D

Ε

 H_i

 L_n

Ν

 n_i

 P_i

 P_s

 S_k

 w_f

х

y β

γ θ

λ

 λ_k

 μ_k σ

Biologically Inspired Model with Feature Selection for Target Recognition Using Biogeography-Based Optimization

Haibin Duan* and Yimin Deng[†]

State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, 100191 Beijing, People's Republic of China

DOI: 10.2514/1.I010122

To detect salient ground targets precisely and rapidly during aerial reconnaissance, this paper describes a novel object recognition method based on the feature selection of a biologically inspired model and biogeography-based optimization. As a promising approach to object recognition, the biologically inspired model is a hierarchical system of building an increasingly complex and invariant feature representation, which closely follows the process of object recognition in the visual cortex. These scale- and position-tolerant features are constructed by alternating between a template-matching and a maximum-pooling operation. Because of the many patches extracted in the standard biologically inspired model, the random mechanism may extract patches from irrelevant parts of an image and consume a lot of time. In this work, a feature selection method is proposed based on a new population-based evolutionary algorithm called biogeography-based optimization to choose the proper set of patches with high accuracy of classification and recognition. A support vector machine classifier is used for evaluation of the fitness function in biogeography-based optimization and to calculate the recognition rate in testing. A series of experiments are conducted, and the comparative results demonstrate the feasibility and effectiveness of the approach.

Nomenclature

- $A_{\rm cc}$ = classification accuracy
 - dimension of C2 feature vector =
- G_{xy} = output of Gabor filter at position (x, y)
 - maximum emigration rate =
 - = one habitat (solution)
 - = maximum immigration rate
- Iter = maximum number of iterations
 - normalized summation of selected patches =
 - = mutation rate of modifying a habitat
- m_s = population size
- N^{Σ} = size of a grid cell
- $n_{\rm elite}$ = number of elitism
 - = size of the patch P_{i}
 - one patch extracted at the C1 layer =
 - = probability of the habitat containing exact species
 - = number of species
- w_a = weight associated to the error rate
 - = weight associated to the normalized summation of selected patches
 - horizontal coordinate =
 - = vertical ordinate
 - = sharpness of the exponential function
 - = aspect ratio of Gabor filter
 - = orientation of Gabor filter
 - = wavelength of Gabor filter
 - = immigration rate of the kth solution
 - = emigration rate of the kth solution
 - = effective width of Gabor filter

I. Introduction

N OWADAYS, unmanned aerial vehicles (UAVs) [1] have been widely applied to information acquirement, threat reconnaissance, cooperative surveillance, heterogeneous cooperation [2], and high-speed strikes [3]. Automatic target recognition is a key problem to be solved for fulfilling these complicated tasks. The target-tracking performance is necessary for improving in aerial tasks, and researchers have proposed many algorithms such as the shape-matching approach with optimized edge potential function [4], vision-based feature matching [5,6], navigation [7,8], and real-time task scheduling [9]. However, understanding how the visual cortex recognizes objects is a critical question for neuroscience [10]. Establishing an emulating cortex recognition system has always been a challenging issue. References [11–13] have investigated the biological

Received 10 March 2013; revision received 11 February 2014; accepted for publication 11 March 2014; published online 4 July 2014. Copyright © 2014 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 2327-3097/14 and \$10.00 in correspondence with the CCC

School of Automation Science and Electrical Engineering; hbduan@buaa.edu.cn.

[†]School of Automation Science and Electrical Engineering.

feasibility of explaining aspects of higher-level visual processing. The aim of object recognition can be understood by representing objects as collections of view-specific features and taking the view that the basic recognition processes occur in a bottom-up way. Although these models have been tested on some natural images and perform well on some specific datasets, these neurobiological models of object recognition in cortex are difficult to apply in real-world scenes.

As described by Riesenhuber and Poggio, the hierarchical model [14,15] is consistent with the physiological data and quantitative theory from primates' visual cortex. This model consists of a hierarchy of layers with both linear (S units, performing template matching) and nonlinear operations (C pooling units, performing a maximum operation). It can handle the complex visual tasks and make testable predictions. A standard model of the visual cortex [16] summarizes object recognition in a quantitative way and consists of four layers of alternate simple S units and complex C units. Some experimental results have verified that this model can quantitatively duplicate the generalization properties of remaining highly selective for particular objects while being invariant to a range of scales and positions. Serre et al. established an extended standard model [10,17] by learning the feature vocabulary from images, and they applied it to the real-world object recognition. In this framework, a prototype is an image patch, which can be randomly selected from training images. C2 feature extraction can be interpreted as the best match between the S2 units and given prototypes. Deng and Duan proposed hybrid C2 features and a spectral residual approach [18]. Their approaches can be applied to fast location of object regions. However, the model in [18] has some shortcomings in random mechanisms. As the prototypes are randomly selected, the reliability of the matching requires a large number of prototypes, which makes the computational burden extremely high [19,20]. Furthermore, this biologically inspired model (BIM) is a feedforward procedure, which takes the first process in the ventral stream of primate visual cortex into account. To achieve stable performance, a large number of patches are randomly sampled from training images. As many patches are required to be extracted from the background instead of target regions, mismatches would easily appear. Without a predictionverification loop to remove redundant patches, this feedforward procedure greatly increases the computational burden and results in a large number of mismatches.

To solve this problem, there has been recent interest in developing feature selection algorithms to reduce the side effect of too many patches. In [19,21], Huang et al. developed the enhanced BIM by removing uninformative inputs via imposing sparsity constraints and applying a feedback loop to middle-level feature selection. A boosting-based feedback algorithm was proposed to replace the random patch selection stage in the BIM. This algorithm can significantly reduce the number of prototypes for matching, and thus greatly speeds up the classification procedure. Eliasi et al. [22] applied two clustering methods, K means and sequential backward feature selection, to *S*2 layer features optimization. However, this sequential backward features from training. Mutch and Lowe [23,24] improved the BIM by using sparsity, lateral inhibition, and a simple feature selection technique based on the binary support vector machines (SVMs). Through selecting the features with high weight, the resulting model with dropping features can improve classification performance with more economical computational resources. However, these extensions of feature selection can hardly guarantee that the set of selected patches are globally optimal ones. To solve this problem, some metaheuristic search algorithms, like particle swarm optimization (PSO), genetic algorithm, and evolutionary optimization, have been proposed for feature selection [25–27]. The experimental results show that these optimization algorithms are suitable for solving feature selection problems.

By applying the mathematics of biogeography to evolutionary computation, Simon [28] proposed a new bioinspired computing algorithm: biogeography-based optimization (BBO). BBO shares information between individuals just as species migrate between islands to get better solutions [29]. Markov models are also derived for BBO with selection, migration, and mutation operators [30], which can provide theoretically exact limiting probabilities for each possible population distribution for a given problem. BBO has shown good performance, both on benchmark problems [28] and on real-world problems (e.g., aircraft engine sensor selection [28], economic load dispatch [31], and satellite image classification [32]).

To reduce the heavy computation load and achieve high recognition rate in reconnaissance missions of UAVs, this paper proposes a novel feature selection scheme on the basis of BIM and BBO. We attempt to adopt this optimization procedure to get the best set of patches in training and to reduce the number of patches or features by eliminating redundant patches. The relevant properties of data can be obtained by selecting the most informative patches. With the patches extracted in the standard BIM, the feature selection scheme by using BBO can choose a number of patches with high recognition rates and few redundant patches. The set of patches is evaluated by recognition rate, which can be calculated by using a SVM classifier and the fitness function in BBO. Our model can significantly reduce the number of redundant patches, and hence achieve a high recognition rate, which is difficult for fast object recognition in aerial surveillance.

The rest of the paper is organized as follows. Section II first presents a brief description and mathematical formulation of the feature selection problem in BIM and BBO strategies, and then it gives a detailed description of our presented approach. The parameter settings and simulation studies are discussed in Sec. III. Our concluding remarks and future work are contained in Sec. IV.

II. Detailed Implementation of Our Presented Approach

A. Biologically Inspired Model of Visual Cortex

The architecture of the BIM has two primary layers: the *S* layer and the *C* layer. It consists of four layers of computational units: S1, C1, S2, and C2, where simple *S* units alternate with complex *C* units. The overview to obtain the output of this biologically inspired model of the visual cortex can be summarized as follows:

Step 1: Obtain the responses of the S1 layer (i.e., simple cells in the first visual cortical stage).

In this layer, an initial gray-value input image is firstly analyzed by a multidimensional array of simple *S*1 units. Batteries of Gabor filters [33] are applied to an input image, which have been shown to provide a good model of cortical simple cell receptive fields and good orientation and frequency selectivity for image processing. This processing can been described by

$$G_{xy} = \exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right) \times \cos\left(\frac{2\pi}{\lambda}X\right)$$
(1)

where $X = x \cos \theta + y \sin \theta$ and $Y = -x \sin \theta + y \cos \theta$. Orientation θ , aspect ratio γ , effective width σ , and wavelength λ are four filter parameters that can be adjusted. The ranges of *x* and *y* are associated with the scales of the Gabor filters, and θ controls the orientations. After convolving the input image with the filters, 64 maps (16 scales × 4 orientations) are obtained and arranged in eight bands. Each filter band contains two sizes at all four orientations.

Step 2: Get the C1 features, corresponding to complex cells.

The C1 units describe complex cells in the visual cortex, corresponding to cortical complex cells that show some tolerance to shift and size. As the key procedure to generate a C1, nonlinear max operation selects the maximum of S1 units in a local area and processes the responses of Step 3: Compute the S2 units through the learning process.

In the extended standard model, the learning is conducted in the S2 stage. This part consists of two stages: the learning stage and the classification stage. In the former stage, the learning process corresponds to selecting a set of prototypes or patches for the S2 units. During training, large numbers of patches of various sizes at random positions and all four orientations are extracted at the C1 layer. One patch P_i of size $n_i \times n_i$ contains $n_i \times n_i \times 4$ elements. Then, those patches are stored as prototypes of the S2 units after the training process ends. In the classification stages, the S2 units describe the similarity between C1 images and prototypes via convolution operation as

$$S2 = \exp(-\beta ||P - P_i||^2)$$
(2)

where β defines the sharpness of the exponential function, *P* is one patch obtained from the test image, and *P_i* is one of the patches from the stored prototype. The default value of β is chosen as 1. This process can be realized with the radial basis function [34], which is a real-valued function for which the value depends only on the distance from the origin. Each *S*2 response depends on the Euclidean distance between the new input and the stored prototype in a Gaussian function.

Step 4: Acquire the final C2 features.

The final set of C2 responses is computed by taking a global maximum operation over all scales and positions for each S2 output corresponding to one prototype. Since each prototype corresponds to a group of S2 images, a vector of NC2 values is produced for each input image, where N corresponds to the number of stored prototypes. All these steps ensure that the C2 features have the property of shift and scale invariance, which can be used in a classifier for the final analysis of object recognition.

The major limitation of the BIM is that the patches are extracted randomly. High reliability of the match depends on a large number of randomly selected prototypes, which leads to high computational burden. Furthermore, as many patches are extracted from the background instead of target regions and the patches have varying degrees of usefulness for the classification task, mismatches would easily appear. The feedforward procedure in the BIM greatly increases computation cost and causes the mismatches, which is almost impossible for practical recognition applications without the prediction–verification loop to remove redundant patches.

B. Basic Principles and Mathematical Description of BBO

As presented in [28], BBO is a population-based biologically inspired optimization method, inspired by mathematical biogeographic models. Biogeography is the nature's way of distributing species. It is used in BBO to determine the mutation rate to prevent the local trapping of the algorithm.

In BBO, each possible solution is like an island, with their features characterizing habitability. The goodness of each solution is called its habitat suitability index (HSI). A good solution is analogous to an island with a high HSI, and a poor solution represents an island with a low HSI [28]. High-HSI solutions resist change more than low-HSI solutions, whereas low-HSI solutions tend to copy good information from high-HSI solutions. As a result, the shared information remains in the high-HSI solutions and appears as new features at the same time in the low-HSI solutions. In biogeography, this procedure is similar to some representatives of a species migrating to a habitat, whereas other representatives remain in their original habitat. Poor solutions accept a lot of new features from good solutions that may improve the quality of those solutions. The usage of migration operator probabilistically shares information between solutions based on the fitness values, and thus enables BBO to enhance the exploitation performance.

BBO consists of two mechanisms: migration and mutation. It has certain unique features as mentioned next:

1) Although their characteristics change as the optimization process progresses, all solutions survive forever. Without the crossover operation, solutions get fine-tuned gradually as the process goes on through the migration operation. Elitism operation is adopted to improve the algorithm with more efficiency.

2) Due to its new type of mutation operation, solutions do not have the tendency to cluster in BBO, which is different from PSO.

3) By probabilistically sharing information between solutions, poor ones accept new features from good ones, which may improve the quality of those solutions. This is a unique feature of the BBO algorithm to make constraint satisfaction much easier.

In BBO, a habitat *H* is initialized randomly with a vector of *D* integers that characterize habitability integers [35]. Then, each individual of population is evaluated by a fitness function before optimizing. As a follow-up, migration and mutation steps are taken to reach the global optimum. In migration, the information is shared between habitats that depend on emigration rates μ and immigration rates λ of each solution. The *k*th ($k \in \{1, ..., N\}$) individual has its own μ and λ for *S* number of species, which are computed, respectively, by [28]

$$\mu_k = \frac{ES_k}{S_{\text{max}}} \tag{3}$$

$$\lambda_k = I \left(1 - \frac{S_k}{S_{\max}} \right) \tag{4}$$

where *E* is the maximum emigration rate, *I* is the maximum immigration rate, and in general, the immigration and emigration curves are straight lines for the special case of E = I. Therefore, a habitat with few species (i.e., low-HSI poor solution) has lower μ and higher λ , whereas the habitat with more species (i.e., high-HSI good solution) has higher μ and lower λ . In the migration operator, poor solutions accept more useful information from good solutions. Each nonelite habitat can be potentially modified on the basis of a habitat modification probability. As the migration operator is only applied for candidate nonelite habitats, some of the best habitats in the previous generation are preserved as elites to improve the exploitation ability.

Once the migration process is over, a mutation operator is probably applied to each nonelite habitat to increase the diversity of the population. The mutation operator modifies a habitat's species randomly based on mutation rate m, which is expressed as [28]

$$m_s = m_{\max} \left(\frac{1 - P_s}{P_{\max}} \right) \tag{5}$$

Table 1 Performance between our proposed model and the original BIM in six categories^a

	Our model			Origina	al BIM	
	Feature number	Time	EER	Feature number	Time	EER
Fighter	220	5.188	0.967	220	5.065	0.830
-				1000	18.197	0.847
Helicopter	210	5.086	0.917	210	4.905	0.883
				1000	18.408	0.877
Ketch	174	4.446	0.900	174	4.320	0.857
				1000	17.901	0.840
Motorbike	171	4.420	0.983	171	4.270	0.923
				1000	17.743	0.937
Car	168	4.400	0.983	168	4.223	0.933
				1000	17.677	0.950
Airplane	166	4.384	0.967	166	4.188	0.920
*				1000	17.685	0.913

^aThe measures are a number of selected features, EERs, and extraction times.

where m_{max} is a user-defined parameter, P_s is the probability that the habitat contains exact species number *S*, and P_{max} is the maximum. Each habitat member has an associated probability, which indicates the likelihood that it exists as a solution to a given problem. In the mutation operator of BBO, both low- and high-HSI solutions are likely to mutate. This provides a chance of improving both types of solutions. Elitism operator is also adopted to preserve some types of elites, which can keep the global best elites and abandon the worse ones.

The computational complexity of BBO in terms of the $\mathcal{O}()$ notation is also presented in this section. The detailed complexity of BBO is shown in Table 1, where $F = \mathcal{O}(F)$ is the computational complexity of the objective function; and Iter, N, and D denote the maximum number of iterations, population size, and the dimension of the C2 feature vector, respectively. Then, the computational complexity of the migration process is $\mathcal{O}(N^2 \cdot D)$, whereas the one of mutation is $\mathcal{O}(N)$. Therefore, the computational complexity of complete BBO is $\mathcal{O}(\text{Iter} \cdot (N^2 \cdot D + N))$. The BBO algorithm can get the optimized solution with less computational cost.

C. Feature Selection Scheme Using BBO

Based on the aforementioned analysis, a new approach to implement the BBO algorithm will be described to solve the feature selection problems in the BIM (see Fig. 1). The feature selection problem is converted to a task of finding the best patch set that minimizes a given objection function. The classification accuracy A_{cc} and the number of selected patches L_{patch} are the two criteria considered in a fitness function. To find the smallest patch set that improves classifier performance, a fitness function similar to [25] is introduced in BBO for the optimization problem, which is defined by

$$fitness = w_a \times (1 - A_{cc}) + w_f \times L_n \tag{6}$$

where w_a represents the predefined weight associated to the error rate of the classifier $(1 - A_{cc})$, and w_f denotes the predefined weight associated to the normalized summation L_n of the selected patches. The normalized summation L_n is calculated by L_{patch}/D , where D represents the dimension of the C2 feature vector (i.e., the number of patches extracted at the C1 layer). To precisely establish a BBO-based feature selection optimization approach, the following steps must be conducted.

1) The first step is data preprocessing. Similar to binary code in the genetic algorithm, we define the patch set by $\lambda_{\text{Patch}} = [\lambda_1, \lambda_2, \dots, \lambda_D]^T$, where λ_i indicates the position value in λ_{Patch} . If the *i*th patch is selected, λ_i is equal to 1; otherwise, it is 0. Then, L_n is a normalized summation L_{patch} of selected patches. Two parts of the fitness function are confirmed in the same range.

2) The second step is solution evolution. To evaluate classifier performance, the binary patch set is converted to phenotype data. To save the computation time and reduce the training cost, a large number of patches are extracted previously along with corresponding *C*2 values. One patch with four orientations corresponds to a feature vector (i.e., the four-dimensional row of the *C*2 units), which can be easily found in the right position of the stored dataset on the basis of binary solution of patch set. Then, these *C*2 features are used to train the SVM classifier to get the performance evaluation A_{cc} of the validation set. SVM is a powerful classifier to achieve a fast convergent rate and is widely used in classification applications. In this paper, OSU SVM[‡] is chosen as a toolbox for creating models for classification. Our experiments are performed by this SVM with a linear kernel (i.e., LinearSVC) whereas the parameter *C* is set to a default value (i.e., C = 1).

As stated previously, the basic BBO algorithm for feature selection can be summarized by using the pseudocode shown as follows:

For step 1, initialize the BBO parameters: S_{max} , E, I, m_{max} , w_a , w_f , maximum number of iterations Iter, elitism parameter n_{elite} , population number N, and initial random set of solutions (habitats) H_i (i = 1, 2, ..., N).

For step 2, extract corresponding patches and C2 features for each solution.

For step 3, train SVM classifier and evaluate each fitness.

For step 4, compute S_i , λ_i and μ_i for each solution.

For step 5, apply elitism to preserve n_{elite} best solutions as elites.

For step 6, select remaining solution H_i for migration based on probability proportional to λ_i and μ_i .

For step 7, select solution H_i to update with mutation based on probability.

For step 8, recompute fitness of the modified solutions.

For step 9, implement elitism to replace and preserve n_{elite} elites and find the current best solution.

For step 10, go to step 2 for the next iteration if needed.

It is worth noting that our proposed feature selection scheme in this paper is different from the ones introduced by Mutch and Lowe [23,24] and Huang et al. [19,21]. Mutch and Lowe [23,24] used a simple feature selection technique based on binary SVMs. It selected features with highly weighted values by the SVM, whereas it was not adaptive and required a predefined value to decide the iteration of dropping features. Huang et al. [19,21] applied a feedback loop with a boosting-based feedback algorithm to the middle-level feature selection stage. It significantly reduced the

^{*}Data available online at http://sourceforge.net/projects/svm/ [retrieved 2014].



Fig. 1 Architecture 9 of our feature selection 10 model.



Fig. 2 Diagram of the object recognition system applied to every input test image.

number of prototypes for matching; thus, it greatly improved the classification procedure and reduced mismatches. However, it is difficult to obtain optimized patches using their method on the basis of chosen SVMs corresponding to the most discriminative patches. Theoretical analysis [36] has proved that the population-based bioinspired optimization algorithm can converge to the global optimal solution set. Given infinite computational resources, our proposed evolutionary feature selection scheme can ensure the globally optimal set of patches.

D. Framework of Object Recognition System

According to feature selection, the process of our proposed object recognition method is presented in Fig. 2.

Our feature selection scheme using BBO chooses the proper set of patches with a high recognition rate and less redundant information. The set of patches is evaluated by the recognition rate calculated with the SVM classifier and fitness function in BBO. After the training process, C2 units of the test set are generated with selected patches and sent to the SVM classifier chosen in the training stage to make a final decision.

III. Experimental Results and Analysis

To measure the performance of our method, the following datasets from the California Institute of Technology (Caltech) databases[§] were used. In our experiments, we first chose six classes of objects (fighter, helicopter, ketch, motorbike, airplane, and car) from the Caltech 101 database. All these objects are with potential threats in battlefield reconnaissance. Figure 3 shows the sample images of each category. The datasets are randomly divided into three parts (i.e., training set, validation set, and test set). The training set is used to build the model to get a large number of patches and training features, and then a validation set is used to measure the performance of feature selection using the BBO algorithm. The chosen patches and trained SVM classifier are applied on the test set.

As a preprocessing step, we apply scaling to prevent numerical difficulties in the calculation and ensure all filters can be used effectively on too small or too large inputs. All images are normalized in size as 120×90 pixels. Also, converting color images to gray values is necessary to simplify computation. All experiments are performed using MATLAB 7.14.0(R2012a) on a PC with a Core II 2.4 GHz CPU and 3G of RAM. The main parameters in our experiments for BBO and the BIM were set as follows:

1) For the BIM, the S1 and C1 layers parameters are taken as in [10]. The number of initial patches is detailed in each experiment, changing from 10 to 1000.

2) For BBO, $S_{\text{max}} = N = 50$, D = 1000, E = I = 1, $n_{\text{elite}} = 2$, $m_{\text{max}} = 0.5$, and Iter = 1000.

[§]Data available online at http://www.vision.caltech.edu/Image_Datasets/Caltech101/ [retrieved 2014].



Fig. 3 Sample images from Caltech 101 database.

A. Comparisons of Different Feature Numbers

To investigate the contribution of patch number on performance of accuracy and speed, all samples of categories are randomly divided into three equal parts (i.e., training set, validation set, and test set). In our experiment, each set of different categories is assigned with the same number of samples. As some categories in the Caltech 101 database contain very few samples, we adopt the eclectic strategy. Forty positive samples and 40 negative samples are selected at random as the training set. Thirty other positive samples and 30 other negative samples are randomly selected as the validation set. Thirty positive and negative samples, respectively, are selected in the remaining for testing.

To conduct this experiment, we first set the different patch numbers to be 10, 50, 100, 200, 500, and 1000. Each patch contains four orientations and corresponds to four *C*2 units. The results are averaged over five independent trials. Figure 4 shows the performance of accuracy and speed of different feature numbers. The accuracy performance is measured by equal error rate (EER), which means the classification rate in all experiments of this paper. The performance of speed is the computation time of extracting *C*2 features of one test image.

Figure 4 shows the comparisons among the six categories. Performances are still satisfactory with a small number of features. Even though the performance could be improved generally with more features or patches, the time is increased approximately with an exponentially expandation. For example, the BIM takes approximate about 1.8 s to deal with an image in the case of selecting 10 patches and about 18 s in the case of selecting 1000 patches. The large computing time is the bottleneck for application, especially when there are a large amount of test images. Figure 4 shows that the performances of speed are almost the same among different categories with an authorized number of features. However, the classification rates have some differences between categories. This is because the six categories selected from the Caltech 101 database have different backgrounds and angles of rotation.

B. Evaluation of Our Proposed Model

To evaluate our proposed model and investigate the effectiveness, a series of experiments are conducted on the aforementioned six categories. For feature selection, 1000 initial patches are used in our experiments. In the fitness evaluation part, we set $w_a = 0.8$ and $w_f = 0.2$, where w_a is the important adjustable parameter, and the test on the influence of different w_a for feature selection and classification rate is conducted in the next section. As a limit of training time, the maximum number of iteration is set to 1000.

Figures 5 and 6 show the performances of our optimization method among the six categories. Figure 5 gives the evaluation of our optimization method among three categories: fighter, helicopter, and ketch. The first row shows curves of the fitness value in the BBO algorithm, and the second row shows curves of accuracy (classification rate) and length (number of selected patches). The fitness value is the weighted sum of accuracy and length. A lower fitness value usually results in better optimization results. Figure 6 shows the experimental results of the other three categories: motorbike, airplane, and car. All the results show that the accuracy increases along with reducing the patch number using our model, and some jumping and high gradient change of the curves exist because of the probabilistic operation in BBO. The fitness curve has the convergent tendency with the iteration limited to 1000. In our experiment, about 20% of the patches are selected with high accuracy in 1000 iterations, which is sufficient for the recognition system to obtain a stable performance.



100 200 300

Fitness

0.22



500 600 700 800 900 1000

Iteration

Fighter





b) Curves of accuracy and length for fighters



c) Curves of fitness value in BBO algorithm for helicopters

d) Curves of accuracy and length for helicopters



Fig. 5 Evaluation of our optimization method among three categories: fighter, helicopter, and ketch.

A performance comparison between our model and the original BIM is shown in Table 1. Three measures (i.e., number of features, time of extracting C2 features per test image, and classification accuracy) are considered. Our model can automatically obtain a number of selected features with a tradeoff between accuracy and speed. With the same feature number, a higher recognition rate can be obtained with our model because of reduction of a large amount of redundant patches. A large amount of time is needed in the original model to acquire a high accuracy, which is impracticable for fast object recognition.

C. **Evaluation of Weighting Parameters**

In the feature selection procedure, the number of patches is evaluated by the weighted sum of accuracy and speed. An important adjustable parameter is w_a , so we test the influence of different weights for recognition accuracy. The results are shown in Fig. 7 and Table 2. For all six categories, fitness functions with ($w_a = 0.95$, $w_f = 0.05$) have the minimum fitness value. Higher weight associated with accuracy can bring a better EER in some cases, whereas a smaller number of selected features can be obtained with a higher weight associated to the patch's length.

D. Comparisons of Optimization Methods

To investigate the effectiveness of BBO for feature selection, we perform a comparison between BBO and another optimization algorithm.

Motorbike

0.6

Accuracy

Motorbike



Fig. 6 Evaluation of our method among three other categories: motorbike, airplane, and car.

In this experiment, binary particle swarm optimization (BPSO) is introduced [25,37]. BPSO is an extended version of the basic PSO algorithm for optimization of the discrete binary combinational problem. The particles take the values of binary vectors of encoding length and the velocity corresponds to the probability of 1 bit to take the value 1. The position of each individual particle in BPSO is given in binary form, and the changes in particle velocity reflect a change in the probability of finding the particle in one state or another.

The basic parameters of the BIM and fitness function are set to be same in the experiments. We select the weighting parameters in both fitness functions with ($w_a = 0.95$, $w_f = 0.05$) and use 1000 initial patches. Figure 8 and Table 3 show the comparisons of BPSO and BBO for feature selection with the BIM. In Fig. 8, BPSO has a fast convergence speed and achieves steady state in about 100 iteration rounds. The computational complexity of BPSO is on the order of $\mathcal{O}(NX)$, where N is the number of initial particles and X is the number of iterations to reach the global optimum for PSO. Let $F = \mathcal{O}(F)$ be the computational complexity of the objective function and **D** be the dimension of function values. Then, the expected number of generations for the BPSO to optimize the function is $\mathcal{O}(ND \log N)$. Thus, the computational complexity of BPSO is $\mathcal{O}(N^2D \log N)$, which is lower than that of BBO. Even though BPSO has fast convergence speed, BBO has a lower fitness value and shows better performance than BPSO. In Table 3, comparisons of a number of selected features and recognition accuracy between two methods are given. In the same case, BBO obtains a smaller number of selected features with little better accuracy.

0.12

Helicopter

Fighter



Fig. 7 Comparisons of different weight parameters among six categories.

E. Comparisons with Scale-Invariant Feature Transform Features

We also compare our model with several state-of-the-art approaches based on scale-invariant feature transform (SIFT) features [38] on the Caltech 101 data. The SIFT features, on the basis of appearance at particular interest points, are local image features, of which the description has been used successfully in object recognition. Bosch et al. proposed Pyramid of Histograms of Orientation Gradients (PHOG) [39] by computing

 Table 2
 Experimental comparisons of different weight parameters on six categories

	$w_a = 0.80, w_f = 0.20$		$w_a = 0.95, w_f = 0.05$		$w_a = 0.50, w_f = 0.50$	
	Feature number	EER	Feature number	EER	Feature number	EER
Fighter	220	0.967	245	0.900	186	0.900
Helicopter	210	0.917	217	0.917	173	0.917
Ketch	174	0.900	185	0.917	131	0.883
Motorbike	171	0.983	163	0.983	145	0.983
Car	168	0.983	151	0.983	155	0.967
Airplane	166	0.967	259	0.967	152	0.983



dense multiscale SIFT features together with a spatial pyramid kernel. Their best experimental performance was 77.8% for classification of

 Table 3
 Experimental comparisons of BBO and BPSO for feature selection on six categories

Caltech 101. Lazebnik et al. established a spatial pyramid framework [40] for recognizing scene categories and achieved a high recognition rate on the Caltech 101 database with 30 training images and 50 test images per class. In this paper, the PHOG is trained on 15 training images per class

	BBO		BPSO	
	Feature number	EER	Feature number	EER
Fighter	245	0.900	319	0.900
Helicopter	217	0.917	319	0.917
Ketch	185	0.917	314	0.917
Motorbike	163	0.983	307	0.983
Car	151	0.983	279	0.967
Airplane	259	0.967	275	0.950

Table 4 Performance results and comparison with several approaches on Caltech-101

	Our model	PHOG [39]	Spatial pyramid [40]
Fighter	0.967	0.765	0.767
Helicopter	0.917	0.788	0.833
Ketch	0.900	0.841	0.883
Motorbike	0.983	0.771	0.967
Car	0.983	0.818	0.867
Airplane	0.967	0.747	0.950

Table 5 Comparison of several approaches on Caltech 256

	Our model	Original BIM	PHOG	Spatial pyramid
Radio telescope	0.833	0.816	0.788	0.783
Speedboat	0.850	0.767	0.750	0.700
Touring bike	0.983	0.883	0.850	0.900

Table 6				
Comparison results on				
the aircraft dataset				

Model	Aircraft
Our model	0.933
Original BIM	0.817
PHOG	0.800
Spatial pyramid	0.883



Fig. 9 Sample images of radio telescope and speedboat from Caltech 256 database.

and tested with the rest images on Caltech 101. Lazebnik et al.'s spatial pyramid model is trained and tested on the same sets as what we have used in our model (i.e., 40 images for training and 30 images for testing per class).

Table 4 summarizes the performance of our model compared with other state-of-the-art approaches. The performance measure is the classification accuracy. Results obtained by our model are comparable and superior to other approaches on the chosen Caltech 101 datasets. Our model can achieve an average recognition rate of 95.3%, which is higher than PHOG (78.3%) and spatial pyramid (87.8%) on these datasets.

To investigate the performance of our model on different databases, we also provide the comparison results on another popular database. The Caltech 256 database¹ contains 256 object classes plus a background class and 30,608 images. For the need of aerial reconnaissance in UAVs, we also chose the objects with potential significance for reconnaissance, such as the radio telescope in Fig. 9.

For the Caltech 256 database, we adopt the same strategy used for Caltech 101. Our model outperforms other approaches in all classes, as shown in Table 5. Our model can achieve above 85% average recognition rate on these classes from Caltech 256: much higher than PHOG and spatial pyramid. Compared to the original BIM, our model obtains better results with a higher recognition rate and smaller feature numbers (i.e., about 10% of the patches are selected). The comparison results in Table 5 verify the effectiveness of feature selection based on BBO in both accuracy and speed on computer vision tasks.

Finally, we report comparison results on the aircraft dataset in the view of aerial surveillance. Figure 10 shows sample images that contain aircraft at the airport and background. The experimental results are shown in Table 6 with comparison with the state-of-the-art approaches. The recognition rate of our model is 93.3% and higher than the others. By using the feature selection based on BBO, the number of selected features for training is about 50. The small number of features can help can help reduce much more computational time than the original BIM.

[®]Data available online at http://www.vision.caltech.edu/Image_Datasets/Caltech256/ [retrieved 2014].



Fig. 10 Sample images of aircraft and backgrounds.

F. Future Work

The proposed biologically inspired model with feature selection based on BBO is a positive strategy for the purpose of reducing features for training and lowering computational costs in testing. We have also tried to apply the other improved BBO models to template matching [36,41]. However, our proposed model requires more time to complete the training part. Time consumption is also the bottleneck for many biologically inspired models [42]. The processing speed of our model is the main limitation in real-world applications, such as the analysis of video. Another aspect in which we would like to further improve our model is to investigate the satisfactory explanation of the superiority of human-level recognition system.

IV. Conclusions

In this paper, a novel object recognition approach based on the BIM and BBO has been proposed. To solve the limitation of random mechanism in the BIM, a feature selection scheme based on BBO is proposed. By choosing a proper set of patches, the scheme can give high accuracy of the classification rate. Experiments on six categories from the Caltech 101 database have demonstrated that the model performs well in both recognition accuracy and extraction time. The feature selection scheme imposes BBO as an efficient optimization procedure and works on the patch layer. It takes about 1 h to complete the whole training process and obtain the optimized patches. However, this bioinspired computing method optimizes the model be more adaptive. It can also greatly reduce the redundant patches, and hence speed up the test processes significantly.

Acknowledgments

This work was partially supported by Natural Science Foundation of China under grants 61333004, 6127034, and 60975072; the National Key Basic Research Program of China (973 Project) under grants 2014CB04610 and 2013CB033503; the National Magnetic Confinement Fusion Research Program of China under grant 2012GB102006; the Top-Notch Young Talents Program of China, Aeronautical Foundation of China under grant 20135851042; and the Innovation Foundation of Beihang University for Ph.D. Graduates. The authors are grateful to Ben M. Chen, Robert C. Michelson, Delin Luo, Hong-Tao Liu, and Shiyu Zhao for their detailed guidance and valuable polishing of this work.

References

- Misovec, K., Inanc, T., Wohletz, J., and Murray, R. M., "Low-Observable Nonlinear Trajectory Generation for Unmanned Air Vehicles," *IEEE Conference on Decision and Control*, Vol. 3, IEEE, Piscataway, NJ, 2003, pp. 3103–3110. doi:10.1109/CDC.2003.1273100
- [2] Liu, S. Q., and Duan, H. B., "Unmanned Air/Ground Vehicles Heterogeneous Cooperative Techniques: Current Status and Prospects," SCIENCE CHINA Technological Sciences, Vol. 53, No. 5, 2010, pp. 1349–1355. doi:10.1007/s11431-010-0122-4
- [3] Li, P., and Duan, H. B., "Progress in Control Approaches for Hypersonic Vehicle," SCIENCE CHINA Technological Sciences, Vol. 55, No. 10, 2012, pp. 2965–2970.
 - doi:10.1007/s11431-012-5036-x
- [4] Xu, C. F., and Duan, H. B., "Artificial Bee Colony (ABC) Optimized Edge Potential Function (EPF) Approach to Target Recognition for Low-Altitude Aircraft," *Pattern Recognition Letters*, Vol. 31, No. 13, 2010, pp. 1759–1772. doi:10.1016/j.patrec.2009.11.018
- [5] Fravolini, M., Campa, G., and Napolitano, M., "Evaluation of Machine Vision Algorithms for Autonomous Aerial Refueling for Unmanned Aerial Vehicles," *Journal of Aerospace Information Systems*, Vol. 4, No. 9, 2007, pp. 968–985. doi:10.2514/1.17269
- [6] Fowers, S., Lee, D., Ventura, D., and Tippetts, B., "Novel Feature Descriptor for Low-Resource Embedded Vision Sensors for Micro Unmanned-Aerial-Vehicle Applications," *Journal of Aerospace Information Systems*, Vol. 10, No. 8, 2013, pp. 385–395. doi:10.2514/1.1010028
- [7] Johnson, E., Calise, A., Watanabe, Y., Ha, J., and Neidhoefer, J., "Real-Time Vision-Based Relative Aircraft Navigation," *Journal of Aerospace Information Systems*, Vol. 4, No. 4, 2007, pp. 707–738. doi:10.2514/1.23410
- [8] Wu, A., Johnson, E., Kaess, M., Dellaert, F., and Chowdhary, G., "Autonomous Flight in GPS-Denied Environments Using Monocular Vision and Inertial Sensors," *Journal of Aerospace Information Systems*, Vol. 10, No. 4, 2013, pp. 172–186. doi:10.2514/1.1010023
- [9] Clare, A. S., Cummings, M. L., and How, J. P., "Operator Objective Function Guidance for a Real-Time Unmanned Vehicle Scheduling Algorithm," *Journal of Aerospace Information Systems*, Vol. 9, No. 4, 2012, pp. 161–173. doi:10.2514/1.1010019
- [10] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T., "Robust Object Recognition with Cortex-Like Mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 3, March 2007, pp. 411–426. doi:10.1109/TPAMI.2007.56
- [11] Porat, M., and Zeevi, Y. Y., "The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, 1988, pp. 452–468. doi:10.1109/34.3910

- [12] Balakrishnan, N., Hariharakrishnan, K., and Schonfeld, D., "A New Image Representation Algorithm Inspired by Image Submodality Models, Redundancy Reduction, and Learning in Biological Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 9, Sept. 2005, pp. 1367–1378. doi:10.1109/TPAMI.2005.170
- [13] Miao, J., Qing, L., Zou, B., Duan, L., and Gao, W., "Top-Down Gaze Movement Control in Target Search Using Population Cell Coding of Visual Context," *IEEE Transactions on Autonomous Mental Development*, Vol. 2, No. 3, 2010, pp. 196–215. doi:10.1109/TAMD.2010.2053365
- [14] Riesenhuber, M., and Poggio, T., "Hierarchical Models of Object Recognition in Cortex," Nature Neuroscience, Vol. 2, No. 11, Nov. 1999, pp. 1019–1025. doi:10.1038/14819
- [15] Riesenhuber, M., and Poggio, T., "Neural Mechanisms of Object Recognition," Current Opinion in Neurobiology, Vol. 12, No. 2, April 2002, pp. 162–168. doi:10.1016/S0959-4388(02)00304-5
- [16] Riesenhuber, M., and Poggio, T., "Models of Object Recognition," Nature Neuroscience, Vol. 3, No. 11, Nov. 2000, pp. 1199–1204. doi:10.1038/81479
- [17] Serre, T., Wolf, L., and Poggio, T., "Object Recognition with Features Inspired by Visual Cortex," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, IEEE, Piscataway, NJ, 2005, pp. 994–1000. doi:10.1109/CVPR.2005.254
- [18] Deng, Y. M., and Duan, H. B., "Hybrid C2 Features and Spectral Residual Approach to Object Recognition," Optik International Journal for Light and Electron Optics, Vol. 124, No. 18, 2013, pp. 3590–3595. doi:10.1016/j.ijleo.2012.11.063
- [19] Huang, Y., Huang, K., Tao, D., Tan, T., and Li, X., "Enhanced Biologically Inspired Model for Object Recognition," *IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics*, Vol. 41, No. 6, Dec. 2011, pp. 1668–1680. doi:10.1109/TSMCB.2011.2158418
- [20] Song, D., and Tao, D., "Biologically Inspired Feature Manifold for Scene Classification," *IEEE Transactions on Image Processing*, Vol. 19, No. 1, 2010, pp. 174–184.
- doi:10.1109/TIP.2009.2032939
- [21] Huang, Y., Huang, K., Wang, L., Tao, D., Tan, T., and Li, X., "Enhanced Biologically Inspired Model," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ, 2008, pp. 1–8. doi:10.1109/CVPR.2008.4587599
- [22] Eliasi, M., Yaghoubi, Z., and Eliasi, A., "Intermediate Layer Optimization of HMAX Model for Face Recognition," *IEEE International Conference on Computer Applications and Industrial Electronics*, IEEE, Piscataway, NJ, 2011, pp. 432–436. doi:10.1109/ICCAIE.2011.6162174
- [23] Mutch, J., and Lowe, D. G., "Object Class Recognition and Localization Using Sparse Features with Limited Receptive Fields," *International Journal of Computer Vision*, Vol. 80, No. 1, 2008, pp. 45–57. doi:10.1007/s11263-007-0118-0
- [24] Mutch, J., and Lowe, D. G., "Multiclass Object Recognition with Sparse, Localized Features," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ, 2006, pp. 11–18. doi:10.1109/CVPR.2006.200
- [25] Boubezoul, A., and Paris, S., "Application of Global Optimization Methods to Model and Feature Selection," *Pattern Recognition*, Vol. 45, No. 10, Oct. 2012, pp. 3676–3686.
- doi:10.1016/j.patcog.2012.04.015
 [26] Huang, C. L., and Wang, C. J., "A GA-Based Feature Selection and Parameters Optimization for Support Vector Machines," *Expert Systems with Applications*, Vol. 31, No. 2, Aug. 2006, pp. 231–240.
 doi:10.1016/j.eswa.2005.09.024
- [27] Schneider, G., Wersing, H., Sendhoff, B., and Korner, E., "Evolutionary Optimization of a Hierarchical Object Recognition Model," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 35, No. 3, 2005, pp. 426–437. doi:10.1109/TSMCB.2005.846649
- [28] Simon, D., "Biogeography-Based Optimization," IEEE Transactions on Evolutionary Computation, Vol. 12, No. 6, 2008, pp. 702–713. doi:10.1109/TEVC.2008.919004
- [29] Simon, D., "A Dynamic System Model of Biogeography-Based Optimization," Applied Soft Computing, Vol. 11, No. 8, Dec. 2011, pp. 5652–5661. doi:10.1016/j.asoc.2011.03.028
- [30] Simon, D., Ergezer, M., Du, D., and Rarick, R., "Markov Models for Biogeography-Based Optimization," IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics, Vol. 41, No. 1, Feb. 2011, pp. 299–306. doi:10.1109/TSMCB.2010.2051149
- [31] Bhattacharya, A., and Chattopadhyay, P. K., "Biogeography-Based Optimization for Different Economic Load Dispatch Problems," *IEEE Transactions on Power Systems*, Vol. 25, No. 2, May 2010, pp. 1064–1077. doi:10.1109/TPWRS.2009.2034525
- [32] Gupta, S., Arora, A., Panchal, V. K., and Goel, S., "Extended Biogeography Based Optimization for Natural Terrain Feature Classification from Satellite Remote Sensing Images," *Contemporary Computing*, Vol. 168, 2011, pp. 262–269. doi:10.1007/978-3-642-22606-9_28
- [33] Lee, T., "Image Representation Using 2D Gabor Wavelets," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 10, 1996, pp. 959– 971.

- [34] Buhmann, M. D. A., "Radial Basis Functions: Theory and Implementations," Cambridge Univ. Press, Cambridge, 2003, pp. 99–146. doi:10.1017/CBO9780511543241
- [35] Lohokare, M., Panigrahi, B., Pattnaik, S., Devi, S., and Mohapatra, A., "Neighborhood Search-Driven Accelerated Biogeography-Based Optimization for Optimal Load Dispatch," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 42, No. 5, 2012, pp. 641–652. doi:10.1109/TSMCC.2012.2190401
- [36] Wang, X. H., and Duan, H. B., "Predator-Prey Biogeography-Based Optimization for Bio-Inspired Visual Attention," International Journal of Computational Intelligence Systems, Vol. 6, No. 6, 2013, pp. 1151–1162. doi:10.1080/18756891.2013.820957
- [37] Kennedy, J., and Eberhart, R. C., "A Discrete Binary Version of the Particle Swarm Algorithm," *IEEE International Conference on Computational Cybernetics and Simulation*, Vol. 5, IEEE, Piscataway, NJ, 1997, pp. 4104–4108. doi:10.1109/ICSMC.1997.637339
- [38] Lowe, D. G., "Object Recognition from Local Scale-Invariant Features," IEEE International Conference on Computer Vision, IEEE, Piscataway, NJ, 1999, pp. 1150–1157.
- doi:10.1109/ICCV.1999.790410
 [39] Bosch, A., Zisserman, A., and Munoz, X., "Representing Shape with a Spatial Pyramid Kernel," *Proceedings of the ACM International Conference on Image and Video Retrieval*, Assoc. for Computing Machinery, New York, 2007, pp. 401–408. doi:10.1145/1282280.1282340

doi:10.1109/34.541406

DUAN AND DENG

- [40] Lazebnik, S., Schmid, C., and Ponce, J., "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ, 2006, pp. 2169–2178. doi:10.1109/CVPR.2006.68
- [41] Wang, X. H., Duan, H. B., and Luo, D. L., "Cauchy Biogeography-Based Optimization Based on Lateral Inhibition for Image Matching," *Optik*, Vol. 124, No. 22, 2013, pp. 5447–5453. doi:10.1016/j.ijleo.2013.03.124
- [42] Duan, H. B., and Li, P., Bio-Inspired Computation in Unmanned Aerial Vehicles, Springer–Verlag, Berlin, 2014, pp. 215–245. doi:10.1007/978-3-642-41196-0

J. How Associate Editor