I., Della Cioppa, A., Maisto, D., & Tarantino, E. (2008). Differential Evolution as a viable tool for satellite image registration. *Applied Soft Computing, 8*(4), 1453–1462.

Dorigo, M., & Di Caro, G. (1999). The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New ideas in optimization* (pp. 11–32). Nueva York, NY: McGraw-Hill. UU. EE.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. MIT Press.

Farhaana, Y., Zuwairie, I., Amar Faiz Zainal, A., Zulkifli Md., Y., Mohd Saberi, M., Khairunizam, W., et al. (2012). Self-evolving ant colony optimization and its application to traveling salesman problem. *International Journal of Innovative Computing, Information and Control, 8*(10B), 7329–7339.

Fitzpatrick, J.,Grefenstette, J., & Gucht, D. 1984. Image registration by genetic search. In IEEE Southeast Conference, Louisville, EEUU (pp. 460–464).

Fogel, D. (2005). *Evolutionary computation: Toward a new philosophy of machine intelligence*. Wiley-IEEE Press.

Gal, R., & Cohen-Or, D. (2006). Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics, 25*(1), 130–150.

Geem, Z. W. 2007. Harmony search algorithm for solving Sudoku. In *Proceedings of the 11th international conference, KES 2007 and XVII Italian workshop on neural networks conference on Knowledge-based intelligent information and engineering systems: Part I*, ISBN 978-3-540-74817-5 (pp. 371–378).

Geem, Z. W., 2013. Harmony Search algorithm homepage. <https://sites.google.com/a/hydroteq.com/www/>.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm, Simulation 76 (pp. 60–68).

Gil-López, S., Ser, J. D., Salcedo-Sanz, S., Pérez-Bellido, A. M., Cabero, J. M., & Portilla-Figueras, J. A. (2012). A hybrid harmony search algorithm for the spread spectrum radar polyphase codes design problem. *Expert Systems with Applications, 39*(12), 11089–11093.

Glover, F., & Kochenberger, G. A. (Eds.). (2003). *Handbook of metaheuristics*. Kluwer Academic Publishers.

Godin, G., Hebert, P., Masuda, T., & Taubin, G. (2009). Special issue on new advances in 3D imaging and modeling. *Computer and Vision Image Understanding, 113*(11), 1105–1180.

Goldberg, D. E. (1989). *Genetic algorithms in search and optimization*. New York, EEUU: Addison-Wesley.

Jaberipour, M., & Khorram, E. (2010). Solving the sum-of-ratios problems by a harmony search algorithm. *Journal of Computational and Applied Mathematics, 234*(3), 733–742.

Karaboga, N. (2009). A new design method based on artificial bee colony algorithm for digital IIR filters. *Journal of the Franklin Institute*, 0016-0032*, 346*(4), 328–348.

Karaboga, D. 2013. Artificial Bee Colony (ABC) algorithm homepage. <http://mf.erciyes.edu.tr/abc/index.htm>.

Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 0096-3003*, 214*(1), 108–132.

Karaboga, D., & Basturk, B. (2007). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In *Proceedings of the 12th international fuzzy systems association world congress on foundations of fuzzy logic and soft computing, IFSA '07*, ISBN:978-3-540-72917-4. (pp. 789–798).

Karaboga, D., & Basturk, B. (2007b). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 0925-5001*, 39*, 459–471.

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony ABC algorithm. *Applied Soft Computing*, 1568-4946*, 8*, 687–697.

Kennedy, J., & Eberhart, R. (2001). *Swarm intelligence*. San Francisco, CA: Morgan Kaufmann.

Krasnogor, N., & Smith, J. (2005). A Tutorial for competent memetic algorithms: model, taxonomy and design issues. *IEEE Transactions on Evolutionary Computation, 9*(5), 474–488.

Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering, 194*(36–38), 3902–3933.

Lehmann, E. (1975). *Nonparametric statistical methods based on ranks*. New York NY: McGraw-Hill.

Liu, Y. (2004). Improving ICP with easy implementation for free form surface matching. *Pattern Recognition, 37*(2), 211–226.

Luke, S. 2009. Essentials of metaheuristics, Lulu. Available at <http://cs.gmu.edu/~sean/book/metaheuristics/>.

MacArthur, R., & Wilson, E. (1967). *The theory of biogeography*. Princeton, New Jersey: Princeton University Press.

Masuda, T. (2002). Registration and integration of multiple range images by matching signed distance fields for object shape modeling. *Computer and Vision Image Understanding, 87*(1–3), 51–65.

Ma, H. (2010). An analysis of the equilibrium of migration models for biogeography-based optimization, Information Science 180.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs*. Springer-Verlag.

Moscato, P., & Norman, M. G. (1992). A memetic approach for the traveling salesman problem – implementation of a computational ecology for combinatorial optimization on message-passing systems. In *International conference on parallel computing and transputer applications*. Amsterdam: IOS Press.

Omran, M. G., & Mahdavi, M. (2008). Global-best harmony search. *Applied Mathematics and Computation, 198*(2), 643–656.

Ovreiu, M., & Simon, D. 2010. Biogeography-based optimization of neuro-fuzzy system parameters for diagnosis of cardiac disease. In *Proceedings of the 12th annual conference on genetic and evolutionary computation, GECCO '10* (pp. 1235–1242).

Price, K. (1999). An introduction to differential evolution. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New ideas in optimization* (pp. 79–108). Cambridge, UK: McGraw-Hill.

Rarick, R., Simon, D., Villaseca, F. E., & Vyakaranam, B. 2009. Biogeography-based optimization and the solution of the power flow problem. In Proceedings of the 2009 ieee international conference on systems, man and cybernetics. ISBN 978-1-4244-2793-2 (pp. 1003–1008).

Rodrigues, M., Fisher, R., & Liu, Y. (2002). Special issue on registration and fusion of range images. *Computer and Vision Image Understanding, 87*(1–3), 1–7.

Rusinkiewicz, S., & Levoy, M. 2001. Efficient variants of the ICP algorithm. In *Third international conference on 3D digital imaging and modeling (3DIM'01)*, Quebec, Canada (pp. 145–152).

Salvi, J., Matabosch, C., Fofi, D., & Forest, J. (2007). A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing, 25*(5), 578–596.

Santamaría, J., Cordón, O., & Damas, S. (2011). A comparative study of state-of-the-art evolutionary image registration methods for 3D modeling. *Computer and Vision Image Understanding, 115*, 1340–1354.

Santamaría, J., Cordón, O., Damas, S., García-Torres, J., & Quirin, A. (2009). Performance evaluation of memetic approaches in 3D reconstruction of forensic objects. *Soft Computing, 13*(8–9), 883–904.

Santamaría, J., Damas, S., García-Torres, J. M., & Cordón, O. (2012). Self-adaptive evolutionary image registration using differential evolution and artificial immune systems. *Pattern Recognition Letters*, 0167-8655*, 33*(16), 2065–2070.

Sharp, G. C., Lee, S. W., & Wehe, D. K. (2002). ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(1), 90–102.

Silva, L., Bellon, O. R. P., & Boyer, K. L. (2005). Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(5), 762–776.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation, 12*(6), 702–713.

Simon, D. 2013. Biogeography-Based Optimization algorithm homepage. <http://embeddedlab.csuohio.edu/BBO/>.

Singh, A. (2009). An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing*, 1568-4946*, 9*, 625–631.

Song, Y., Liu, M., & Wang, Z. 2010. Biogeography-based optimization for the traveling salesman problems. In *Third international joint conference on computational science and optimization (CSO), 2010* (vol. 1, pp. 295–299).

Storn, R. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 341–359.

Yamany, S. M., Ahmed, M. N., & Farag, A. A. (1999). A new genetic-based technique for matching 3D curves and surfaces. *Pattern Recognition, 32*, 1817–1820.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision, 13*(2), 119–152.

Zhou, X., & Wang, R. (2012). Self-evolving ant colony optimization and its application to traveling salesman problem. *International Journal of Innovative Computing, Information and Control, 8*(12), 8311–8321.

Zitová, B., & Flusser, J. (2003). Image registration methods: A survey. *Image and Vision Computing, 21*, 977–1000.