

Research Article

A Novel Artificial Bee Colony Algorithm Based on Internal-Feedback Strategy for Image Template Matching

Bai Li,¹ Li-Gang Gong,² and Ya Li³

¹ School of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China

² School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

³ School of Mathematics and Systems Science & LMIB, Beihang University, Beijing 100191, China

Correspondence should be addressed to Ya Li; yli@buaa.edu.cn

Received 24 December 2013; Accepted 26 March 2014; Published 29 April 2014

Academic Editors: P. Melin, D. Simson, C.-W. Tsai, and F. Yu

Copyright © 2014 Bai Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Image template matching refers to the technique of locating a given reference image over a source image such that they are the most similar. It is a fundamental mission in the field of visual target recognition. In general, there are two critical aspects of a template matching scheme. One is similarity measurement and the other is best-match location search. In this work, we choose the well-known normalized cross correlation model as a similarity criterion. The searching procedure for the best-match location is carried out through an internal-feedback artificial bee colony (IF-ABC) algorithm. IF-ABC algorithm is highlighted by its effort to fight against premature convergence. This purpose is achieved through discarding the conventional roulette selection procedure in the ABC algorithm so as to provide each employed bee an equal chance to be followed by the onlooker bees in the local search phase. Besides that, we also suggest efficiently utilizing the internal convergence states as feedback guidance for searching intensity in the subsequent cycles of iteration. We have investigated four ideal template matching cases as well as four actual cases using different searching algorithms. Our simulation results show that the IF-ABC algorithm is more effective and robust for this template matching mission than the conventional ABC and two state-of-the-art modified ABC algorithms do.

1. Introduction

Template matching is defined as the action of recognizing predefined template patterns in a source image. It is a fundamental issue in pattern recognition and has been widely applied to the fields such as face recognition [1], pulmonary nodules detection [2], handwriting identification [3], and road detection [4] over the past few decades.

In general, template matching involves two critical aspects: similarity measurement and best-match search [5, 6]. When measuring the similarity, a source image and a predefined template image are superimposed in a certain location and then the similarity evaluation is made on the basis of a selected model or criterion. The sum of absolute differences (SAD), the sum of squared differences (SSD), and the normalized cross correlation (NCC) are all popular similarity measurement models. SAD is sensitive to the illumination

disturbances and may lead to large variations in the intensity values [7]. SSD has a similar drawback. The NCC model utilizes rotation and scale invariant evaluations for the degree of similarity [8] and is confirmed to be more robust than SAD and SSD, especially in terms of uniform illumination changes in the source image [6, 9]. Therefore, the NCC model is more widely used than the other two models. With regard to the search strategy for a best-match position, a thorough search algorithm is proposed as a pioneering work [10], in which all the pixel-candidate positions are checked until the one with the maximum similarity is located. However, such exhaustive search is computationally expensive, which restricts its applications, especially in terms of some real-time recognition issues.

In order to reduce the computation complexity in the template matching schemes, search strategies based on evolutionary algorithms have been developed and investigated.

TABLE 1: Parameter settings for four ABCs.

ABC	I-ABC	Gbest-ABC	IF-ABC
SN = 20	SN = 20	SN = 20	SN = 20
Limit = 10	Limit = 10	Limit = 10	Limit = 10
	ap = fitness(1) _{iter=1}	C = 2	$\alpha = 0.1$

Species-based genetic algorithm (Sb-GA) [11], bat algorithm (BA) [12], chaotic quantum-behaved particle swarm optimization (CQ-PSO) [13], chaotic imperialist competitive algorithm (C-ICA) [14], and a states-of-matter search (SMS) algorithm [5] have been proposed for this template matching problem. Although these algorithms aim to reduce the computation load of the global optimums searching, they cannot avoid the derivation of suboptimal matching results. It is worth pointing out that most evolutionary algorithms are generally suitable for the optimization of convex or nearly convex functions [15]. But the template matching field contains a finite number of candidate matching locations. Such a discrete function is usually anything but smooth. In other words, the objective function concerning template matching may drastically oscillate along the domain, which critically restricts the advantages of such evolutionary algorithms. Therefore, it calls for a new way to modify these existing evolutionary algorithms so as to accommodate the discreteness and oscillation in an objective function.

Artificial bee colony (ABC) is a relatively new swarm intelligence algorithm. It is motivated by the foraging behavior of bee swarms, in which both local exploitation and global exploration are implemented [16]. Applications and developments of this algorithm have been proposed in a variety of ways [17–38]. In this paper, our previously proposed internal-feedback artificial bee colony (IF-ABC) algorithm is adopted as a search approach to find the best-match location for the template matching scheme. IF-ABC features by its effort to fight against premature convergence. In a sense, IF-ABC sacrifices part of its convergence speed for the ability to avoid premature convergence [22, 39, 40]. Our work intends to have an intensive evaluation about the true performance of IF-ABC regarding image template matching, in comparison with some other state-of-the-art ABCs.

The remainder of this paper is organized as follows. In Sections 2 and 3, basic principles of NCC model and the conventional ABC algorithm are introduced. Section 4 holds a description of IF-ABC algorithm in detail. Then, several cases of comparative experiments have been conducted in Section 5. Further discussions concerning the comparable convergence performances are included in Section 6. The final section includes the conclusions, the limitations of this study, and our future work.

2. Model of Image Template Matching

Image template matching aims to locate a given reference image (which is also called template image) over a source image such that they best match each other. Typically it is

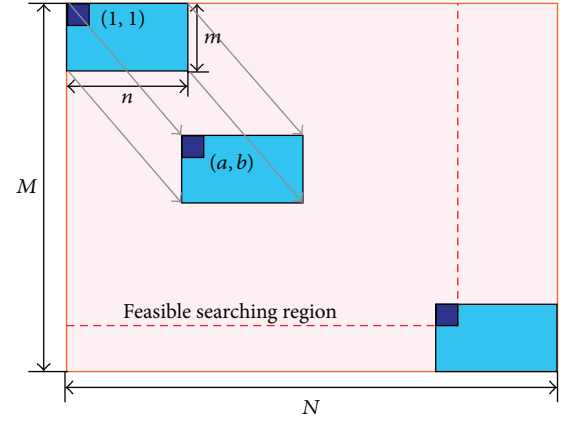


FIGURE 1: Schematic diagram of template matching process.

assumed that the predefined template image and the source image are given in RGB format. Before the matching process starts, the corresponding grayscale images are derived in the preprocessing procedure. The grayscale template image is described by a matrix $\text{temp}_{m \times n}$, in which $\text{temp}(a, b)$ refers to the gray level of the pixel located at (a, b) in the grayscale template image. It is obvious that $\text{temp}(a, b) \in [0, 255] \cap \mathbb{Z}$. Similarly, the grayscale source image is described by a matrix $\text{test}_{M \times N}$. The template matching scheme can be expressed as finding an optimal location (a, b) for the grayscale template image $\text{temp}_{m \times n}$ so that the similarity between $\text{temp}(1 : m, 1 : n)$ and $\text{test}(a : (a + m - 1), b : (b + n - 1))$ is maximized within the feasible searching region (see Figure 1).

There are a number of ways to measure such similarity. Considering the robustness in terms of disturbing illuminations, we choose the following NCC model (see (1)) in this work. Consider

$$\begin{aligned} \text{NCC}(a, b) & \triangleq \left(\sum_{x=1}^m \sum_{y=1}^n [\text{temp}(a+x-1, b+y-1) \cdot \text{test}(x, y)] \right) \\ & \times \left(\sqrt{\sum_{x=1}^m \sum_{y=1}^n [\text{temp}^2(a+x-1, b+y-1)]} \right. \\ & \quad \left. \cdot \sqrt{\sum_{x=1}^m \sum_{y=1}^n [\text{test}^2(x, y)]} \right)^{-1}, \end{aligned} \quad (1)$$

where (a, b) represents the target location of top-left-corner pixel in the grayscale template. In other words, the grayscale template image shifts according to a vector of $\beta = (a, b)$ over the grayscale source image. In this particular case, it is required that $a \in [1, M - m + 1] \cap \mathbb{Z}, b \in [1, N - n + 1] \cap \mathbb{Z}$, which are regarded as the feasible searching conditions as demonstrated in Figure 1. It is easy to see that $\text{NCC}(\cdot) \in [0, 1]$ and the optimal (a^*, b^*) satisfies $\text{NCC}(a^*, b^*) = 1$.

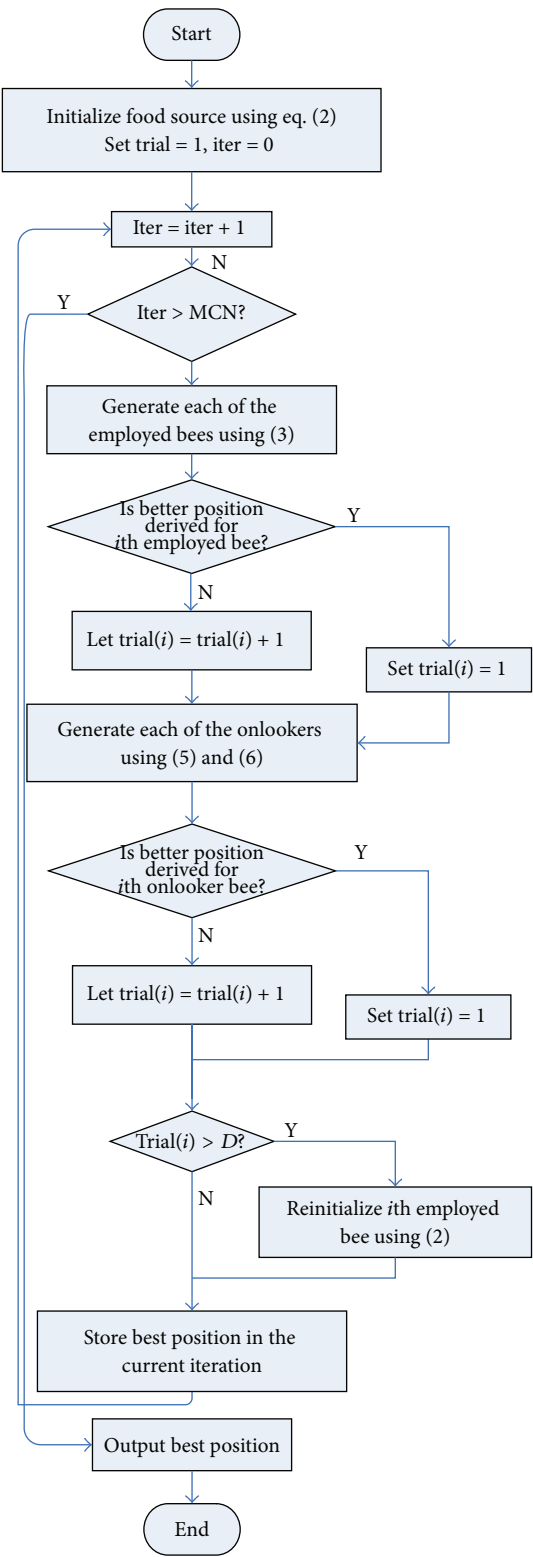


FIGURE 2: Flow chart of IF-ABC algorithm.

TABLE 2: Comparative convergence performances of four ABCs.

Case number	MCN	ABC		I-ABC		Gbest-ABC		IF-ABC	
		Mean	C.R.	Mean	C.R.	Mean	C.R.	Mean	C.R.
1	100	0.9874	39.60%	0.9647	1.20%	0.9870	58.80%	0.9927	79.80%
2	1000	0.9701	25.40%	0.9400	0.40%	0.9709	36.40%	0.9829	57.80%
3	1000	0.9517	22.00%	0.9103	1.80%	0.9637	41.40%	0.9698	50.60%
4	1000	0.8806	3.00%	0.8664	0.20%	0.8746	1.80%	0.9445	44.20%

Those bold values denote the best value (mean or C.R.) in every single line.

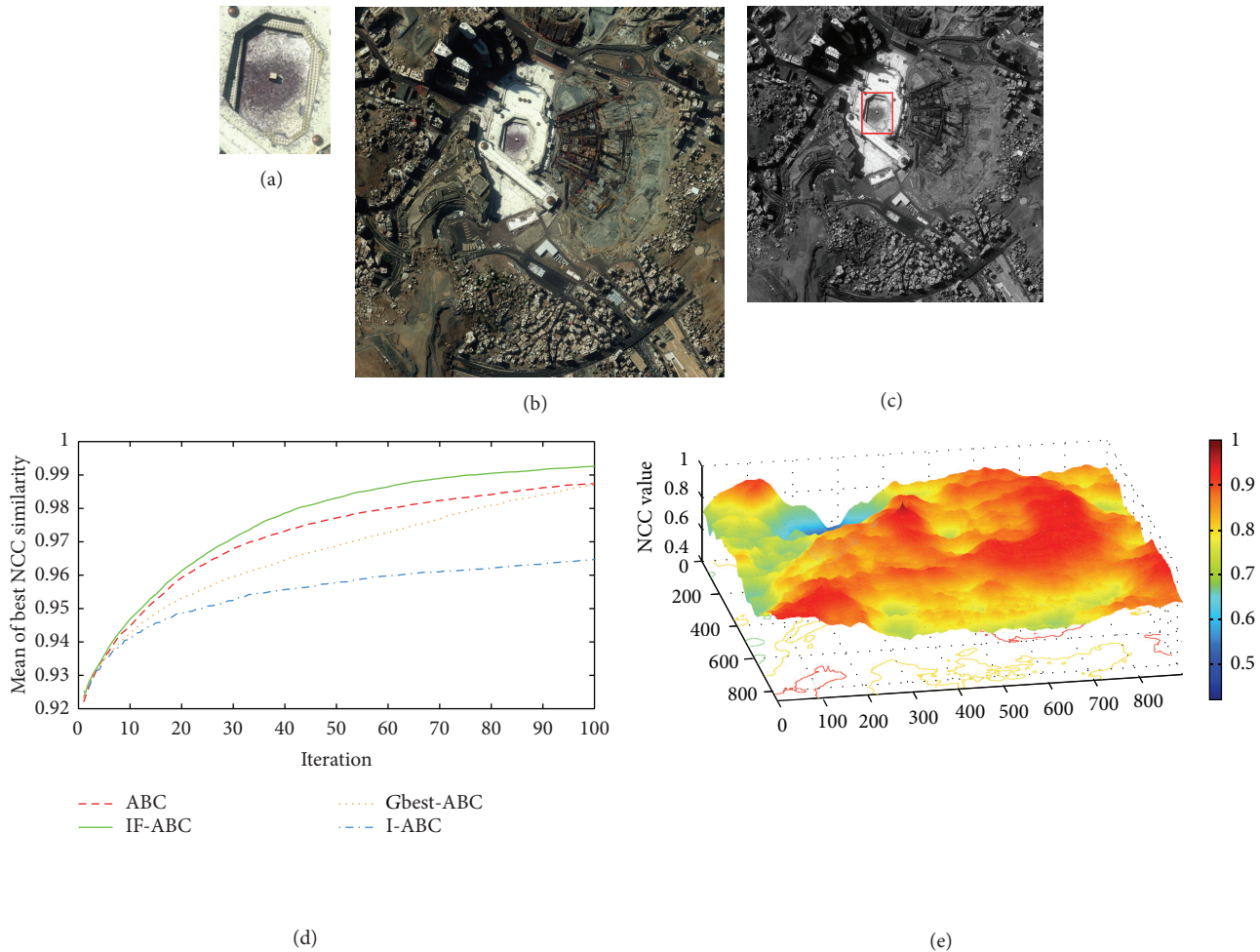


FIGURE 3: Ideal template matching case 1: satellite imagery of Holy City in Mecca. (a) Predefined template image (107×142); (b) source image (1000×1002); (c) ground matching truth; (d) comparative evolution curves of four ABC algorithms; (e) surface illustration of searching space.

Implementation procedures of template matching are given as follows.

Step 1. Import original source image and predefined template image (both in RGB form).

Step 2. Convert template and source image to grayscale.

Step 3. Choose NCC model as similarity criteria.

Step 4. Search among all feasible locations aiming to find the best-match location using IF-ABC algorithm.

Step 5. Terminate searching procedure when terminal condition is reached; then output best-match result.

3. Brief Review of Conventional ABC Algorithm

The preceding section introduces the NCC model, which is taken as a similarity criterion in this work. But how can one find the very (a^*, b^*) that maximizes NCC? In this section and the next, two intelligent algorithms named ABC and IF-ABC will be introduced, respectively, either of which can

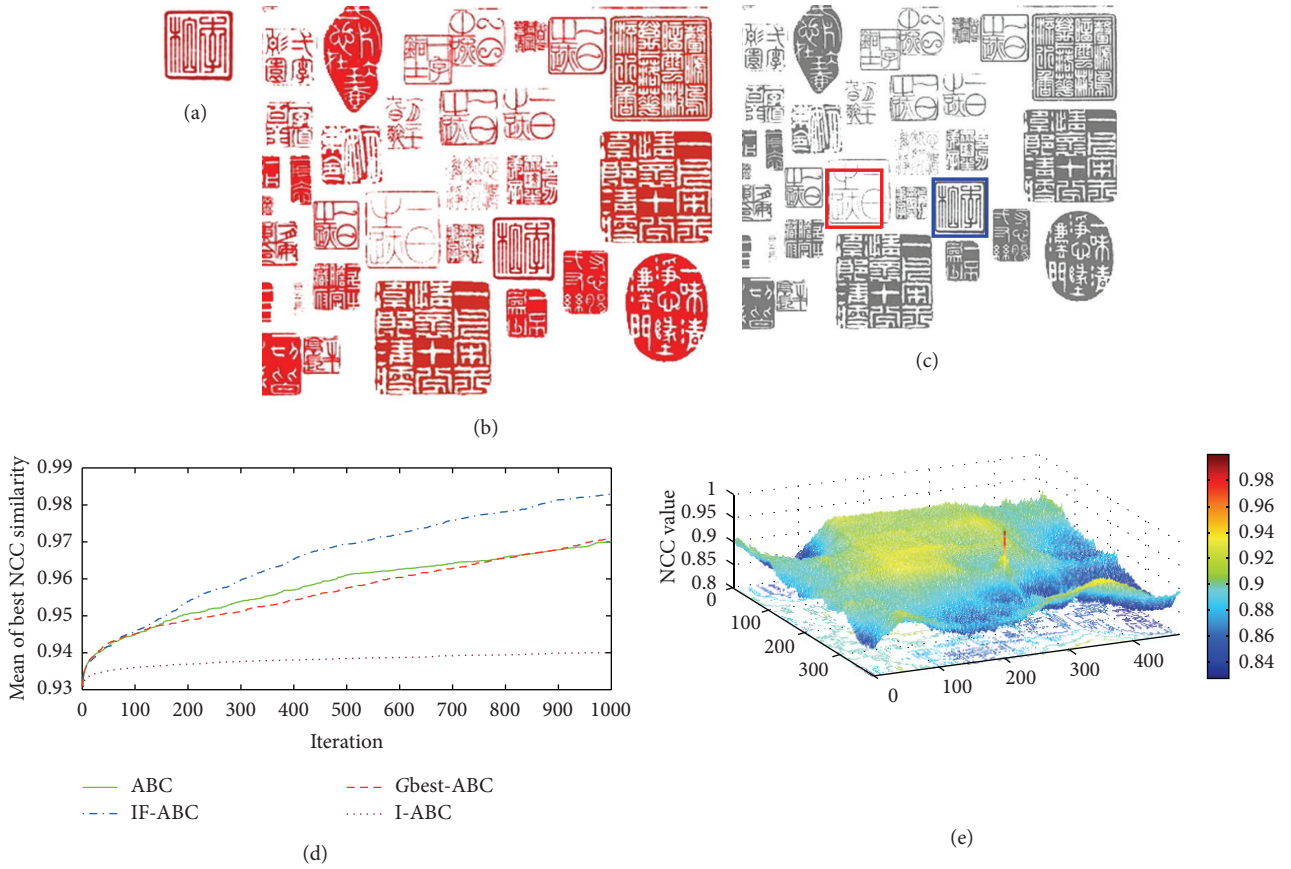


FIGURE 4: Ideal template matching case 2: traditional Chinese seal patterns. (a) Predefined template image (84×88); (b) source image (552×479); (c) ground matching truth (blue box) and most common false matching result (red box); (d) comparative evolution curves of four ABC algorithms; (e) surface illustration of searching space.

TABLE 3: Comparative convergence performances of four ABC relevant algorithms.

Case number	MCN	ABC Mean	I-ABC Mean	Gbest-ABC Mean	IF-ABC Mean
1	100	0.9564	0.9521	0.9557	0.9588
2	1000	0.9588	0.9588	0.9596	0.9598
3	1000	0.9197	0.9055	0.9241	0.9221
4	1000	0.8746	0.8613	0.8718	0.9310

Those bold values denote the best mean value in every single line.

be taken as a searching approach for the optimal matching location.

ABC is a swarm intelligence-based optimization algorithm. It is inspired by the forging behavior of bees. In this algorithm, there are three kinds of bees, namely, the employed bees, the onlooker bees, and the scout bees. They cooperate to search for the optimal nectar source in the space [16, 23].

At the beginning, an initial population is randomly generated, which contains as many as SN food sources (i.e., SN feasible solutions), using (2). Consider

$$\mathbf{X}_i \leftarrow \mathbf{X}_{\min} + \text{rand}(0, 1) \cdot (\mathbf{X}_{\max} - \mathbf{X}_{\min}), \quad (2)$$

$$i = 1, 2, \dots, \text{SN}.$$

In the above equation, each solution $\mathbf{X}_i = (x_i^1, x_i^2, \dots, x_i^D)$ is a D -dimensional vector, \mathbf{X}_{\max} and \mathbf{X}_{\min} are the predefined constraints set for the optimization problem, and $\text{rand}(0, 1)$ denotes a random number in the range $(0, 1)$ obeying the uniform distribution.

Then, the iteration process starts. Generally, as many as SN employed bees search globally in each cycle of iteration, and then SN onlooker bees search locally around the “qualified” employed bees. Those employed bees who cannot make any progress within some certain cycles will be replaced by the scout bees. The qualification standard concerns the roulette selection strategy and will be introduced later.

In detail, each employed bee utilizes the position of its randomly chosen companion to generate a new searching

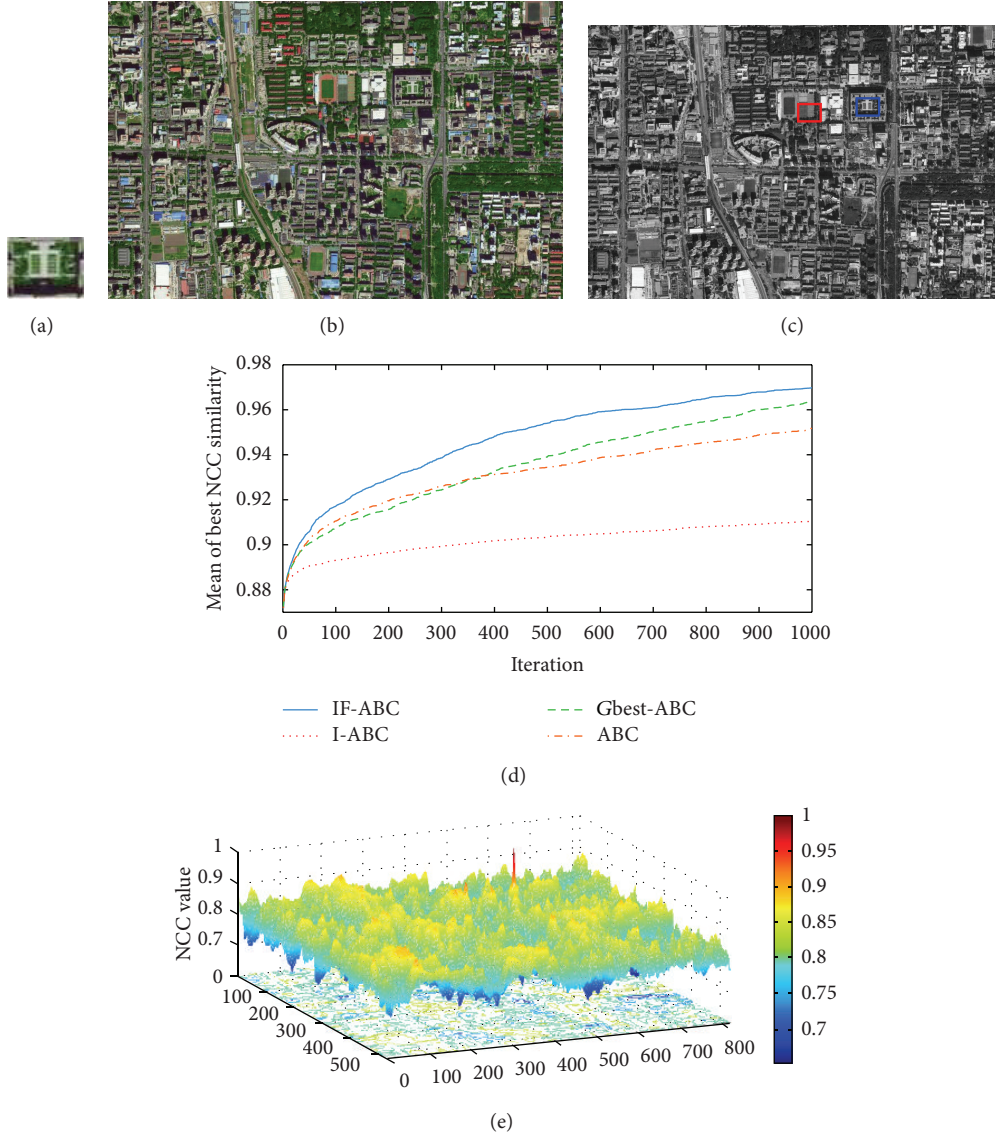


FIGURE 5: Ideal template matching case 3: satellite imagery of Beihang University campus. (a) Predefined template image (51×41); (b) source image (870×570); (c) ground matching truth (blue box) and most common false matching result (red box); (d) comparative evolution curves of four ABC algorithms; (e) surface illustration of searching space.

direction, as shown in (3). Consider

$$x_i^j \leftarrow x_i^j + \text{rand}(-1, 1) \cdot (x_k^j - x_i^j), j \neq k, \quad (3)$$

$$j, k \in [1, D] \cap \mathbb{Z}, \quad i \in [1, \text{SN}] \cap \mathbb{Z}.$$

Here, $\mathbf{X}_i = (x_i^1, x_i^2, \dots, x_i^D)$ denotes the position of the i th employed bee, $\mathbf{X}_k = (x_k^1, x_k^2, \dots, x_k^D)$ stands for the position of a randomly chosen companion, and the searching location changes in the j th element of \mathbf{X}_i .

Thereafter, the greedy selection procedure is implemented. If the new position updated by (3) is better (i.e., the corresponding objective function value is higher), the previous position is discarded; otherwise, the employed bee remains at the previous position. When all the SN employed bees complete the searching procedure mentioned above, an

index P is calculated as the qualification measurement for the employed bees using (4). Consider

$$P(i) = \frac{\text{fitness}(i)}{\sum_{j=1}^{\text{SN}} \text{fitness}(j)}, \quad (4)$$

$$\text{fitness}(i) = \begin{cases} \frac{1}{1 + \text{obj}(\mathbf{X}_i)} & \text{if } \text{obj}(\mathbf{X}_i) \geq 0 \\ 1 + \text{abs}(\text{obj}(\mathbf{X}_i)) & \text{if } \text{obj}(\mathbf{X}_i) < 0 \end{cases},$$

where $\text{obj}(\cdot)$ denotes the objective function, and $\text{fitness}(\cdot)$ is conventionally defined. Each onlooker bee needs to search around an employed bee using (3). In this case, x_k^j stands for the corresponding element of the selected employed bee, and x_i^j denotes that of the i th onlooker bee. Again, the greedy selection procedure is implemented here.

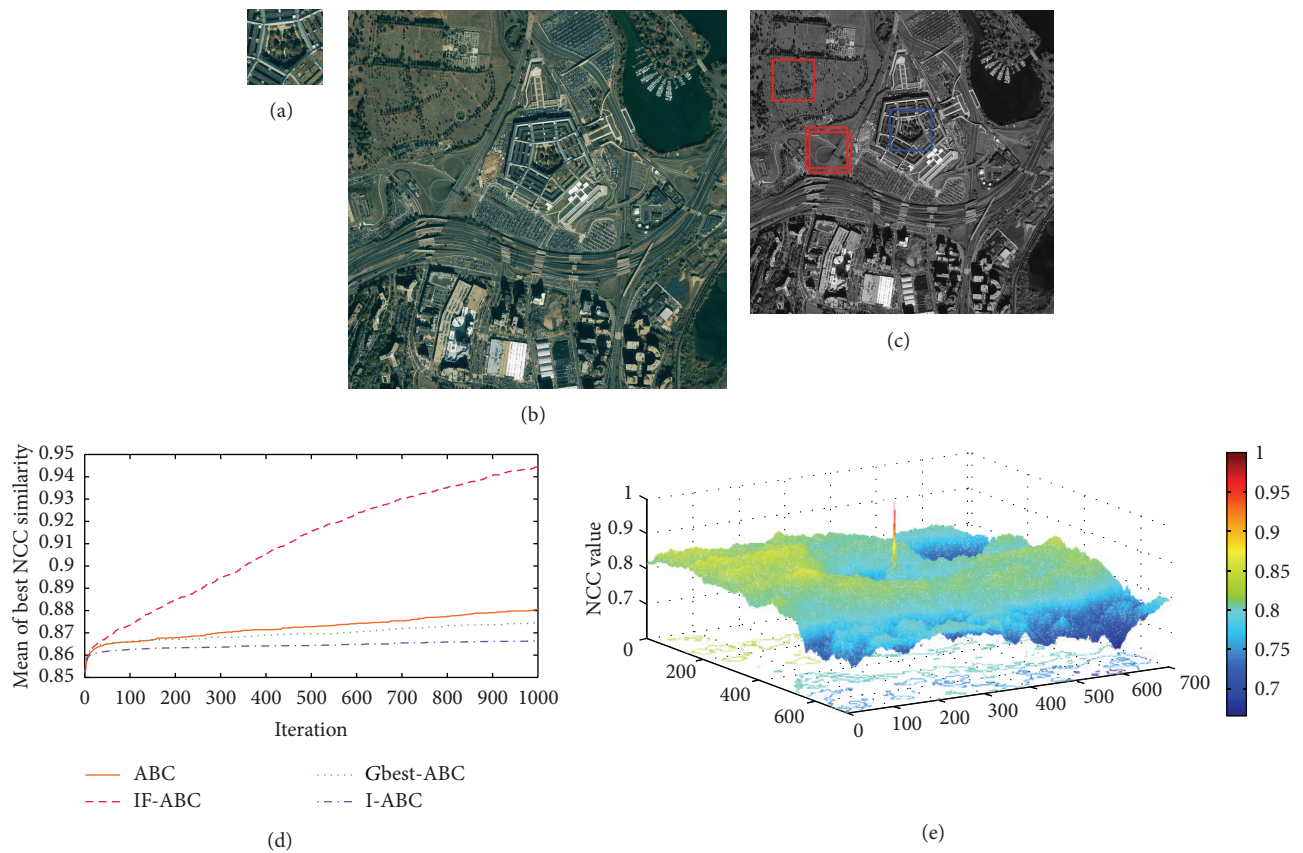


FIGURE 6: Ideal template matching case 4: satellite imagery of the Pentagon. (a) Predefined template image (116×114); (b) source image (820×820); (c) ground matching truth (blue box) and most common false matching results (red boxes); (d) comparative evolution curves of four ABC algorithms; (e) surface illustration of searching space.

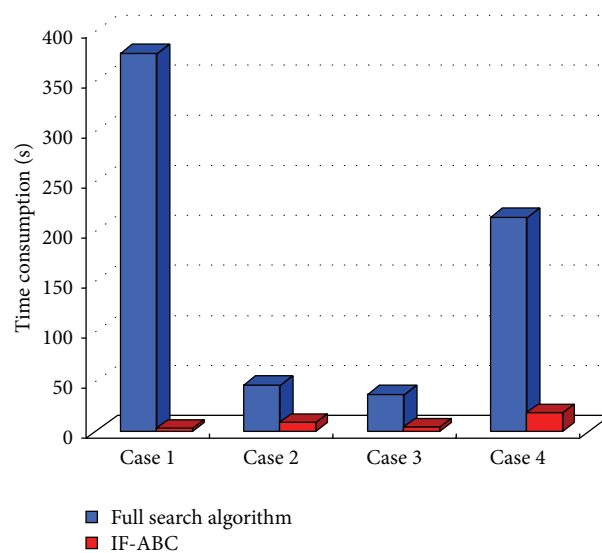


FIGURE 7: Comparisons of time consumptions between full search algorithm and IF-ABC in four ideal cases.

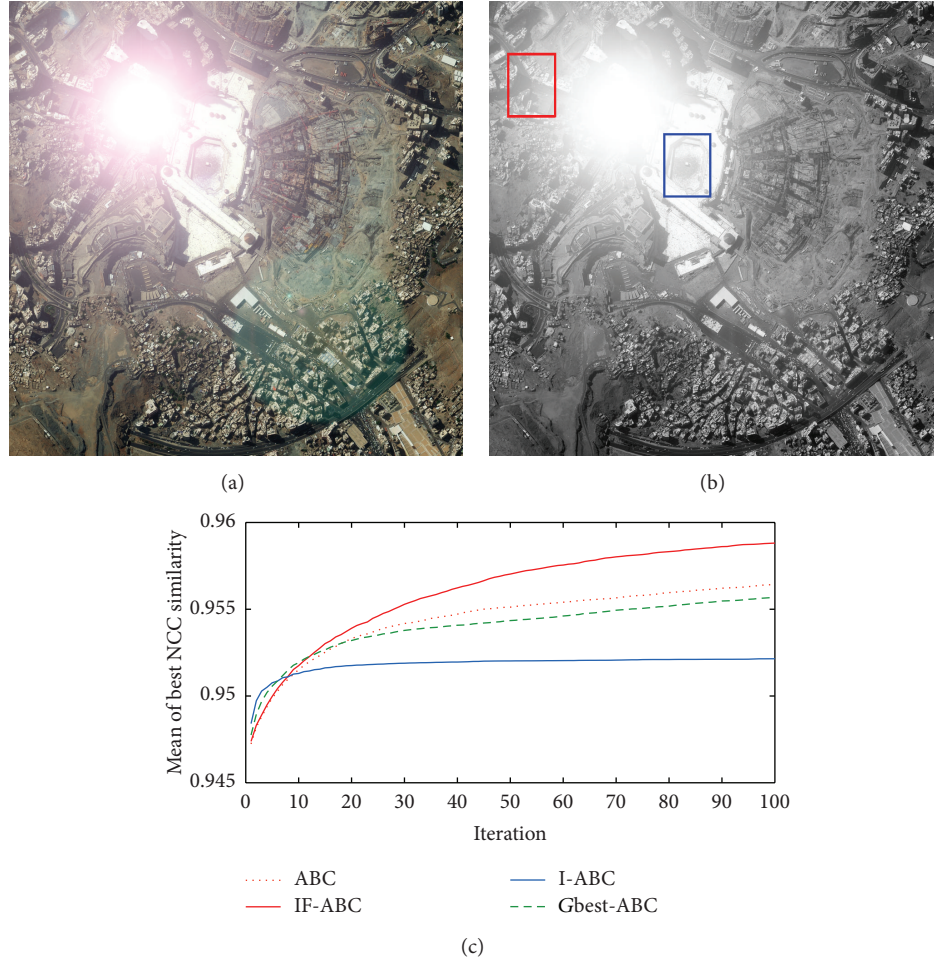


FIGURE 8: Practical template matching case 1. (a) Nonuniform illumination disturbed source image; (b) ground matching truth (blue box) and most common false matching result (red box); (c) comparative evolution curves of four ABC algorithms.

The selection principle for the qualified employed bees concerns the roulette selection strategy. If $P_1 \geq \text{rand}(0, 1)$, the first employed bee is chosen for the specific onlooker bee; otherwise, comparison between P_2 and $\text{rand}(0, 1)$ is carried on. If all the P_i are smaller than $\text{rand}(0, 1)$, such process goes over again until one employed bee satisfies the condition. In this way, each of the SN onlooker bees determines which employed bee to follow, respectively.

During each cycle of the iteration, once the i th employed bee or an onlooker bee (which searches around the i th employed bee) finds a better position in the crossover procedure, the parameter $\text{trial}(i)$ is directly reset to zero; otherwise, it is added by one. In this sense, trial is regarded as a counter recording the invalid searching times around the i th employed bee. Before a new cycle of iteration starts, it is necessary to check whether any $\text{trial}(i)$ exceeds a certain threshold Limit. If $\text{trial}(i) > \text{Limit}$, the i th employed bee will be directly replaced by a scout bee. A scout bee simply stands for a randomly initialized position utilizing (2).

4. Principle of IF-ABC Algorithm

IF-ABC algorithm was originally proposed to optimize protein secondary structures in our previous works [22, 40]. In this paper, it is slightly modified to accommodate well to the template matching problem.

At first, as many as SN employed bees are randomly sent out to explore in the feasible solution space using (2). Then, the iteration process gets started. In each cycle of iteration, an employed bee utilizes the location of one randomly selected companion to generate a new searching location using (3). Afterwards, the onlooker bees carry on the searching process.

In the IF-ABC algorithm, each of the employed bees is given a chance to be followed by an onlooker no matter whether they are “qualified” or not. In other words, we discard the conventional roulette selection procedure mentioned in ABC here, pursuing to bring more chances (i.e., more dynamics and indeterminacy) to the evolution process. Therefore, the i th onlooker bee in IF-ABC directly chooses the i th employed bees to follow. At this point, a new idea is proposed (see (5)), which presents a new principle for onlooker bees

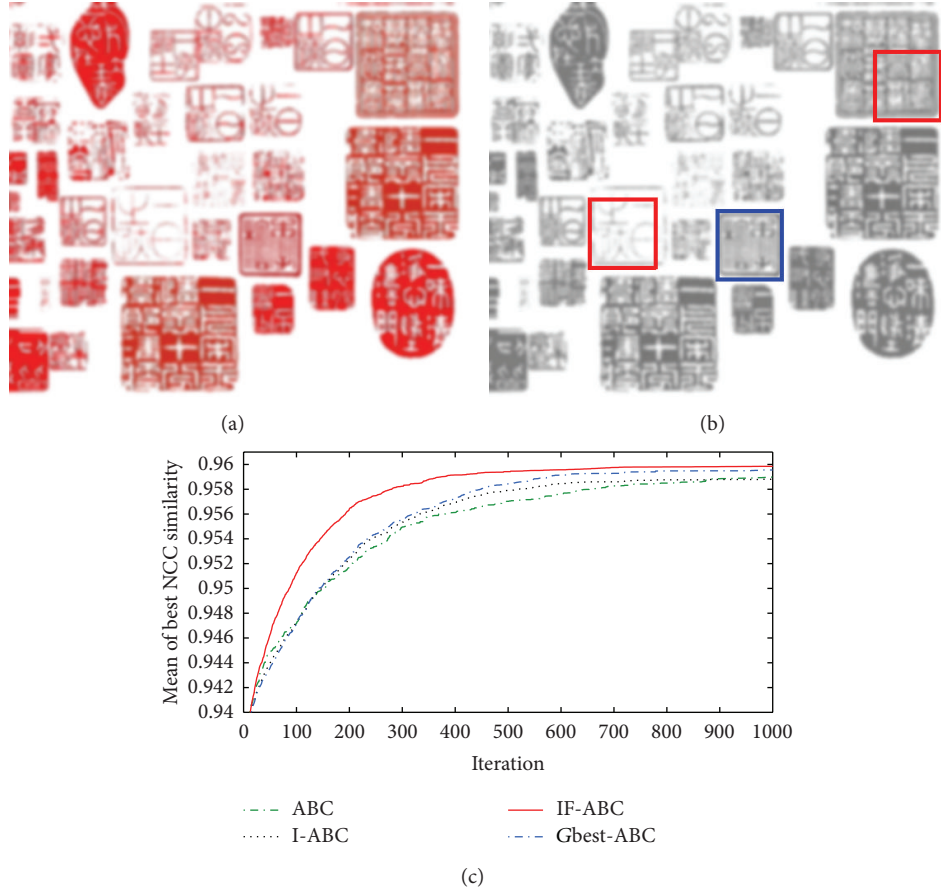


FIGURE 9: Practical template matching case 2. (a) Gaussian blurred source image; (b) ground matching truth (blue box) and most common false matching results (red boxes); (c) comparative evolution curves of four ABC algorithms.

to follow the employed bees. It is notable that we introduce a multiplier $\gamma(i)$ in this equation, which is defined in (6). Consider

$$x_i^j \leftarrow x_i^j + \gamma(i) \cdot \text{rand}(-1, 1) \cdot (x_k^j - x_i^j), \quad (5)$$

$$j \neq k, \quad j, k \in [1, D] \cap \mathbb{Z}, \quad i \in [1, \text{SN}] \cap \mathbb{Z},$$

$$\gamma(i) = \exp\left((\text{trial}(i) - 1) \cdot \frac{\ln \alpha}{\text{Limit} - 1}\right). \quad (6)$$

Similar to ABC, the parameter trial in (6) records the number of inefficient searching times. In IF-ABC, if the i th employed/onlooker bee finds a position that is better than the previous one it stays at, trial(i) is reset to 1 (but not zero); otherwise, we add 1 to it. If trial(i) exceeds Limit, the i th employed bee will be reinitialized using (2). $\gamma(i)$ manipulates the exploitation accuracy, which decreases exponentially to α as trial(i) gradually approaches Limit. Here, α is a user-specified lower boundary of convergent scale. In this sense, the search process is intensified gradually with an increasing trial(i). That is, the search accuracy will be gradually enhanced before the position of the i th employed bee is eventually discarded by the whole bee swarm.

Figure 2 illustrates the flow chart of our IF-ABC algorithm. The pseudocode of IF-ABC for numerical optimization is given in Algorithm 1.

5. Experimental Results

In this section, besides the conventional ABC and IF-ABC, two state-of-the-art versions of ABC, named Gbest-ABC [41] and I-ABC [42], were tested on four ideal template matching cases as well as four practical cases. All the simulations were implemented in MATLAB R2010a and executed on an Intel Core 2 Duo CPU with 2 GB RAM running at 2.53 GHz under Windows XP. Every single type of experiment was repeated 500 times with different random seeds. User-specified parametric settings for these ABCs are listed in Table 1, where Limit denotes invalid trial time and SN stands for half of the swarm population. Satellite images involved in this paper originate from the Google Earth (visit <http://www.google.com/earth/>).

Figures 3, 4, 5, and 6 illustrate the comparative simulation results concerning the four ideal cases; in each case the template is exactly part of the source image. Some typical local optimal matching locations are plotted (see the red

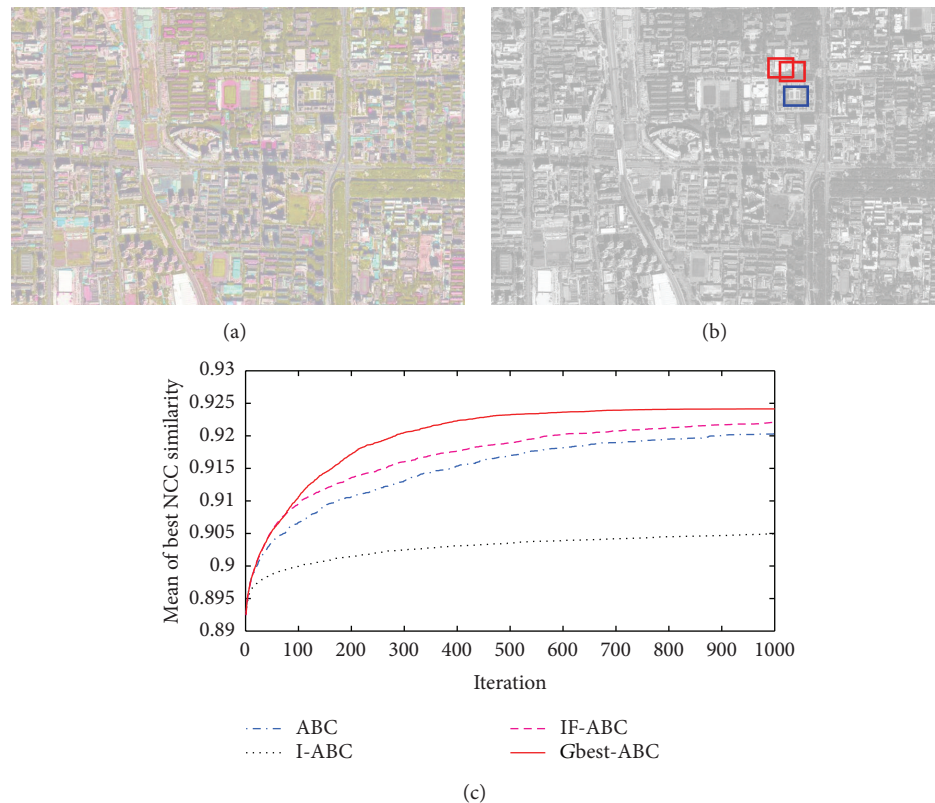


FIGURE 10: Practical template matching case 3. (a) Overexposed source image; (b) ground matching truth (blue box) and most common false matching results (red boxes); (c) comparative evolution curves of four ABC algorithms.

boxes in Figures 3(c), 4(c), 5(c), and 6(c)), together with the global optimal matching locations (see the blue boxes). For the convenience of evaluation, two indexes that reflect the convergence performances (i.e., the mean and convergence rate C.R.) are listed in Table 2, where MCN stands for the predefined maximum cycle number. In order to show the necessity of applying evolutionary algorithms, we also made comparisons regarding the average time consumptions between IF-ABC algorithm and the classical full search algorithm in the four template matching cases. Figure 11 demonstrates the comparative results, where each pair of comparison was repeated 50 times.

Similarly, another four cases of experiments with illumination disturbances considered were conducted. The results are shown in Figures 7, 8, 9, and 10, and the detailed comparisons are listed in Table 3.

Since template matching process does not take rotation or rescaling into consideration in our study, the two-dimensional domain is discrete (i.e., there exist a finite number of feasible solutions in total). Therefore, feasible solutions on a continuous domain need to be discretized so as to fit the objective function as in (1). As the objective function under consideration is a real-valued function of two variables, it is possible to plot the function surface using 3D graphics (as illustrated in Figures 3(e), 4(e), 5(e), and 6(e)).

6. Discussions

As shown in Table 2, the search process on the basis of IF-ABC is more efficient and robust than the other three algorithms. In detail, the convergence curves in Figures 3–6 suggest that the advantage of IF-ABC in case 1 is not as significant as it is in the rest of the three cases. It is also notable that the feasible solution surface in case 1 (see Figure 3(c)) is relatively smoother than the surfaces in the other three ideal cases (see Figures 4(c), 5(c), and 6(c)). In this particular case, classical methods are not inefficient. However, if the feasible solution surface is oscillatory or rugged, internal-feedback strategy will take effect. Take case 2 as an example, the overall surface is like a peak function (see Figure 4(c)), so it is more difficult to search for the global optimal location than case 1. The situations are quite similar in cases 3 and 4. In addition, although we observe from Table 2 that IF-ABC always possesses convergence rates far higher than those of the other three algorithms, the derivation of global optimum is not always guaranteed within the predefined iterations. Here comes the question: what is the advantage of IF-ABC in the case that a classical full search algorithm has been proposed nearly 40 years ago and always guarantees the derivation of a perfect matching result? Figure 7 gives a direct answer: the time consumption. For example, it takes IF-ABC an average of 3.6412 seconds to help find the optimal location (x^*, y^*) with $NCC(x^*, y^*) = 1$ in case 1, but it takes

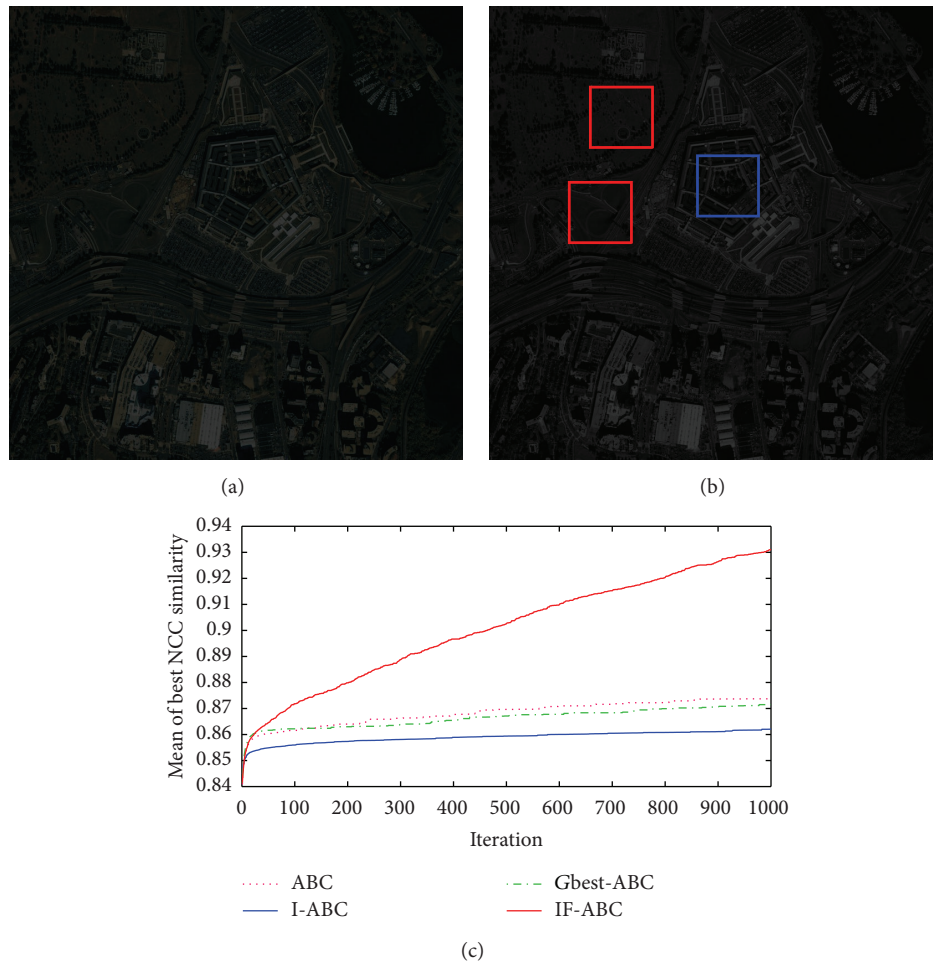


FIGURE 11: Practical template matching case 4. (a) Underexposed source image; (b) ground matching truth (blue box) and most common false matching results (red boxes); (c) comparative evolution curves of four ABC algorithms.

378.9624 seconds for the full search algorithm. We conclude that it may be an epitome that accounts for the prosperity of developments in intelligent algorithms.

Experiments mentioned above are based on a default assumption that a source image does perfectly contain the predefined template image. However, such assumption is never satisfied in practical terms. Therefore, to testify the practical value of template matching (concerning the utilization of NCC model), four practical cases were added in our work. The original source images are blurred, overexposed, or underexposed in those practical cases, and the advantages of IF-ABC are still remarkable, expect for practical case 3 (see Figure 10). We still are not aware of what makes the situation different. But we did not intend to conceal this truth. It should be recognized that there is still room for improvement in IF-ABC.

IF-ABC is mainly highlighted by its ability to fight against premature convergence. On one hand, IF-ABC discards the well-known roulette selection procedure. On the other hand, it advocates to fully utilize the variables or indexes hiding in the convergence system, rather than to seek for some hybridization from the “outside world.” The authors

believe that the roulette selection procedure does contribute a lot when optimizing some continuous unimodal objective functions. Nevertheless, when the objective function is discontinuous or multimodal, the roulette selection procedure will cause premature convergence because swarm diversity will be significantly reduced. In our point of view, convergence efficiency of the bees should be measured not by the corresponding objective function values, but by the fact whether they are better than the previous one. In this sense, it provides more possibilities for the so-called unqualified employed bees to be exploited locally by onlooker bees in the IF-ABC algorithm.

7. Conclusions and Future Work

The innovation of this work lies in the application of our previously proposed IF-ABC algorithm to solve the template matching problem. Experimental results clearly demonstrated the efficiency of IF-ABC in comparison with some state-of-the-art ABCs. Besides, it is also preliminarily confirmed that the NCC model works robustly when the source image can not 100% match the template image.

```

(1) Set population size SN, and maximum cycle number MCN. Set inefficient trial
    time counter  $\text{trial}(i) = 1$  ( $i = 1, 2, \dots, \text{SN}/2$ ).
(2) Randomly initialize locations of  $\text{SN}/2$  scout bees using (2)
(3) For iter = 1 to MCN, do % Employed bee phase
(4)   For item = 1 to  $\text{SN}/2$ , do
(5)     Generate  $\mathbf{X}_{\text{item}}$  for the item-th employed bee to search according to (3)
(6)     If better position is found, then % Implementation of greedy selection
(7)       Update  $\mathbf{X}_{\text{item}}$  and set  $\text{trial}(\text{item}) \leftarrow 1$ 
(8)     Else
(9)        $\text{trial}(\text{item}) \leftarrow \text{trial}(\text{item}) + 1$ 
(10)    End if
(11)  End for
(12) For item = 1 to  $\text{SN}/2$ , do % Onlooker bee phase
(13)   Choose item-th employed bee to follow, and then generate local search location using (5) and (6)
(14)   If better position is found, then % Implementation of greedy selection
(15)     Update  $\mathbf{X}_{\text{item}}$  and set  $\text{trial}(\text{item}) \leftarrow 1$ 
(16)   Else
(17)      $\text{trial}(j) \leftarrow \text{trial}(j) + 1$ 
(18)   End if
(19) End for
(20) Collect all item  $\in [1, \text{SN}/2] \cap \mathbb{Z}$  satisfying  $\text{trial}(\text{item}) > \text{Limit}$  in an index set  $\Omega$ 
(21) While  $\Omega \neq \emptyset$ , do
(22)   Choose one  $k \in \Omega$ 
(23)   Re-initialize location of the  $k$ th employed bee using (2)
(24)   Eliminate the index  $k$  from  $\Omega$ 
(25)   Set  $\text{trial}(k) = 1$ 
(26) End while
(27) Memorize best-ever solution in the current cycle of iteration
(28) End for
(29) Output best-ever solution

```

ALGORITHM 1

Our future work will focus on adopting this approach to some more complicated applications in the field of aeronautics and astronautics. Besides that, the efficiencies and robustness of different similarity measurement models (including the NCC model) under nonuniform illumination conditions will be theoretically evaluated.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the School of Advanced Engineering (SAE) in Beihang University. It was sponsored in part by the 5th and 6th National College Students' Innovation and Entrepreneurial Training Programs in China.

References

- [1] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, 1993.
- [2] Y. Lee, T. Hara, H. Fujita, S. Itoh, and T. Ishigaki, "Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique," *IEEE Transactions on Medical Imaging*, vol. 20, no. 7, pp. 595–604, 2001.
- [3] H. Peng, F. Long, and Z. Chi, "Document image recognition based on template matching of component block projections," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1188–1192, 2003.
- [4] G. Fu, H. Zhao, C. Li, and L. Shi, "Road detection from optical remote sensing imagery using circular projection matching and tracking strategy," *Journal of the Indian Society of Remote Sensing*, vol. 41, no. 4, pp. 819–831, 2013.
- [5] E. Cuevas, A. Echavarría, D. Zaldívar, and M. Pérez-Cisneros, "A novel evolutionary algorithm inspired by the states of matter for template matching," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6359–6373, 2013.
- [6] H. Grailu, M. Lotfizad, and H. Sadoghi-Yazdi, "An improved pattern matching technique for lossy/lossless compression of binary printed Farsi and Arabic textual images," *International Journal of Intelligent Computing and Cybernetics*, vol. 2, no. 1, pp. 120–147, 2009.
- [7] J. C. Yoo, H. K. Choi, and C. W. Ahn, "Template matching of occluded object under low PSNR," *Digital Signal Processing*, vol. 23, no. 3, pp. 870–878, 2013.
- [8] S.-D. Wei and S.-H. Lai, "Fast template matching based on normalized cross correlation with adaptive multilevel winner update," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2227–2235, 2008.

- [9] A. C. Berg and J. Malik, "Geometric blur for template matching," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 607–614, Kauai, Hawaii, USA, December 2001.
- [10] A. Rosenfeld and G. J. Vanderbrug, "Coarse-fine template matching," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 2, pp. 104–107, 1977.
- [11] N. Dong, C.-H. Wu, W.-H. Ip, Z.-Q. Chen, C.-Y. Chan, and K.-L. Yung, "An improved species based genetic algorithm and its application in multiple template matching for embroidered pattern inspection," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15172–15182, 2011.
- [12] J. W. Zhang and G. G. Wang, "Image matching using a bat algorithm with mutation," *Applied Mechanics and Materials*, vol. 203, pp. 88–93, 2012.
- [13] F. Liu, H. Duan, and Y. Deng, "A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching," *Optik*, vol. 123, no. 21, pp. 1955–1960, 2012.
- [14] H. B. Duan, C. F. Xu, S. Liu, and S. Shao, "Template matching using chaotic imperialist competitive algorithm," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1868–1875, 2010.
- [15] G. Rudolph, "Convergence rates of evolutionary algorithms for a class of convex objective functions," *Control and Cybernetics*, vol. 26, no. 3, pp. 374–390, 1997.
- [16] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Erciyes University Press, Erciyes, Turkey, 2005.
- [17] C. Ju and C. Xu, "A new collaborative recommendation approach based on users clustering using Artificial Bee Colony algorithm," *The Scientific World Journal*, vol. 2013, Article ID 869658, 9 pages, 2013.
- [18] H. Sun, H. Luş, and R. Betti, "Identification of structural models using a modified Artificial Bee Colony algorithm," *Computers & Structures*, vol. 116, pp. 59–74, 2012.
- [19] M. S. Uzer, N. Yilmaz, and O. Inan, "Feature selection method based on Artificial Bee Colony algorithm and support vector machines for medical datasets classification," *The Scientific World Journal*, vol. 2013, Article ID 419187, 10 pages, 2013.
- [20] H. Sun, H. Waisman, and R. Betti, "Nondestructive identification of multiple flaws using XFEM and a topologically adapting Artificial Bee Colony algorithm," *International Journal for Numerical Methods in Engineering*, vol. 95, no. 10, pp. 871–900, 2013.
- [21] B. Li, L. G. Gong, and W. L. Yang, "An improved Artificial Bee Colony algorithm based on balance-evolution strategy for unmanned combat aerial vehicle path planning," *The Scientific World Journal*, vol. 2014, Article ID 232704, 10 pages, 2014.
- [22] B. Li, Y. Li, and L. G. Gong, "Protein secondary structure optimization using an improved Artificial Bee Colony algorithm based on AB off-lattice model," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 70–79, 2014.
- [23] Z. Yin, X. Liu, and Z. Wu, "A multiuser detector based on Artificial Bee Colony algorithm for DS-UWB systems," *The Scientific World Journal*, vol. 2013, Article ID 547656, 8 pages, 2013.
- [24] A. Banharnsakun, B. Sirinaovakul, and T. Achalakul, "Job shop scheduling with the best-so-far ABC," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 583–593, 2012.
- [25] M. K. Apalak, D. Karaboga, and B. Akay, "The Artificial Bee Colony algorithm in layer optimization for the maximum fundamental frequency of symmetrical laminated composite plates," *Engineering Optimization*, vol. 46, no. 3, pp. 420–437, 2014.
- [26] Q. K. Pan, L. Wang, J. Q. Li, and J. H. Duan, "A novel discrete Artificial Bee Colony algorithm for the hybrid flowshop scheduling problem with makespan minimisation," *Omega*, vol. 45, pp. 42–56, 2014.
- [27] M. Yahya and M. P. Saka, "Construction site layout planning using multi-objective Artificial Bee Colony algorithm with Levy flights," *Automation in Construction*, vol. 38, pp. 14–29, 2014.
- [28] S. Zhong, Y. F. Dong, and A. Sarosh, "Artificial Bee Colony algorithm for parametric optimization of spacecraft attitude tracking controller," in *Foundations and Practical Applications of Cognitive Systems and Information Processing*, pp. 501–510, Springer, Berlin, Germany, 2014.
- [29] S. Biswas, S. Das, S. Debchoudhury, and S. Kundu, "Co-evolving bee colonies by forager migration: a multi-swarm based Artificial Bee Colony algorithm for global search space," *Applied Mathematics and Computation*, vol. 232, pp. 216–234, 2014.
- [30] J. Zhou, X. Liao, S. Ouyang, R. Zhang, and Y. Zhang, "Multi-objective Artificial Bee Colony algorithm for short-term scheduling of hydrothermal system," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 542–553, 2014.
- [31] A. Yurtkuran and E. Emel, "A modified Artificial Bee Colony algorithm for p -center problems," *The Scientific World Journal*, vol. 2014, Article ID 824196, 9 pages, 2014.
- [32] H. Garg, "Solving structural engineering design optimization problems using an Artificial Bee Colony algorithm," *Journal of Industrial and Management Optimization*, vol. 10, no. 3, pp. 777–794, 2014.
- [33] J. M. García-Torres, S. Damas, O. Cordón, and J. Santamaría, "A case study of innovative population-based algorithms in 3D modeling: Artificial Bee Colony, biogeography-based optimization, harmony search," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1750–1762, 2014.
- [34] M. K. Ahirwal, A. Kumar, and G. K. Singh, "Adaptive filtering of EEG/ERP through Bounded Range Artificial Bee Colony (BR-ABC) algorithm," *Signal Processing*, vol. 25, pp. 164–172, 2014.
- [35] J. A. Bullinaria and K. AlYahya, "Artificial Bee Colony training of neural networks," in *Nature Inspired Cooperative Strategies for Optimization (NICSO '13)*, vol. 512, pp. 191–201, Springer, 2014.
- [36] D. Aydın, S. Özyön, C. Yaşar, and T. Liao, "Artificial Bee Colony algorithm with dynamic population size to combined economic and emission dispatch problem," *International Journal of Electrical Power & Energy Systems*, vol. 54, pp. 144–153, 2014.
- [37] K. Buyukozkan and A. Sarucan, "Applicability of Artificial Bee Colony algorithm for nurse scheduling problems," *International Journal of Computational Intelligence Systems*, vol. 7, supplement 1, pp. 121–136, 2014.
- [38] B. Li and Y. Yao, "An edge-based optimization method for shape recognition using atomic potential function," In press.
- [39] B. Li, "Research on WNN modeling for gold price forecasting based on improved Artificial Bee Colony algorithm," *Computational Intelligence and Neuroscience*, vol. 2014, Article ID 270658, 10 pages, 2014.
- [40] B. Li, L. G. Gong, and Y. Yao, "On the performance of internal feedback artificial bee colony algorithm (IF-ABC) for protein secondary structure prediction," in *Proceedings of the 6th International Conference on Advanced Computational Intelligence (ICACI '13)*, pp. 33–38, Hangzhou, China, 2013.

- [41] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.
- [42] G. Li, P. Niu, and X. Xiao, "Development and investigation of efficient Artificial Bee Colony algorithm for numerical function optimization," *Applied Soft Computing*, vol. 12, no. 1, pp. 320–332, 2012.

