

Article

New Evolutionary-Based Techniques for Image Registration

Catalina-Lucia Cocianu and Alexandru Stan * 

Computer Science Department, Bucharest University of Economics, 010374 Bucharest, Romania;
catalina.cocianu@ie.ase.ro

* Correspondence: alexandru.stan1@yahoo.com

Received: 12 December 2018; Accepted: 29 December 2018; Published: 5 January 2019



Abstract: The work reported in this paper aims at the development of evolutionary algorithms to register images for signature recognition purposes. We propose and develop several registration methods in order to obtain accurate and fast algorithms. First, we introduce two variants of the firefly method that proved to have excellent accuracy and fair run times. In order to speed up the computation, we propose two variants of Accelerated Particle Swarm Optimization (APSO) method. The resulted algorithms are significantly faster than the firefly-based ones, but the recognition rates are a little bit lower. In order to find a trade-off between the recognition rate and the computational complexity of the algorithms, we developed a hybrid method that combines the ability of auto-adaptive Evolution Strategies (ES) search to discover a global optimum solution with the strong quick convergence ability of APSO. The accuracy and the efficiency of the resulted algorithms have been experimentally proved by conducting a long series of tests on various pairs of signature images. The comparative analysis concerning the quality of the proposed methods together with conclusions and suggestions for further developments are provided in the final part of the paper.

Keywords: image registration; image recognition; evolutionary computing; evolution strategies; firefly technique; hybrid techniques; mutual information; affine perturbation

1. Introduction

One of the most challenging tasks of image recognition is to align images acquired at different times, by different sensors and from different angles. The registration process consists of finding correspondence between all pixels in two images of a specific scene. Registration methods are applied in various real-world problems, including object detection and recognition, motion analysis, change detection and object tracking.

Basically, image registration techniques deal with geometric distortions, the ultimate goal being to find out a set of transformation parameters that maximize the similarity degree between the images to be aligned. The class of perturbation models addressed by registration includes spatial transformations as for instance rigid, affine, projective, and global polynomial perturbation functions. The research reported in our paper involves the most commonly studied geometric degradation model, namely the affine transformation, defined as a mixture of translation, rotation, scale changes, shear and aspect ratio changes.

So far, various classes of registration techniques, mainly depending on the perturbation model, have been presented in the literature [1–7]. The most commonly used classes of registration methods include Principal Axes Transform (PAT), multiresolution registration, boundary registration, model-based registration, adaptive registration and optimization-based registration. PAT method belongs to PCA-based image processing techniques. Basically, the main advantages of using principal components (PCs) in digital signal analysis is that PCs are uncorrelated, with the information encoded

in each one of them being the maximum for the whole set [8]. Multiresolution methods align two images using sets of increasingly smaller size representations. Since smaller images reduce geometric differences, a global transformation function successfully registers the images and speeds up the registration process. Boundary registration uses the idea of aligning images using their boundaries. The adaptive registration techniques are developed based on a collection of tools and use information about the geometric and intensity differences between the images to select the right tool for alignment. The model-based methods are applied when traditional dissimilarity measures cannot be used to find an initial registration of the images. In such cases, additional information about the images should guide the alignment process.

The techniques proposed in this paper belong to the optimization class and consist of evolutionary computation (EC) approaches. The EC approaches of image registration are usually three stage mechanisms. First, a similarity measure which can be associated to the fitness function has to be defined. In the second stage the initial parameter values that approximately register the images are automatically established by randomly generating them in a certain search space. Finally, an EC-based algorithm that takes the initial variants of registration to the optimal one is applied. The field of EC consists of genetic algorithms (GA), evolution strategies (ES), genetic programming, differential evolution, evolutionary programming and swarm intelligence. So far, the most commonly used EC methods for image registration includes GAs, memetic algorithms and particle swarm optimization [9–14].

Our research focuses on the development of image registration techniques for signature recognition. In our work we consider the degradation model that combines the rigid transformation with a shear transformation acting along the x axis. We propose and develop several EC registration methods in order to obtain accurate recognition rates and fast algorithms. First, we introduce two variants of the firefly method that proved to have excellent accuracy and fair run times. In order to speed up the computation, we propose two variants of Accelerated Particle Swarm Optimization (APSO) method. The resulted algorithms are significantly faster than the first ones, but the recognition rates are a little bit lower as compared to firefly techniques. Finally, we propose new hybrid models that combine evolution strategies and APSO methods in order to develop fast and efficient registration algorithms.

The rest of the paper is organized as follows. The proposed general EC approach of image registration developed based on the considered degradation model is supplied in Section 2. Next, a brief description of two standard EC methods, namely firefly algorithm and evolution strategies, is provided. The proposed variants of the firefly method and a new hybrid EC approach of image registration are introduced in the core section of the paper. A series of experimental results and the comparative analysis concerning the accuracy and the efficiency of the proposed algorithms are exposed in Section 6. The final part of the paper includes conclusions and suggestions for further developments regarding EC solutions for image registration.

2. The Proposed Evolutionary Computing General Framework for Image Registration

The main idea of performing image registration task using a particular EC approach is quite simple and can be described as follows. Let us denote by I_1 the reference image and we assume that the sensed image I_2 is defined by

$$I_2(x, y) = I_1(x_1, y_1) \quad (1)$$

$$(x_1, y_1) = T_{SP}(x, y) \quad (2)$$

where T_{SP} is the geometric transformation that depends on a particular set of parameters, SP . We assume that T_{SP} is invertible and the analytic expressions of T_{SP} and T_{SP}^{-1} are known. The attempt to align I_2 to I_1 consists of computing the transformation parameters SP and applying the inverse T_{SP}^{-1} ,

$$(x, y) = T_{SP}^{-1}(x_1, y_1) \quad (3)$$

The computation of the transformation parameters can be carried out by an evolutionary algorithm (EA), where the search space is established based on SP, the fitness is defined in terms of a similarity/dissimilarity function computed either for the pair $(I_1, I_2(\check{T}_{SP}^{-1}))$ or for $(I_2, I_1(\check{T}_{SP}))$, and \check{T}_{SP} corresponds to the computed values of the parameters.

One of the most commonly used degradation models corresponds to the affine transformation. The main geometric transformations involved with an affine function are rigid spatial transformations, the shear effect acting along axes and changes in the aspect ratio.

The rigid transformation can be described in terms of translation, rotation and scale changes. Its main property is that the objects in images preserve their relative shape and dimensions. The 2D rigid transformation having as inputs the image I_1 and the parameters (a, b, s, θ) produces the output I_2 defined by

$$I_2(x, y) = I_1(x_1, y_1) \tag{4}$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} + s \cdot \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{5}$$

where $\begin{bmatrix} a \\ b \end{bmatrix}$ defines the translation, s stands for the scale factor and $R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ is the rotation matrix.

The shear effect acting along axes is defined by

$$F_x = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix}, F_y = \begin{bmatrix} 1 & 0 \\ h & 1 \end{bmatrix} \tag{6}$$

while the changes in aspect ratio are given by the matrix

$$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \tag{7}$$

The affine transformations are obtained by applying any sequence of rigid transformations, shears and changes in the aspect ratio [15].

In our work we use the perturbation model given by a sequence of a shear transformation, acting along the x axis, and a rigid function

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \cdot \left(\begin{bmatrix} a \\ b \end{bmatrix} + s \cdot \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \right) \tag{8}$$

In this case, the set of transformation parameters is $SP = \{a, b, s, \theta, d\}$. Consequently, an EA should operate on chromosomes having their solution part represented by a five sized real-valued vector. Each candidate solution corresponds to a potential transformation parameter $(a_{sol}, b_{sol}, s_{sol}, \theta_{sol}, d_{sol})$. In the case of self-adaptive procedures, the chromosome representation should include a parameter part, the additional alleles being related to the mutation operator.

Various similarity/dissimilarity measures have been reported so far, each one having its own strengths and shortcomings. In our developments we use one of the most studied similarity measures for solving the image registration problem, namely the Shannon mutual information (MI) method [16–18]. The main drawback of any MI-based EA that processes arbitrary grey level images is its computational complexity. In order to solve the complexity problem involved by the MI quality evaluation, we use the mutual information computed between the binary variants of the reference I_1 and the transformed image $I_2(\check{T}_{SP}^{-1})$ respectively.

Let I_1, I_2 be $M \times N$ binary images. The mutual information between I_1 and I_2 is defined by [19]:

$$MI(I_1, I_2) = H(I_1) + H(I_2) - H(I_1, I_2) \tag{9}$$

where $H(\cdot)$ is the Shannon entropy, and $H(I_1, I_2)$ is the joint entropy

$$H(I) = - \sum_{i=1}^2 p_i \cdot \log(p_i) \tag{10}$$

$$H(I_1, I_2) = - \sum_{1 \leq i, j \leq 2} p(i, j) \cdot \log(p(i, j)) \tag{11}$$

Since

$$MI(I_1, I_2) \leq \min(H(I_1), H(I_2)) \tag{12}$$

we get

$$\max_J MI(I, J) = MI(I, I) = H(I) \tag{13}$$

We define the fitness function of each genotype g corresponding to the transformation \check{T}_{SP} , by

$$\text{fitness}(g) = \frac{MI(I_1, I_g)}{MI(I_1, I_1)} = \frac{MI(I_1, I_g)}{H(I_1)} \tag{14}$$

where $I_g = I_2(\check{T}_{SP}^{-1})$. Note that $\text{fitness}(g) \leq 1$ and the maximum value of the fitness function is obtained when $I_g = I_1$, which is equivalent to $I_g(\check{T}_{SP}) = I_2$.

3. The Firefly Algorithm

The Firefly algorithm developed by Xin-She Yang [20] belongs to a meta-heuristic nature inspired class of algorithms, based on swarm intelligence, being typically used to solve a variety of optimization problems in real-world applications, from project scheduling problems to rich vehicle routing, and from image processing to engineering machining parameters optimization [21–24].

Let us consider the optimization problem

$$\max_{x \in D} F(x) \tag{15}$$

The firefly algorithm is based on the flashing behavior, bioluminescence or flashing signals, of fireflies and it relies on three idealized rules underlying the mathematical model. First, the intensity of bioluminescence represents an indicator of fitness for the firefly. The brightness of a firefly is influenced by F and it is determined by the landscape of F . The light intensity is proportional to the value of the objective function F . The second rule is based on the attractiveness of the individuals, which is directly proportional to the light intensity. This means that a less bright firefly will move to the brighter one. The brightness of the firefly is directly influenced by the distance between fireflies due to the fact that the air absorbs light. Finally, the last rule states that all fireflies of the swarm are unisexual, therefore they attract one another regardless of gender.

The main features of the standard firefly algorithm are the attractiveness, the movement and the distance. The attractiveness of a firefly is proportionally influenced by light intensity, which itself is associated with the objective function and this attractiveness decreases as the distance between individuals increases. From the mathematical point of view, this idea is expressed by the following monotonically decreasing function

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{16}$$

where β_0 is the attractiveness when the distance between two fireflies is zero ($r = 0$), γ is the light absorption coefficient controlling the decrease of light intensity, and r is the distance between two fireflies.

The movement of a firefly i toward a more attractive, brighter, firefly j is defined by

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \cdot \varepsilon \tag{17}$$

where x_i represents the current position of firefly i , the second term represents the attraction to the light intensity while ε is randomly generated from $U(0, 1)$.

In the following we refer to a firefly i by its current position, x_i .

Let n be the number of the fireflies, MaxGeneration the maximum generations, β_0 the initial attractiveness, γ the light absorption coefficient, and α controlling the randomness. The standard version of the firefly algorithm is described in the Algorithm 1.

Algorithm 1 Firefly algorithm

- 1: Randomly generate an initial population $X_0 = \{x_1, x_2, x_3, \dots, x_n\}$
 - 2: Compute L_i , the light intensity of each $x_i \in X_0$; $t \leftarrow 0$
 - 3: **while** ($t < \text{MaxGeneration}$) **do**
 - 4: **for** $i = 1 : n$
 - 5: **for** $j = 1 : n$
 - 6: **if** $L_j > L_i$
 - 7: Move firefly x_i toward firefly x_j using (17)
 - 8: **end if**
 - 9: **end for**
 - 10: **end for**
 - 11: Evaluate all candidate solutions and update the light intensity values
 - 12: Rank the fireflies and identify the current best solution
 - 13: $t \leftarrow t + 1$
 - 14: **end while**
 - 15: Rank all fireflies and return the best one
-

In most of the cases the α step is static or linear with its value decreasing from iteration to iteration, each firefly from generation having the same step value. For instance, the parameter α is updated after each generation by [25]

$$\alpha_t = \alpha_{t-1} \cdot \alpha_{\text{damp}} \tag{18}$$

where $\alpha_{\text{damp}} \in (0, 1)$. Usually $\alpha_{\text{damp}} \geq 0.9$.

Note that if the firefly x_j is replaced by the current best individual, g , we obtain a simplified version of firefly algorithm, namely the APSO method. APSO accelerates the convergence and reduces the complexity effort of the algorithm, but also can decrease its accuracy [26]. The updating rule of APSO is expressed by

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (g - x_i) + \alpha \cdot \varepsilon \tag{19}$$

One of the most important steps in developing the firefly algorithm is the parameter setting. Various algorithms can be obtained based on the previous parameters, including Differential Evolution, Simulated Annealing, Harmony Search and Particle Swarm Optimization [27]. In the standard firefly algorithm one of the factors that influence the diversity of the population is the randomization $\alpha \cdot \varepsilon$, where α is the randomization parameter. In general the α step is static or linear with its value decreasing from iteration to iteration, each firefly from generation having the same step value. If α has a small value, the current solution may be trapped in the local optimum, resulting in a premature convergence. Large values of randomness parameter may lead to divergent algorithms. The parameter α directly affects the exploration and exploitation of the algorithm. The value of α should rapidly

decrease through the first generations to explore new search space, while it should slightly vary from one iteration to another in the last part of the algorithm.

4. Evolution Strategies

The evolution strategies (ESs) represent one of the major paradigms of the evolutionary algorithm, used in a wide variety of optimization tasks and typically continuous but also discrete, combinatorial search spaces with and without constraints [28]. One of the main features of ESs is self-adaptability, a mechanism that adjusts the parameters of the exploration distribution during the search process [29]. In most of the cases, the self-adaptation is strictly related to the mutation mechanism.

In ESs the mutation process represents the main variation operator and usually implements the following strategies: uncorrelated mutation with one step, uncorrelated mutation with n steps and correlated mutations respectively.

The first mutation procedure uses a single standard deviation parameter, or step size, to compute each allele of the resulted chromosome. The uncorrelated mutation with n steps deals with possible different step sizes to affect different alleles in a chromosome in order to treat dimensions differently. This way, the mutation procedure introduces different standard deviations for each axis, the resulted offspring being placed on an ellipse around the individual to be mutated, orthogonal to the axes. In case of the correlated mutation, the resulted offspring belong to an ellipse having any orientation by rotating it with a rotation matrix given by a covariance parameter. Mutation mainly depends on the mutation strength parameter [30]. At the beginning of the optimization process, the value of this parameter should be large to ensure the exploring of the search space and later, in the vicinity of the optimum point, the value of the parameter should decrease to ensure the exploitation phase.

In the ES the phenotype space is typically R^n . In this case, the genotype and the phenotype coincide and the representation of the chromosome is straightforward, with no codification being needed. If the search procedure implements self-adaptation, each candidate solution c is defined in terms of solution part c_{sol} and parameter part c_{part} [29]. For instance, in the case of the standard uncorrelated multistep mutation, the chromosome representation is as follows

$$c = (c_{sol}, c_{part}) \tag{20}$$

$$c_{sol} = (x_1, x_2, \dots, x_n) \tag{21}$$

$$c_{part} = (\sigma_1, \sigma_2, \dots, \sigma_n) \tag{22}$$

where, for each i , $1 \leq i \leq n$, σ_i is the step size of mutation.

The algorithm uses a population P of μ potential solutions named chromosomes, which are randomly generated in the first generation. Usually, the size of population is fixed. In the evolution strategies, the parent selection is not influenced by the fitness function, meaning that the entire population could generate offspring solutions. The size of offspring population should be larger than the parent set, the recommended size being $\lambda \cong 7 \cdot \mu$.

The crossover procedure creates new individuals, mixing the genetic material of parents. The recombination procedures are either local or global and use discrete and respectively intermediate crossover mechanisms. In case of the local recombination, a new offspring is generated using two parents randomly selected from population. In contrast to this, in case of the global recombination, each allele of the new offspring is obtained using a pair of parents. The resulted child has multiple parents.

In our hybrid approach described in Section 5.2, the standard uncorrelated multistep mutation procedure is used to add some randomness to the solution. The mutated version of a chromosome $c = (x_1, \dots, x_n, \sigma_1, \dots, \sigma_n)$ is the genotype $c' = (x'_1, \dots, x'_n, \sigma'_1, \dots, \sigma'_n)$ computed as follows:

$$\sigma'_i = \sigma_i \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)} \tag{23}$$

$$x'_i = x_i + \sigma'_i \cdot N_i(0, 1) \tag{24}$$

where σ_i represents the mutation step, $N(0, 1)$ represent a random number generated from normal distribution, τ' and τ stand for the learning steps

$$\tau' \simeq 1/\sqrt{2n}, \tau \simeq 1/\sqrt{2\sqrt{n}} \tag{25}$$

Note that, in order to keep the step sizes above a certain threshold, one can apply the rule

$$\text{if } \sigma'_i < \varepsilon_\sigma \text{ then } \sigma'_i \leftarrow \varepsilon_\sigma \tag{26}$$

In the ES algorithm, the survivor selection mechanism is deterministic. In case of (μ, λ) selection, the next generation of μ chromosomes are selected from offspring population only. In case of $(\mu + \lambda)$ the best μ chromosomes are chosen from both the parent population and the children multiset [28].

The general ES scheme is described in the Algorithm 2.

Algorithm 2 ES algorithm

- 1: $t \leftarrow 0$
 - 2: Initialization—Randomly generate initial population— $P_t \in \{c_1, c_2, \dots, c_n\}$
 - 3: Evaluate P_t
 - 4: **while** Termination Condition does not hold **do**
 - 5: Use recombination and mutation operator to obtain new offspring solutions $O_t \in \{c'_1, c'_2, \dots, c'_\lambda\}$
 - 6: Evaluate O_t
 - 7: Rank and select the μ best chromosomes using (μ, λ) or $(\mu + \lambda)$ to obtain P_{t+1}
 - 8: $t \leftarrow t + 1$
 - 9: **end while**
-

5. The Proposed Evolutionary Algorithms for Image Registration

In order to solve the image registration problem, we propose two variants of the firefly algorithm together with their corresponding APSO versions. Also, a hybridization of APSO algorithm with an ES-based technique is supplied next.

5.1. The Proposed Variants of Firefly Algorithm and APSO

In the following we describe two proposed updating rule variants to develop new algorithms based on the general scheme of the standard firefly algorithm and APSO mechanism respectively.

Each candidate solution (firefly) is a D-dimensional real valued vector whose entries are the values of a potential transformation parameter. Note that if the search space corresponds to the degradation model (8) then $D = 5$. We considered the fixed-size model, with each current population having n chromosomes, $X = \{x_1, x_2, x_3, \dots, x_n\}$,

$$x_i = (x_i(1), x_i(2), \dots, x_i(D)) \tag{27}$$

Also, we assume that for each candidate solution x_i

$$x_i(k) \in [lb(k), hb(k)], k = 1, \dots, D \tag{28}$$

In other words, each transformation parameter belonging to SP is assumed to be bounded by certain given values, the search space being defined by

$$S = \prod_{k=1}^D [lb(k), hb(k)] \tag{29}$$

The initial firefly population, X_0 , is randomly generated according to

$$x_i(k) = (hb(k) - lb(k)) \cdot d + lb(k) \tag{30}$$

where d is a random number generated from uniform distribution $U(0, 1)$.

We introduce two modalities to define the randomness parameter α and we compute the dissimilarity between each component of two different individuals instead of the classic Euclidian distance to deal with the attributes diversity.

First, we take into account the search space bounds to define the updating rule

$$x_i(k) = x_i(k) + \beta_{ij}(k) \cdot (x_j(k) - x_i(k)) + \frac{hb(k) - lb(k)}{\max_k (hb(k) - lb(k))} \cdot \epsilon \cdot c \tag{31}$$

$$\beta_{ij}(k) = \beta_0 e^{-\gamma r(k)_{ij}^2} \tag{32}$$

$$x_i(k) = (hb(k) - lb(k)) \cdot d + lb(k) \tag{33}$$

where c is a given constant scale factor, ϵ is randomly generated from the uniform distribution $U(0, 1)$. Note that the factor β_{ij} depends on the value of k . The vector r measures the differences between x_i and x_j , its k th component being given by

$$r_{ij}(k) = \frac{|x_i(k) - x_j(k)|}{D_{max}(k)} \tag{34}$$

where

$$D_{max}(k) = \frac{hb(k) - lb(k)}{\sqrt{D}} \tag{35}$$

The updating rule (31) modifies each component k of the individual x_i based on its corresponding range.

The second proposed updating rule takes into account both the variable ranges and the quality of the attractor

$$x_i(k) = x_i(k) + \beta_{ij}(k) \cdot (x_j(k) - x_i(k)) + \frac{hb(k) - lb(k)}{\max_k (hb(k) - lb(k))} \cdot \epsilon \cdot c \cdot \exp(1 - \text{fitness}(x_j)) \tag{36}$$

$$\alpha_2 = \frac{hb(k) - lb(k)}{\max_k (hb(k) - lb(k))} \cdot \exp(1 - \text{fitness}(x_j)) \cdot c \tag{37}$$

where $\text{fitness}(x_j)$ represents the quality of the attractor, c is the constant scale factor and ϵ is randomly generated from uniform distribution. In the proposed attractiveness formula, the quality of the attractor affects the randomness parameter defined by (37). In case of high luminous intensity individuals x_j less randomness value ϵ is added. If the firefly x_j is weak then the perturbation grows.

As the search progresses, unfeasible individuals can be obtained. In order to deal with unfeasibility, we use the following border reflection mechanism. If the updated value of $x_i(k)$ exceeds the upper bound $hb(k)$ then it is set according to

$$x_i(k) = U(\text{val}, hb(k)) \tag{38}$$

where

$$\text{val} = c_1 \cdot lb(k) + (1 - c_1) \cdot hb(k) \tag{39}$$

$c_1 \in (0, 1)$ and U stands for the uniform distribution $U(\text{val}, hb(k))$. If the computed value of $x_i(k)$ is below $lb(k)$ then we apply the following updating rule

$$x_i(k) = U(lb(k), val) \tag{40}$$

where $c_1 \in (0, 1)$, val is defined by (39), and U stands for the uniform distribution $U(val, hb(k))$.

In case x_j is replaced by the best individual of the current firefly population, g , we get the modified updating rules of APSO algorithm

$$x_i(k) = x_i(k) + \beta_i(k) \cdot (g(k) - x_i(k)) + \frac{hb(k) - lb(k)}{\max_k(hb(k) - lb(k))} \cdot \epsilon \cdot c \tag{41}$$

$$x_i(k) = x_i(k) + \beta_i(k) \cdot (g(k) - x_i(k)) + \frac{hb(k) - lb(k)}{\max_k(hb(k) - lb(k))} \cdot \epsilon \cdot c \cdot \exp(1 - \text{fitness}(g)) \tag{42}$$

where

$$\beta_{ij}(k) = \beta_0 e^{-\gamma r(k)_{ij}^2} \tag{43}$$

$$r_i(k) = \frac{|x_i(k) - g(k)|}{D_{\max}(k)} \tag{44}$$

5.2. Hybridization between ES Search and APSO

The main aim of the hybrid approach introduced next is to find a trade-off between the accuracy and the efficiency of the obtained algorithms. The resulted techniques combine the ability of auto-adaptive ES search to correctly identify the direction of fitness increase to find out a global optimum solution with the strong quick convergence ability of APSO.

The proposed hybrid technique is a two-stage mechanism that first uses an ES search mechanism to detect the direction of fitness increase and a certain population, Pop , then applies one of the proposed variants of APSO using a subset of good individuals $X_0 \subseteq Pop$ to quickly reach an optimal solution. The idea behind this hybridization relies on the following remarks. Despite the fact that ES mechanisms are in general very suitable to rapidly identify promising areas of the space search, they are less good for finely tuning the solutions. A more efficient method might be to incorporate a more systematic search of the vicinity of good solutions by adding a search mechanism with a quick convergence ability.

The ES component of the hybrid technique is characterized by: σ_{ini} , the initial values of σ -parameter; ϵ_σ , the minimum value of each step size; $NMax$, the maximum number of generations; and the threshold parameter τ controlling the desired sub-optimal fitness value, $\tau < 1$. The parameters of APSO are: $MaxGeneration$, the maximum number of firefly populations, β_0, γ the parameters of the updating rule, and n the APSO population size.

The computation scheme of the proposed method is described as follows

Algorithm 3 Hybrid ES-APSO algorithm

- 1: Inputs: $\mu, \lambda, \sigma_{ini}, \epsilon_\sigma, NMax, \tau, I, T, MaxGeneration, \beta_0, \gamma, n$
 - 2: $t \leftarrow 0$
 - 3: Randomly generate P_t
 - 4: Evaluate each $c \in P_t$ and compute the best fitted individual, $best$
 - 5: **while** $t < NMax$ and $\text{fitness}(best) < \tau$ **do**
 - 6: Compute O_t using a recombination procedure
 - 7: Compute MO_t , the mutated variants of individuals belonging to O_t
 - 8: Evaluate each $c \in MO_t$
 - 9: Select the next generation P_{t+1} using either $(\mu + \lambda)$ procedure or (μ, λ) mechanism and determine the chromosome best;
 - 10: $t \leftarrow t + 1$
 - 11: **end while**
-

Algorithm 3: *Cont.*

```

12: Select the best  $n$  individuals from the last computed  $P_t$  to compute  $X_0$ 
13: Initialize  $L_i$  for each  $x_i \in X_0$ ;  $t \leftarrow 0$ 
14: while ( $t < \text{MaxGeneration}$ ) do
15:   Compute  $g$ , the best fitted firefly from the current population
16:   for  $i = 1:n$ 
17:     Move firefly  $x_i$  toward firefly  $g$  using (41) or (42)
18:   end for
19:   Evaluate all candidate solutions and update the light intensity values
20:    $t \leftarrow t + 1$ 
21: end while
22: Rank all fireflies and return the best one

```

6. Experimental Results

In order to derive conclusions regarding the accuracy and the computational complexity of the proposed methods, we have conducted a long series of experiments on various images representing signatures.

The quality of each algorithm from the accuracy of resulted alignment point of view has been measured in terms of success rate, computed as follows. For each input pair of images, we denote by Tests the number of tests and let Success be the number of successful tests. We call an algorithm test a success if the fitness of the resulted best individual exceeds 0.85. Note that, for each image, each technique has been tested 500 times to obtain meaningful results. The success rate of a certain algorithm, SR, is given by

$$SR = \frac{\text{Success}}{\text{Tests}} \cdot 100\% \quad (45)$$

Moreover, one of the most commonly used quantitative similarity measures, SNR, has been computed to derive conclusions on the registration quality. The SNR value computed for a certain sensed image S versus a reference image T of the same size (M, N) is defined by

$$SNR(T, S) = 10^* \log_{10} \left[\frac{\sum_{x=1}^M \sum_{y=1}^N (S(x, y))^2}{\sum_{x=1}^M \sum_{y=1}^N (T(x, y) - S(x, y))^2} \right] \quad (46)$$

The computational complexity/efficiency of each algorithm is evaluated in terms of execution time. Our tests have been conducted on a computer with the following configuration: Processor—Intel Core I7-7700 3.60 GHZ, Memory—8GB DDR4 2400 MHZ, Storage—1 TB HDD 7200 RPM, SATA3.

In the following we present the experimentally established results in case of using each class of metaheuristics. The success rate and the mean values of mutual information ratio (MIR), SNR and run time respectively are reported below.

The mean value corresponding to each above-mentioned performance measure has been computed by averaging the values of the corresponding measure resulted for each run. Note that each algorithm has been run 500 times for each input pair of images $\mathcal{P}\mathcal{J}$. The reported results correspond to a set of 19 pair of images representing signatures, $\mathcal{S}\mathcal{P}\mathcal{J}$, perturbed by the same degradation model (8).

The techniques belonging to the first class, APSO1 and APSO2, implement APSO algorithms using the updating rules (41) and (42) respectively. In our tests the parameters were set as follows: $\beta_0 = 2$, $\gamma = 2$, $\alpha_{\text{damp}} = 0.99$ and the population size is 50.

The results of applying APSO1 and APSO2 techniques to register the images belonging to $\mathcal{S}\mathcal{J}$ are displayed in Tables 1–4. The tests pointed out that APSO2 persistently obtains better results from the accuracy point of view, but the computational effort is slightly increased as compared to APSO1.

Note that there are situations in which the accuracy of APSO2 is significantly better than the accuracy of APSO1. Nevertheless, the accuracy of APSO-based recognition procedure is still low for some input pairs of images.

The second class of algorithms includes Firefly1 and Firefly2, variants of Firefly algorithms considering the updating rules (31) and (36) respectively. In our tests the parameters were set as follows: $\beta_0 = 2$, $\gamma = 2$, $\alpha_{damp} = 0.98$ and the population size is 30. Obviously, the firefly approaches involve a significantly larger computational effort as compared to APSO algorithms.

From the accuracy point of view, SR = 100% for all pair of images in $\mathcal{S}\mathcal{J}$, while MIR values are around 0.87 and the SNR index frequently exceeds 24. The results are displayed in Tables 1–4.

Note that the computational effort is significantly reduced in case Firefly technique is implemented based on the updating rule (36).

The hybrid techniques belonging to the third class of proposed methods, ES-APSO1 and ES-APSO2, implement the algorithm described in Section 5.2 based on the updating rules (41) and (42) respectively. We conducted the tests using the following parameters setting:

- ES $\mu = 50$, $\lambda = 300$, $NMax = 200$, $\tau = 0.2, \sigma_{ini} = [1, 1, 0.01, 0.01, 0.01]$ and $\epsilon_{\sigma} = [0.0075, 0.0075, 0.004, 0.004, 0.004]$;
- APSO— $\beta_0 = 2$, $\gamma = 2$, $\alpha_{damp} = 0.99$, and the population size is 22.

The results of applying ES-APSO1 and ES-APSO2 techniques to register the images belonging to $\mathcal{S}\mathcal{J}$ are displayed in Tables 1–4.

Table 1. The SR values of the proposed algorithms.

Input	APSO1	APSO2	Firefly1	Firefly2	ES-APSO1	ES-APSO2
$\mathcal{P}\mathcal{J}_1$	98%	100%	100%	100%	99%	100%
$\mathcal{P}\mathcal{J}_2$	97%	98%	100%	100%	99.50%	99.50%
$\mathcal{P}\mathcal{J}_3$	98.75%	99.25%	100%	100%	97.50%	99%
$\mathcal{P}\mathcal{J}_4$	98.50%	99.25%	100%	100%	100%	100%
$\mathcal{P}\mathcal{J}_5$	99.25%	100%	100%	100%	99.50%	100%
$\mathcal{P}\mathcal{J}_6$	77.25%	85.75%	100%	100%	100%	100%
$\mathcal{P}\mathcal{J}_7$	93.50%	95%	100%	100%	99%	100%
$\mathcal{P}\mathcal{J}_8$	87.75%	96.50%	100%	100%	99%	99.50%
$\mathcal{P}\mathcal{J}_9$	99.50%	99.50%	100%	100%	99.50%	100%
$\mathcal{P}\mathcal{J}_{10}$	90.50%	97.25%	100%	100%	100%	99.50%
$\mathcal{P}\mathcal{J}_{11}$	95.50%	95.50%	100%	100%	99.50%	99.50%
$\mathcal{P}\mathcal{J}_{12}$	88%	91.75%	100%	100%	99%	100%
$\mathcal{P}\mathcal{J}_{13}$	99.50%	99%	100%	100%	100%	99.50%
$\mathcal{P}\mathcal{J}_{14}$	71%	73.75%	100%	100%	99%	99%
$\mathcal{P}\mathcal{J}_{15}$	81.75%	86%	100%	100%	98.50%	98.50%
$\mathcal{P}\mathcal{J}_{16}$	73.25%	82.75%	100%	100%	99.50%	99%
$\mathcal{P}\mathcal{J}_{17}$	95.75%	96.75%	100%	100%	99%	98.50%
$\mathcal{P}\mathcal{J}_{18}$	81.50%	87.25%	100%	100%	93.75%	97.25%
$\mathcal{P}\mathcal{J}_{19}$	90.25%	98.25%	100%	100%	98.50%	98.50%

Table 2. The mean values of MIR measures.

Input	APSO1	APSO2	Firefly1	Firefly2	ES-APSO1	ES-APSO2
$\mathcal{P}\mathcal{J}_1$	0.851	0.869	0.871	0.874	0.869	0.873
$\mathcal{P}\mathcal{J}_2$	0.852	0.857	0.874	0.873	0.871	0.871
$\mathcal{P}\mathcal{J}_3$	0.861	8.863	0.87	0.87	0.861	0.869
$\mathcal{P}\mathcal{J}_4$	0.86	0.87	0.874	0.871	0.87	0.873
$\mathcal{P}\mathcal{J}_5$	0.866	0.872	0.874	0.875	0.87	0.87
$\mathcal{P}\mathcal{J}_6$	0.712	0.772	0.872	0.875	0.871	0.872
$\mathcal{P}\mathcal{J}_7$	0.822	0.833	0.873	0.871	0.868	0.869
$\mathcal{P}\mathcal{J}_8$	0.789	0.847	0.868	0.874	0.869	0.868

Table 2. Cont.

Input	APSO1	APSO2	Firefly1	Firefly2	ES-APSO1	ES-APSO2
$\mathcal{P}\mathcal{J}_9$	0.868	0.865	0.871	0.869	0.871	0.872
$\mathcal{P}\mathcal{J}_{10}$	0.798	0.85	0.871	0.872	0.871	0.867
$\mathcal{P}\mathcal{J}_{11}$	0.838	0.839	0.873	0.873	0.871	0.868
$\mathcal{P}\mathcal{J}_{12}$	0.779	0.808	0.871	0.872	0.868	0.872
$\mathcal{P}\mathcal{J}_{13}$	0.866	0.862	0.869	0.87	0.869	0.866
$\mathcal{P}\mathcal{J}_{14}$	0.643	0.67	0.87	0.869	0.866	0.862
$\mathcal{P}\mathcal{J}_{15}$	0.736	0.778	0.869	0.871	0.861	0.864
$\mathcal{P}\mathcal{J}_{16}$	0.656	0.733	0.873	0.87	0.867	0.866
$\mathcal{P}\mathcal{J}_{17}$	0.84	0.848	0.868	0.868	0.866	0.86
$\mathcal{P}\mathcal{J}_{18}$	0.73	0.781	0.861	0.863	0.829	0.848
$\mathcal{P}\mathcal{J}_{19}$	0.796	0.861	0.867	0.866	0.861	0.856

Table 3. The mean values of SNR.

Input	APSO1	APSO2	Firefly1	Firefly2	ES-APSO1	ES-APSO2
$\mathcal{P}\mathcal{J}_1$	23.81	24.01	23.99	24.1	23.87	24.02
$\mathcal{P}\mathcal{J}_2$	25.99	26.03	26.3	26.3	26.2	26.19
$\mathcal{P}\mathcal{J}_3$	24.48	24.48	24.65	24.65	24.42	24.6
$\mathcal{P}\mathcal{J}_4$	22.99	23.23	24.69	24.55	24.52	24.67
$\mathcal{P}\mathcal{J}_5$	24.16	24.28	24.33	24.43	24.2	24.21
$\mathcal{P}\mathcal{J}_6$	22.9	23.86	25.55	25.68	25.51	25.56
$\mathcal{P}\mathcal{J}_7$	24.09	24.21	24.9	24.8	24.7	24.72
$\mathcal{P}\mathcal{J}_8$	24	24.93	25.23	25.48	25.28	25.3
$\mathcal{P}\mathcal{J}_9$	24.19	24.03	24.15	24.11	24.24	24.21
$\mathcal{P}\mathcal{J}_{10}$	23.37	24.2	24.56	24.58	24.54	24.44
$\mathcal{P}\mathcal{J}_{11}$	25.13	25.18	25.75	25.75	25.72	25.54
$\mathcal{P}\mathcal{J}_{12}$	19.67	20.09	21.12	21.17	21.07	21.18
$\mathcal{P}\mathcal{J}_{13}$	25.16	25.09	25.3	25.39	25.24	25.12
$\mathcal{P}\mathcal{J}_{14}$	22.07	22.48	25.82	25.76	25.71	25.59
$\mathcal{P}\mathcal{J}_{15}$	22.58	23.3	24.82	24.94	24.63	24.74
$\mathcal{P}\mathcal{J}_{16}$	23.18	24.4	26.69	26.53	26.47	26.44
$\mathcal{P}\mathcal{J}_{17}$	25.62	25.75	26.04	26.03	26.06	25.9
$\mathcal{P}\mathcal{J}_{18}$	24.55	25.38	26.72	26.8	26.16	26.53
$\mathcal{P}\mathcal{J}_{19}$	25.66	26.69	26.74	26.72	26.65	26.48

Table 4. The mean run times (s).

Input	APSO1	APSO2	Firefly1	Firefly2	ES-APSO1	ES-APSO2
$\mathcal{P}\mathcal{J}_1$	9.45	11.46	32	24.42	6.92	7.67
$\mathcal{P}\mathcal{J}_2$	8.73	10.04	29.89	21.7	6.81	7.73
$\mathcal{P}\mathcal{J}_3$	13.37	15.43	45.12	32.61	9.35	10.33
$\mathcal{P}\mathcal{J}_4$	10.92	12.98	37.99	26.33	8.2	9.1
$\mathcal{P}\mathcal{J}_5$	12.07	14.21	41.97	29.38	8.95	9.99
$\mathcal{P}\mathcal{J}_6$	10.44	10.87	29.55	21.36	6.9	7.66
$\mathcal{P}\mathcal{J}_7$	10.71	11.61	36.88	26.45	8.43	9.46
$\mathcal{P}\mathcal{J}_8$	10.34	11.47	31.49	23.17	7.33	8.45
$\mathcal{P}\mathcal{J}_9$	10.73	6.09	37.62	26.13	8.72	9.77
$\mathcal{P}\mathcal{J}_{10}$	12.34	13.63	39.53	29.86	8.99	10.32
$\mathcal{P}\mathcal{J}_{11}$	9.5	5.75	32.6	24.13	7.12	8
$\mathcal{P}\mathcal{J}_{12}$	15.74	16.77	61.61	45.48	12.96	16.05
$\mathcal{P}\mathcal{J}_{13}$	10.61	6.02	35.67	26.18	8.15	9.18
$\mathcal{P}\mathcal{J}_{14}$	18.41	18.61	43.65	32.9	10.86	11.84
$\mathcal{P}\mathcal{J}_{15}$	19.68	19.76	46.96	35.85	12.94	14.03
$\mathcal{P}\mathcal{J}_{16}$	18.28	19.58	40.98	30.79	10.52	11.31
$\mathcal{P}\mathcal{J}_{17}$	11.86	13.74	40.99	30.02	9.25	10.11
$\mathcal{P}\mathcal{J}_{18}$	14.24	15.4	44.77	33.93	13	12.86
$\mathcal{P}\mathcal{J}_{19}$	11.71	12.81	37.57	28.81	9.79	10.64

Taking into account both accuracy and efficiency points of view, the proposed hybrid methods prove excellent results.

Note that the APSO and Firefly parameters β_0 and γ are usually set in $(0, 10)$ [25,31]. In order to tune the parameters values to the considered registration problem, we have implemented a standard self-adaptive method. The initial values of β_0 and γ are set to 2. As the APSO/Firefly self-adaptive procedure evolves, the values of β_0 and γ vary between 1.8 and 2.1. We conclude that we can set β_0 and γ to 2, the result being consistent to the parameters used in reference [32] for the PSO algorithm. The parameter α was set either using (31) or (36).

In case of the Firefly algorithm, the recommended population size is usually between 15 and 100 [25]. Since the APSO algorithm is a reduced variant of the Firefly algorithm, we used $\mu = 50$ individuals for APSO and $\mu = 30$ for Firefly. In case of the hybrid approaches, the initial population generation is replaced by an ES method that computes $\mu = 50$ sized populations with best individual having the fitness value above a certain threshold. The APSO algorithms are implemented next taking into account only the best $\mu = 22$ individuals.

Also, a series of tests to evaluate the quality of the proposed classes of algorithms against the quality of the well-known One Plus One Evolutionary Optimizer (EO) have been performed [33]. From both accuracy and computation effort points of view, the results obtained by the proposed hybrid methods are significantly better as compared to the aforementioned optimizer in most of the cases. For instance, in case of the pair of images depicted in Figure 1a,b the register image using ES-APSO2 is presented in Figure 2a (MIR 0.8716, SNR 24.4441) while the best result obtained by One Plus One Evolutionary Optimizer is shown in Figure 2b (MIR 0.6772, SNR 19.8342).

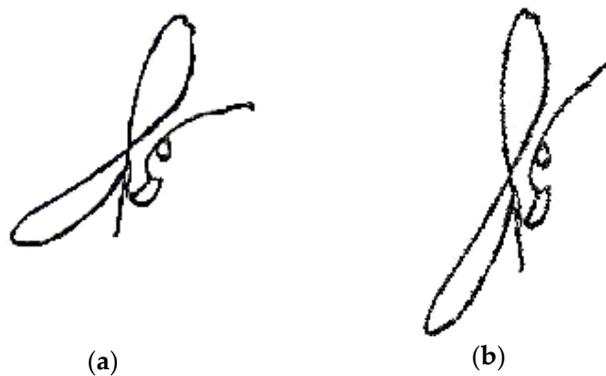


Figure 1. (a) The target image; (b) The observed image.

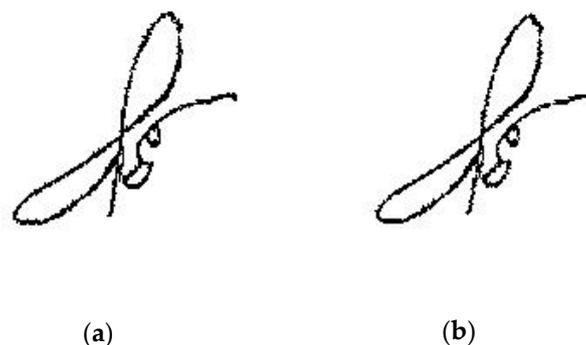


Figure 2. (a) The image obtained by ES-APSO2; (b) The image obtained by EO.

7. Conclusive Remarks and Suggestions for Further Work

The work reported in the paper addresses the problem of binary image registration for signature recognition purposes. We proposed a series of EC techniques in order to align a sensed image to a target one using the mutual information measure to compute the similarities between two images.

We evaluated the quality of the results from both accuracy and computational complexity points of view. The accuracy was established in terms of the percentage of successful runs, the quality of the solutions being also calculated in terms of the signal-to-noise ratio. The computational complexity was defined based on execution time (run time).

The first class of methods consists of APSO approaches, where the updating rules are defined based on the attributes range and the quality of current candidate solution. Based on the obtained results, we conclude that the algorithms are very fast and, in most of the cases, the accuracy is fairly good.

The class of firefly-based techniques comprises very accurate algorithms, the success percentage being 100% for every tested pair of images. The main drawback of the proposed Firefly1 and Firefly2 methods is their execution time. However, the efficiency of firefly-based approach can be significantly improved by defining updating rules that consider, besides other criteria, the attractors quality.

Finally, to find a trade-off between the accuracy and the efficiency of the algorithms, we developed a hybrid method that combines the ability of auto-adaptive ES search to discover a global optimum solution with the strong quick convergence ability of APSO. The obtained algorithms proved very fast and are also highly accurate. Also, a series of experiments led to the conclusion that the proposed hybrid technique outperforms the well-known One Plus One Evolutionary Optimizer (EO) from the accuracy point of view.

We conclude that the results are encouraging and entail future work toward extending the proposed approach to more complex perturbation models as well as more complex hybrid and memetic techniques.

Author Contributions: C.-L.C.: methodology, software, validation, analysis, writing—original draft preparation, writing—review and editing, supervision; A.S.: software, validation, analysis, writing—review and editing.

Funding: This research received no funding.

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

References

1. Goshtasby, A.A. *Image Registration: Principles, Tools and Methods*; Springer Science & Business Media: London, UK, 2012.
2. Modersitzki, J. *Numerical Methods for Image Registration*; Oxford University Press: New York, NY, USA, 2004.
3. Sheng, Q.; Wang, Q.; Zhang, X.; Wang, B.; Zhang, B.; Zhang, Z. Registration of Urban Aerial Image and LiDAR Based on Line Vectors. *Appl. Sci.* **2017**, *7*, 965. [[CrossRef](#)]
4. Pluim, J.P.W.; Maintz, J.B.A.; Viergever, M.A. Mutual information matching in multiresolution contexts. *Image Vis. Comput.* **2001**, *19*, 45–52. [[CrossRef](#)]
5. Zhou, D.; Sun, J.; Lai, C.; Xu, W.; Lee, X. An improved quantum-behaved particle swarm optimization and its application to medical image registration. *Int. J. Comput. Math.* **2011**, *88*, 1208–1223. [[CrossRef](#)]
6. Santamaría, J.; Damas, S.; García-Torres, J.; Cordon, O. Self-adaptive evolutionary image registration using differential evolution and artificial immune systems. *Pattern Recogn. Lett.* **2012**, *33*, 2065–2070. [[CrossRef](#)]
7. Torabi, A.; Massé, G.; Bilodeau, G. An iterative integrated framework for thermal–visible image registration, sensor fusion, and people tracking for video surveillance applications. *Comput. Vis. Image Understand.* **2012**, *116*, 210–221. [[CrossRef](#)]
8. State, L.; Cocianu, C.; Vlamos, P.; Constantin, D. Neural Approaches to Image Compression/Decompression Using PCA based Learning Algorithms. In Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems, PRIS, Barcelona, Spain, 12–13 June 2008; pp. 187–192.
9. Abdul Khalid, N.; Md Ariff, N.; Yahya, S.; Mohamed Noor, N. A Review of Bio-inspired Algorithms as Image Processing Techniques. In Proceedings of the Conference Software Engineering and Computer Systems, ICSECS 2011, Kuantan, Pahang, Malaysia, 27–29 June 2011; pp. 660–673.
10. Valsecchi, A.; Damas, S.; Santamaria, J. An image registration approach using genetic algorithms. In Proceedings of the Conference 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia, 10–15 June 2012; pp. 416–423.

11. Damas, S.; Cordón, O.; Santamaría, J. Medical Image Registration Using Evolutionary Computation: An Experimental Survey. *IEEE Comput. Intell. Mag.* **2011**, *6*, 26–42. [[CrossRef](#)]
12. Cocianu, C.; Stan, A. Towards an Evolution Strategy Approach in Binary Image Registration for Solving Digital Signature Recognition Tasks. In Proceedings of the 20th International Conference on Enterprise Information Systems, Funchal, Portugal, 21–24 March 2018; pp. 469–476.
13. Cocianu, C.; Stan, A. New Attempts in Binary Image Registration. In Proceedings of the 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT), Thessaloniki, Greece, 10–13 April 2018; pp. 253–258.
14. Valsecchi, A.; Dubois-Lacoste, J.; Stutzle, T.; Damas, S.; Santamaria, J.; Marrakchi-Kacem, L. Evolutionary medical image registration using automatic parameter tuning. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 1326–1333.
15. Brown, L. A Survey of Image Registration Techniques. *ACM Comput. Surv.* **1992**, *24*, 325–376. [[CrossRef](#)]
16. Li, Q.; Sato, I. Multimodality Image Registration by Particle Swarm Optimization of Mutual Information. In Proceedings of the Third International Conference on Intelligent Computing, ICIC 2007: Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Qingdao, China, 21–24 August 2007; pp. 1120–1130.
17. Zhuang, Y.; Gao, K.; Miu, X.; Han, L.; Gong, X. Infrared and visual image registration based on mutual information with a combined particle swarm optimization—Powell search algorithm. *Optik Int. J. Light Electron Opt.* **2016**, *127*, 188–191. [[CrossRef](#)]
18. Pluim, J.; Maintz, J.; Viergever, M. Mutual-information based registration of medical images: A survey. *IEEE Trans. Med. Imaging* **2003**, *22*, 986–1004. [[CrossRef](#)] [[PubMed](#)]
19. Cover, T.; Thomas, J. *Elements of Information Theory*; Wiley-Interscience John Wiley & Sons: Chichester, UK, 2006.
20. Yang, X. *Nature-Inspired Metaheuristic Algorithms*; Luniver: Frome, UK, 2010.
21. Osaba, E.; Yang, X.; Diaz, F.; Onieva, E.; Masegosa, A.; Perillos, A. A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. *Soft Comput.* **2016**, *21*, 5295–5308. [[CrossRef](#)]
22. Crawford, B.; Soto, R.; Johnson, F.; Valencia, C.; Paredes, F. Firefly Algorithm to Solve a Project Scheduling Problem. *Adv. Intell. Syst. Comput.* **2016**, *1*, 449–458.
23. Sharma, A.; Sehgal, S. Image segmentation using firefly algorithm. In Proceedings of the 2016 International Conference on Information Technology (InCITe)—The Next Generation IT Summit on the Theme—Internet of Things: Connect your Worlds, Noida, India, 6–7 October 2016.
24. Bharathi Raja, S.; Srinivas Pramod, C.; Vamshee Krishna, K.; Rangunathan, A.; Vinesh, S. Optimization of electrical discharge machining parameters on hardened die steel using Firefly Algorithm. *Eng. Comput.* **2013**, *31*, 1–9. [[CrossRef](#)]
25. Yang, X. *Cuckoo Search and Firefly Algorithm*; Springer: London, UK, 2014.
26. Yang, X.; Deb, S.; Fong, S. Accelerated Particle Swarm Optimization and Support Vector Machine for Business Optimization and Applications. In Proceedings of the International Conference on Networked Digital Technologies, Macau, China, 11–13 July 2011; pp. 53–66.
27. Yang, X. *Nature-Inspired Algorithms and Applied Optimization*; Springer: Cham, Switzerland, 2018.
28. Eiben, A.; Smith, J. *Introduction to Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2003.
29. Edelkamp, S.; Schrödl, S. *Heuristic Search*; Morgan Kaufmann: Amsterdam, The Netherlands, 2012.
30. Kramer, O. *Self-Adaptive Heuristics for Evolutionary Computation*; Springer: Berlin, Germany, 2008.
31. Chintam, J.; Daniel, M. Real-Power Rescheduling of Generators for Congestion Management Using a Novel Satin Bowerbird Optimization Algorithm. *Energies* **2018**, *11*, 183. [[CrossRef](#)]
32. Yang, X. Firefly Algorithms for Multimodal Optimization. *Stoch. Algorithms Found. Appl.* **2009**, *5792*, 169–178.
33. Styner, M.; Brechbuehler, C.; Székely, G.; Gerig, G. Parametric estimate of intensity inhomogeneities applied to MRI. *IEEE Trans. Med. Imaging* **2000**, *19*, 153–165. [[CrossRef](#)] [[PubMed](#)]

