



A multi-objective optimization model and its evolution-based solutions for the fingertip localization problem

Dunwei Gong^{a,b}, Ke Liu^{a,*}^aSchool of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China^bSchool of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China

ARTICLE INFO

Article history:

Received 6 January 2017

Revised 31 August 2017

Accepted 2 September 2017

Available online 14 September 2017

Keywords:

Fingertip localization

Multi-objective optimization

Estimation of distribution algorithm

Probability distribution model

ABSTRACT

Exact fingertip positions are of particular importance to the fingertip-based human-computer interaction. We build a multi-objective optimization model for the problem of fingertip localization, and present a method to solve the above model based on evolutionary algorithms. When building the model, we take the positions of a series of pixels as the decision variable, the shape of the hand-edge curve corresponding to each of the pixels as one objective function, and the distance between each of the pixels and the gravity center of the palm as the other objective function. In addition, based on the correlation among the positions of pixels of the fingertip regions, we present a multi-objective estimation of distribution algorithm to solve the model so as to obtain the best pixel set, thus gaining the fingertip positions. The experimental results demonstrate the effectiveness of the proposed model and algorithm.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Human-computer interaction is an important way to accomplish complex tasks. During the human-computer interaction, making a computer understand the behavior of a human is of considerable importance. Among various behaviors, one based on a human's fingertips can express rich information. Therefore, exactly locating a human's fingertips is very important to the cooperation of a human and a computer to accomplish a given task.

We can gain a binary image of a hand by gesture segmentation [1,2]. Following this, fingertip localization is adopted to obtain the coordinates of each fingertip in the image. There have been several categories of methods in solving the problem, such as the fingertip localization based on contour curvatures [3–5] and the fingertip localization based on a hand skeleton [6,7], among many others. However, these methods generally have a low accuracy of locating fingertips, due to lack of the mechanism with which a human processes visual signals. The so-called accurate localization of fingertips is to obtain the coordinates of the center of each fingertip region.

A fingertip is the upper part of a finger, and covered by a fingernail. Compared with the other parts of a hand, each fingertip has a prominent position, and is easy to be identified by the shape of its edge, which can be called as the two features of each finger-

gertip region. If we select a number of pixels whose features are closest to the two features of fingertip regions from all the hand pixels, the coordinates of the center of each fingertip region can be calculated based on these selected pixels. Based on the above analysis, the problem of selecting the pixels of fingertip regions is essentially an optimization problem.

The distribution of each fingertip center has certain randomness in a hand region. But the edge of the fingertip region is approximately a semicircle arc, and the fingertip region is prominent. So we can formulate fingertip localization as a multi-objective optimization problem to select fingertip pixels according to the above two features of each fingertip region. Fingertip localization is to obtain the position of each fingertip center from a hand region. The fingertip center is in the fingertip region, so it is not accurate to obtain the fingertip center from the hand edge. If we formulate fingertip localization as a multi-objective optimization problem, we can obtain the fingertip center whose position may be in the fingertip region and similar to its real position.

In this paper, a multi-objective optimization model for fingertip localization is built, and a method of solving the above model is presented. When building the model, we take the positions of a series of pixels as the decision variable, the shape of the hand-edge curve corresponding to each of the pixels as one objective function, and the distance between each of the pixels and the gravity center of the palm as the other objective function. In addition, based on the correlation among the positions of pixels of the fingertip regions, we present a multi-objective estimation of distribution algorithm (EDA) to solve the model so as to obtain the best pixel set,

* Corresponding author.

E-mail addresses: dwgong@vip.163.com (D. Gong), [liu791018@126.com](mailto.liu791018@126.com) (K. Liu).

thus obtaining the fingertip positions. In this EDA, the probability distribution model and the sampling method are both aimed at the multi-objective optimization model for fingertip localization.

The main contributions of this paper are mainly reflected in the following three aspects: (1) A multi-objective optimization model for the problem of fingertip localization is built. (2) A multi-objective EDA based on the correlation among the positions of pixels of the fingertip regions is proposed, including a suitable probability distribution model and a new sampling method. (3) The effectiveness of the proposed model and algorithm is verified by a series of experiments. The work of this paper provides a feasible and efficient way to solve the fingertip localization problem.

The remainder of this paper is arranged as follows. [Section 2](#) reviews related work. A multi-objective optimization model for fingertip localization is built in [Section 3](#). [Section 4](#) proposes a multi-objective EDA based on the correlation among the positions of pixels of the fingertip regions. The applications of the proposed model and algorithm in actual problems of fingertip localization are provided in [Section 5](#). Finally, [Section 6](#) concludes this paper, and points out topics to be further studied.

2. Related work

In this paper, we study the problem of fingertip localization. After building a multi-objective optimization model for the above problem, we adopt a multi-objective evolutionary optimization method to solve the above model. In the following, we will give a detailed review of related research.

2.1. Fingertip localization

After segmenting a hand region, a system of human-computer interaction based on the positions of fingertips can find the coordinates of each fingertip. During processing a gesture image, the localization of fingertips is usually based on their shape. In [8], a hand model is utilized to locate fingertips, which is represented by a probability density function. This probability density function considers the location parameters, i.e., rotational angle and translation vector. So the probability that an observed feature configuration corresponds to the trained feature configuration can be obtained. In [9], an input image splits into R, G, and B planes. Edges are detected in each plane by Canny edge detection with initial Canny edge thresholds, and then detected edge regions are expanded in each plane by morphological gradient and smoothing. The pixel values of edge regions in each plane are compared, and the pixels which have the maximum pixel value among the three planes are selected as an edge. Then, the number of edge pixels is compared with the pixel number threshold to find a fingertip edge.

In recent years, a number of methods have been developed for fingertip localization. In [10], Wu et al. proposed an algorithm for finding the local maximum distance outside circles of extended centroid distance for fingertip localization. In the case of the circle whose radius is the average centroid distance, it is easy to determine the number of pixels on the circle. When calculating a great number of consecutive pixels, they can be sure that this region is a wrist and therefore remove the corresponding contour points. Because the circle is rotationally invariant, even if the hand rotates, the algorithm still works. In [11], Suau et al. addressed the problem of fingertip location by making use of the oriented radial distribution descriptor in a structured inference framework. Maxima of the oriented radial distribution of the input patch are likely to represent fingertip positions. In [12], Prasertsakul et al. presented a fingertip detection method that is based on the top-hat transform. The palm of the hand is obtained using the morphological opening. Fingertips are then obtained as the opening residue, the difference between the input image and the palm image. In [13], Wang et al.

extracted the gesture contour, calculated gesture gravity and used the Douglas-Peucker algorithm to approximate contour polygons. Then they detected the convexity defects of the approximated contour as the fingertip points of the candidate.

Additionally, Candela et al. [14] combined the method based on the curvatures of a hand contour with the method based on the skeleton of a hand, and located fingertips through the following two phases. They firstly seek the candidate positions based on the curvatures of a hand contour, and then obtain the accurate positions of fingertips from these candidate positions based on the skeleton of the hand. With the trained classifier, Jang et al. [15] initially detected the candidate points of the fingertips by selecting the maximum probability between true and false choices. Then they found the center position of each fingertip from clustering the candidate points into five groups of points. Heo et al. [16] located fingertip based on the relationship of position between fingertip and center of palm. The finger range is decided between the palm and hand range which are calculated through the analysis of the areas ratio to the polar transform image. The fingertips are detected by k-curvature method which is defined by tangential differentiation of a curve function within the finger range. The use of the curvature method for the detection of fingertips can easily lead to erroneous results, due to interference from false inflection points. Thus, Ho et al. [17] proposed a method for the selection of candidate points with less interference. The point with the greatest curvature among the candidate points is identified as the fingertip. The contours of the finger obtained via edge detection is first presented in the form of an universal set. The contour of the finger poses a problem in that the contour cannot be as loses smoothness due to digitization errors; there are, such that a number of inflection points may be misinterpreted as fingertips. The proposed candidate points method eliminates points that could cause interference, thereby ensuring that the points with the greatest curvatures are indeed the fingertips.

2.2. Multi-objective evolutionary optimization

In real-world applications, one often encounters problems with simultaneously optimizing multiple objectives under specific conditions. They are so-called multi-objective optimization problems. For a multi-objective optimization problem, its objectives often conflict with each other. That is, the improvement of one objective is at the cost of deteriorating one or more other objectives. Therefore, one can obtain a solution set that compromises all the objectives of the optimization problem. To tackle multi-objective optimization problems, scholars have combined evolutionary algorithms [18,19] with multi-objective optimization, developed a variety of methods, and formed a popular research topic, i.e., multi-objective evolutionary optimization [20,21].

Schaffer [22] proposed that evolutionary optimization methods can be employed to solve multi-objective optimization problems, which is the pioneering work of solving multi-objective optimization problems by using evolutionary algorithms. Later, Srinivas and Deb proposed non-dominated sorting genetic algorithm (NSGA) [23] for solving multi-objective optimization problems. Deb et al. improved NSGA, and proposed a famous algorithm, NSGA-II [24]. Since then, a variety of efficient multi-objective evolutionary algorithms have been proposed based on NSGA-II [25–27]. Additionally, scholars have proposed many other multi-objective evolutionary optimization algorithms [28,29].

Compared with genetic algorithms based on the micro mode in the search space [30–32], Estimation of distribution algorithm is based on the macro counterpart in the search space, and has a stronger capability in exploration and a more rapid convergence speed [33,34]. Zhang et al. proposed regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) [35],

and employed it to solve continuous multi-objective optimization problems. This method was developed based on the Karush–Kuhn–Tucker condition. That is, the Pareto-optimal set of a continuous multi-objective optimization problem forms a piece-wise continuous $(u - 1)$ -dimensional manifold in the decision space. Here, u is the number of objectives. At each generation, RM-MEDA builds a probability model by using the approach of the local principal component analysis, generates a number of candidates by sampling the model, and selects solutions with the number of the population size for the next generation by the non-dominated sorting in NSGA-II. Previous studies have shown that RM-MEDA is superior to NSGA-II when solving continuous multi-objective optimization problems with variable linkages.

Other research achievements of EDA have been obtained in recent years. Liang et al. [36] proposed a Boltzmann-based EDA for resource scheduling. This EDA is based on an approximation of the Boltzmann distribution. This approximation method is a tradeoff between solution accuracy and complexity. Reference [37] introduced a hybrid approach consisting of a variable neighborhood search and a new EDA to deal with the flow-shop scheduling problem. The new EDA of the introduced approach has the ability to discover promising regions in the search space. Wang et al. [38] proposed an EDA with stochastic local search to tackle the uncertain capacitated arc routing problem. In this method, a two phase stochastic local search procedure is integrated with an EDA to minimize the maximal total cost over a set of different scenarios. And the stochastic local search procedure avoids excessive fitness evaluations in local search. Wang and Wang [39] proposed an EDA-based memetic algorithm for solving the distributed assembly permutation flow-shop scheduling problem with the purpose of minimizing the maximal completion time. In this EDA, they proposed a novel selective-enhancing sampling mechanism for generating new solutions by sampling the probability model. The EDA and a local search are incorporated within the memetic algorithm framework. Wang et al. [40] presented a hybrid Pareto-archived EDA to solve the mode-identity resource-constrained project scheduling problem with makespan and resource investment criteria. In this EDA, they used a Pareto archive to preserve the non-dominated solutions, and they used another archive to preserve the solutions for updating the probability model. Besides, they provided specific updating mechanism and sampling mechanism for the probability model to track the most promising search area.

3. The proposed multi-objective optimization model for fingertip localization

The position of each fingertip region of a human hand is prominent, and is easy to be identified by the shape of its edge. In a system of human-computer interaction based on the positions of fingertips, the position of each fingertip is regarded as a point [15]. According to the above characteristics of fingertips in a binary image of a hand, we can select a number of pixels in or near each fingertip region, and then obtain the position of a fingertip by calculating the average of these positions.

In order to select a series of pixels located in the regions of fingertips from a binary image of a hand, we build the following multi-objective optimization model:

$$\min \vec{F}(x) = (f_1(x), f_2(x)) \quad \text{s.t. } x \in 2^X \quad (1)$$

where X is the set of positions of all the pixels in a gesture image, 2^X represents the power set of X , x refers to a number of positions of the selected pixels, with each being different from the others, and denoted as $x = (x^1, x^2, \dots, x^n)$, $x^i = (x_i, y_i)$, $i = 1, 2, \dots, n$. Here, n is the number of the selected pixels.

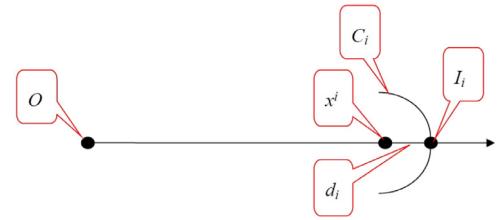


Fig. 1. Illustration of the formulation of $f_1(x)$.

In the following, we give the formulation of $f_1(x)$ and $f_2(x)$. In the binary image of a hand, the edge of each fingertip is approximately a semicircle arc, and can be easily identified. As a result, when selecting pixels of a hand region, we first obtain a half-line which starts from the gravity center of the palm, O , and points to each selected pixel, x^i , of the hand region. Following that, we get the position of the intersection point, I_i , between the above half-line and the curve of the hand edge, and calculate the distance, d_i , between I_i and x^i . Then, we seek a curve segment, C_i , at the hand edge, with its center being I_i , and its length being πd_i . Additionally, let $m = \pi d_i$, and we select m pixels, i.e., $x_1^i, x_2^i, \dots, x_m^i$, at the hand edge nearest to I_i , which are evenly located on both sides of I_i . Finally, the distance between x^i and each of $x_1^i, x_2^i, \dots, x_m^i$ is calculated, and denoted as $l_1^i, l_2^i, \dots, l_m^i$. Illustration of the above formulation is shown in Fig. 1.

Based on $l_1^i, l_2^i, \dots, l_m^i$, we can obtain their average, denoted as \bar{l}^i . So \bar{l}^i can be represented as $\bar{l}^i = \frac{1}{m} \sum_{j=1}^m l_j^i$. Further, we can get $|l_j^i - \bar{l}^i|$ which is the difference between l_j^i and \bar{l}^i . If we denote the sum of $|l_j^i - \bar{l}^i|$ for $j = 1, 2, \dots, m$ as $Z_1(x^i)$, $Z_1(x^i)$ can be represented as $Z_1(x^i) = \sum_{j=1}^m |l_j^i - \bar{l}^i|$. We expect that C_i is approximately a semicircle arc, so the corresponding $l_1^i, l_2^i, \dots, l_m^i$ should approximately be equal to each other. As a result, the difference among $l_1^i, l_2^i, \dots, l_m^i$ should be minimal, which is equivalently to minimizing $Z_1(x^i)$. We aim to produce n pixels located at the fingertip regions. For all these n pixels, we should minimize the sum of $Z_1(x^1), Z_1(x^2), \dots, Z_1(x^n)$. Denote the above sum as $f_1(x)$, then $f_1(x)$ has the following expression:

$$f_1(x) = \sum_{i=1}^n |Z_1(x^i)| \quad (2)$$

Given the fact that a fingertip is prominent, we expect that the distance between each of these n pixels and O maximal. We denote $Z_2(x^i) = |x^i - O|$ as the distance between x^i and O , then $Z_2(x^i)$ should be maximal. For all these n pixels, we should maximize the sum of $Z_2(x^1), Z_2(x^2), \dots, Z_2(x^n)$, or equivalently minimize the reciprocal of the sum. Denote $f_2(x)$ as the reciprocal, then the expression of $f_2(x)$ can be represented as follows:

$$f_2(x) = \frac{1}{\sum_{i=1}^n |Z_2(x^i)|} \quad (3)$$

In the objective function, $f_1(x)$ or $f_2(x)$, the decision variable, x , is a $2n$ -dimensional vector. The vector includes $2n$ coordinates of n pixels, i.e., n horizontal coordinates and n vertical coordinates. Among the n pixels of the decision variable, one pixel is different from the others, and all the pixels are from the hand region. So the decision variable is a pixel set, including n pixels in the hand region. We define these objective functions for each pixel set.

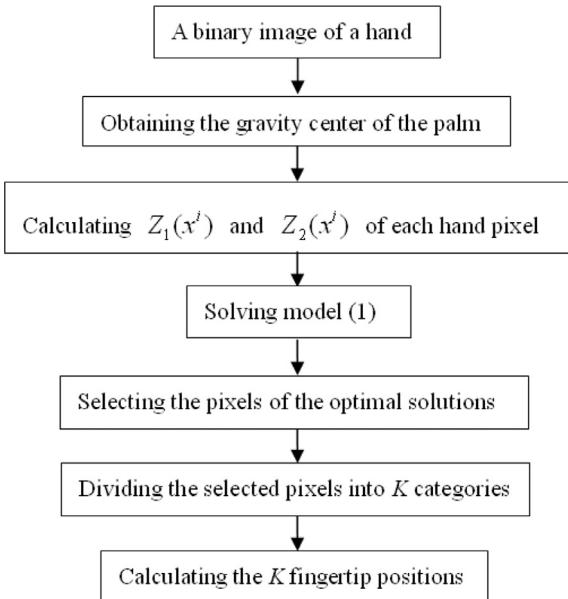


Fig. 2. The overall process.

The unique way to evaluate the goodness of a pixel set in a fingertip region or in the palm region is to employ the objective functions of the optimization problem. That is to say, a pixel set in the fingertip region has the same objective function as that in the palm region. They have, however, different values of the objective functions, and a pixel set in the fingertip region generally has smaller values than that in the palm region. As a result, a pixel set in the fingertip region is better than that in the palm region, and has more opportunities to survive than the latter during solving the optimization problem.

By solving model (1), we can obtain a series of pixels. If K fingertips of a hand stretch out, we should further classify the obtained pixels into K categories according to their coordinates. Here we use the K -means algorithm [41] to classify the pixels, and $K = 1, 2, \dots, 5$. We obtain the center of each category, so as to obtain the position of each fingertip.

The overall process of dealing with a binary image of a hand and obtaining its fingertip positions is described using a flowchart, shown in Fig. 2.

In the next section, we will present a novel algorithm for solving model (1), called multi-objective estimation of distribution algorithm.

4. An estimation of distribution algorithm for fingertip localization

In this section, we use EDA to solve model (1) and obtain a series of pixels. We first generate an initial population randomly. Following that we conduct the following iterations. First, we build the probability distribution model of candidate components based on the optimal solution set. Second, we obtain a temporary population by sampling the probability distribution model. Finally, we obtain an offspring population based on the combined population consisting of the father and the temporary populations by fast non-dominated sorting [24]. We repeat the above process until a termination condition of the algorithm is reached. In this way, we can obtain one or more non-dominated solutions.

4.1. Clustering of fingertip pixels

As we know, each of the fingertip regions is formed by a series of pixels, and the distribution of these pixels is not completely random. Based on this, we divide the pixels of forming K fingertip regions into K categories according to the fingertip positions, with each category corresponding to one fingertip. Here K fingertips of a hand stretch out, so K is 1, 2, ..., or 5.

The candidate solution x of model (1) contains n pixels, and they are different from each other. If x is the optimal solution of model (1), K fingertip regions can be formed based on x . The pixels forming a fingertip region correspond to one fingertip.

In a binary image of a hand, a fingertip position is a point. Therefore, the pixels located in or near the fingertip position form an approximate circular region, and the approximate circular region takes the fingertip position as the center of the circular region. Since K fingers stretch out, we can assume that the pixels of the fingertip regions are located around K points, $L_k, L_k = (x_k^L, y_k^L), k = 1, 2, \dots, K$. So all the pixels of the fingertip regions can be divided into K sets, A_k , and the pixels of each set are distributed around point L_k .

We use the K -means algorithm [42,43] to cluster the pixels of x into K categories. The steps are as follows:

Step 1: Randomly select K pixels, $x^{i_1}, x^{i_2}, \dots, x^{i_K}$, $i_k \in \{1, 2, \dots, n\}$, from the pixels of x as the centers of these clusters, $L_k, k = 1, 2, \dots, K$.

Step 2: Based on the distance between each pixel of x and L_k , cluster the pixels of x into K categories, A_k , with A_k having the following expression

$$A_k = \{x^i | x^i \in x, D(x^i, L_k) \leq D(x^i, L_{k'})\},$$

$$\forall k' \neq k, k' \in \{1, 2, \dots, K\},$$

where $D(x^i, L_k)$ represents the distance between pixel x^i and L_k .

Step 3: Obtain the average coordinates of all the pixels of A_k and take them as a updated center of the cluster, L_k . That is

$$L_k = \frac{1}{|A_k|} \sum_{x^i \in A_k} x^i.$$

Step 4: Based on each updated center, repeat Steps 2 and 3, until all these categories do not change.

4.2. The probability distribution model of candidate components

We use the non-dominated sorting method [24] to select one or more optimal solutions. We assume that all the pixels of each optimal solution are located in the fingertip regions. Thus all the non-coincident pixels of the optimal solution set are also located in the fingertip regions, and they can be divided into K categories, A_1, A_2, \dots, A_K , by the K -means algorithm. Pixel x^i of A_k is located near L_k and in the fingertip regions. So L_1, L_2, \dots, L_K are the probability distribution model of candidate components of model (1). We can obtain the pixels of the fingertip regions by sampling L_1, L_2, \dots, L_K .

4.3. Sampling

Since each fingertip position is a point, the number of pixels in or near each fingertip position is similar to each other, and the degree that the pixels in or near each fingertip position deviate from their center is also similar to each other. As a result, the distribution of the pixels obtained by sampling the above probability distribution model should accord with that of the pixels located in or

near each fingertip position, i.e., the number of the pixels obtained from each L_k is similar to each other, and the degree of this obtained pixels deviating from their center L_k is also similar to each other.

We can obtain a candidate solution by sampling the above probability distribution model in the decision space. We hope that the obtained candidate solution should be reasonable. It means that the i th component, $x^i (i = 1, 2, \dots, n)$, of the candidate solution should meet the following three conditions: (1) The probability of generating x^i from L_k is equal to each other; (2) x^i is located around L_k ; (3) the degree of x^i deviating from L_k is similar to each other. In addition, each pixel of the candidate solution of model (1) is different from the others.

We set $x = \phi$, and the probability of generating x^i from L_k as $p_k = 1/K$, $k = 1, 2, \dots, K$. Let H be the set of all the pixels in a hand region, and $t' = 0$. In order to generate candidate solution x to meet the above conditions, the following steps are adopted.

Step 1: According to the probability of p_k , select $L_k, L_k = (x_k^L, y_k^L)$ from the K points, L_1, L_2, \dots, L_K .

Step 2: Generate a new coordinate x'_i by Gaussian sampling around x_k^L with the sampling variance of λ . Generate a new coordinate y'_i by Gaussian sampling around y_k^L with the sampling variance of λ . The value of λ decreases linearly from the maximum, λ_{\max} , to the minimum, λ_{\min} , when the number of iterations of the proposed EDA increases from 1 to its maximum. λ_{\max} and λ_{\min} are both given in advance according to the size of a hand image. If a component of $x^i = (x'_i, y'_i)$ exceeds its boundary value, let the component be the boundary value. Set $t' = t' + 1$.

Step 3: If $x^i \in H$, set $x = x \cup \{x^i\}$ and $H = H - \{x^i\}$, and go to Step 6.

Step 4: If $t' \leq T'$, go to Step 6.

Step 5: Select a hand pixel, x^i , which is from H and around (x'_i, y'_i) . Set $x = x \cup \{x^i\}$ and $H = H - \{x^i\}$.

Step 6: If $|x| = n$, end the algorithm; otherwise, go to Step 1.

The above steps are the process of obtaining a candidate solution by sampling. We may not obtain a candidate with n different pixels by Gaussian sampling T' times, where T' is a threshold set in advance. When the number of iterations is larger than T' , we obtain the other hand pixels of the candidate using another sampling method. This new sampling method obtains one hand pixel different from the sampled ones in each iteration, and ends after we have obtained all the other hand pixels of the candidate. The worst case is that we cannot obtain any hand pixel by Gaussian sampling. On this circumstance, the new sampling method will run n times. As a result, the maximal number of iterations is $(T' + n)$ when obtaining all the pixels of a candidate. Let $\delta = 1$. The hand pixel, x^i , which is from H and around (x'_i, y'_i) in Step 5 is obtained in the following way.

Step 1': Set $G = H \cap \{(x_i, y_i) | x'_i - \delta \leq x_i \leq x'_i + \delta \text{ and } y'_i - \delta \leq y_i \leq y'_i + \delta\}$.

Step 2': If $G = \phi$, set $\delta = \delta + 1$, and go to Step 1'.

Step 3': Randomly select a hand pixel, x^i , from G , and end the algorithm.

We perform the above process N times, and obtain N candidate solutions. Thus we obtain a temporary population, Q , with its size being N .

4.4. Selection

The evolutionary population of the t th generation is $P(t)$, and the temporary population obtained by sampling the probability

distribution model of $P(t)$ is $Q(t)$. We generate the offspring population, i.e., $P(t+1)$ from $P(t) \cup Q(t)$ by the fast non-dominated sorting selection [24]. Please refer to [24] to view details of the fast non-dominated sorting selection.

A pixel in a hand region is located either in the fingertip regions or in the other regions. Candidates obtained by the proposed EDA generally approximate to the optimal ones of the optimization problem if they are not the optimal ones. On this circumstance, a few pixels of a candidate may be out of the fingertip regions, although they are close to the fingertip regions. For the other most pixels, they are located in the fingertip regions, and near the fingertip centers. We divide all the obtained pixels into K categories, and take the average of all the pixels in a category as the estimated center of a fingertip. As a result, the estimated center much approximates to one of real centers, which guarantees that the estimated center belongs to the fingertip regions.

A palm edge generally less approximates to a semicircle arc than that of a fingertip region. As a result, a fingertip pixel often has a smaller $Z_1(x^i)$ value than a palm pixel. Besides, compared with a fingertip pixel, a palm pixel is closer to the gravity center of the palm, resulting in a larger $Z_2(x^i)$ value of the fingertip pixel than the palm pixel. As a result, a set with more fingertip pixels and less palm pixels will have smaller values of $f_1(x)$ and $f_2(x)$ than that with less fingertip pixels and more palm pixels. What to classify is a set with a number of pixels instead of a pixel. We can clearly classify sets through the values of $f_1(x)$ and $f_2(x)$.

The algorithm of solving the multi-objective optimization model includes the following three parts: building the probability distribution model of candidate components, sampling candidates, and sorting them. The time complexity of building the probability distribution model is $O(KnN)$, according to that of the K -means algorithm [41]. The time complexity of sampling candidates is $O(nN)$, based on the proposed method. According to [23], the time complexity of sorting candidates using the fast non-dominated sorting is $O(2N^2)$. As a result, the overall time complexity of the proposed algorithm is governed by the selection part, and the time complexity of the multi-objective optimization algorithm is $O(2N^2)$.

5. Experiments

In this section, the effectiveness of the proposed model and algorithm is verified by experiments. First, the problems to be verified are proposed. Second, the hand images used in the experiments are introduced. And then, two compared methods are given. After the experiment process is presented, the experimental results are given and analyzed.

5.1. Problems to be verified

The following questions should be answered to illustrate the effectiveness of the model and algorithm proposed in this paper.

- (1) Can the proposed method accurately locate the fingertips? In binary images of hands, we investigate whether the proposed method can seek the fingertip positions or not.
- (2) Can the proposed method excel the existing methods? We compare the proposed EDA with a famous evolutionary algorithm, and compare the proposed method with an advanced method of fingertip localization. Then we evaluate the superiority of the proposed method by the errors of calculation results of the three methods.

5.2. Benchmark gesture images

In this paper, we use several gesture images of the standard American Sign Language image database [44] to carry out experiments. In the image database, there are 630 gesture images, and

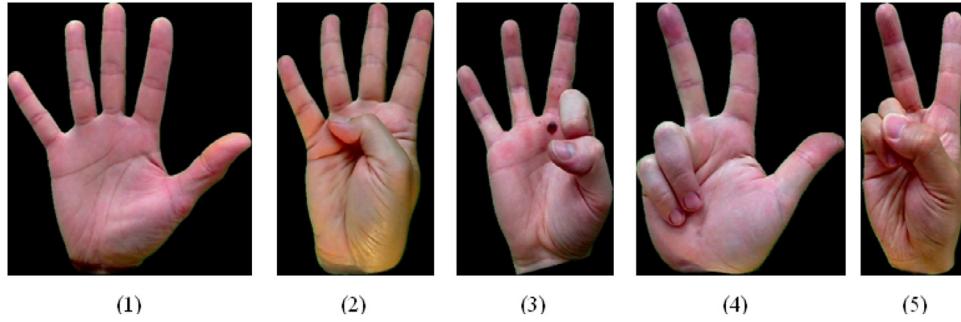


Fig. 3. The gesture images.

each gesture image among them expresses a figure among nine figures, i.e., 1, 2, ..., or 9. We select these gesture images to experiment. We sort the selected gesture images by their file names, and use their serial numbers to rename them. So their file names are changed from hand1-1-bot-seg-1-cropped.png, hand1-1-bot-seg-2-cropped.png, ..., hand5-9-dif-seg-5-cropped.png to 1.png, 2.png, ..., 630.png.

Image 121.png, 301.png, 491.png, 506.png, 551.png are from 5 people respectively, shown in Fig. 3 (1)–(5) in order. The hand regions of the 630 images have been segmented, and their backgrounds are black regions. They are all the binary images of hands.

5.3. The compared methods

We compare the proposed EDA with a famous evolutionary algorithm, and compare the proposed method with an existing method of fingertip localization.

The famous evolutionary algorithm, NSGA-II, utilizes the crossover and mutation operators to obtain a temporary population, Q . The value of the decision component x_i or y_i of model (1) changes continuously. Thus, the crossover and mutation operators of NSGA-II can be employed to solve model (1). In addition, both NSGA-II and the proposed EDA utilize the fast non-dominated sorting selection to obtain the offspring population, $P(t+1)$, from $P(t)$ and $Q(t)$. The method of generating the temporary population, $Q(t)$, of NSGA-II is different from that of the proposed EDA.

Since the computational complexity of an evolutionary algorithm is mainly determined by the evaluations of individuals, which is equal to the population size multiple to the number of generations, we should guarantee that NSGA-II and the proposed EDA have the same evaluations of individuals when comparing. To fulfill this task, we set the population sizes of the two algorithms to be same, and set their maximal numbers of generations to be same. In this way, the comparison of the two algorithms is fair. We use NSGA-II to solve model (1) of the 630 gesture images. By solving model (1), we can obtain a series of pixels. If K fingertips of a hand stretch out, we use the K -means algorithm [41] to classify the obtained pixels into K categories, so as to obtain the position of each fingertip.

The existing method of Candela et al. [14] locate the fingertips by two stages. The first stage is to seek the candidate fingertip positions, and the second stage is to eliminate the wrong positions from the candidate fingertip positions.

First, all the pixels on the hand boundary are sorted by the adjacent sequence. So each pixel on the hand boundary has a sequence value. The sequence value of boundary pixel P_v is v , with its curvature K_w^v being

$$K_w^v = \frac{\overrightarrow{P_v P_{v-w}} \cdot \overrightarrow{P_v P_{v+w}}}{\left| \overrightarrow{P_v P_{v-w}} \right| \left| \overrightarrow{P_v P_{v+w}} \right|}$$

We compare K_w^v with a given threshold, K_{th} . If $K_w^v \geq K_{th}$, we select the boundary pixel. So we obtain the selected boundary pixels (SBPs). If there is a set of SBPs whose sequence values are adjacent to each other, we will keep the SBP whose sequence value is the nearest to the central sequence value of the set, and eliminate the other pixels of the set from the SBPs. Then, if the SBPs can form a convex polygon, we will keep them as the candidate fingertip positions; otherwise we eliminate the wrong positions from them. In this way, the candidate fingertip positions are obtained.

In order to obtain the correct fingertip positions from the candidate fingertip positions, the skeleton of the hand region is first obtained by distance transformation; then, the skeleton lines of the finger regions are obtained by Hough transformation [14]. If a candidate fingertip position is close to the skeleton lines of the finger regions, it will be a correct fingertip position.

We use this existing method to locate the fingertips of the 630 gesture images, and take $K_{th} = 0.5$, $w = 50$.

5.4. The experiment process

In the proposed method, we obtain first $Z_1(x^i)$ and $Z_2(x^i)$ of x^i of each image. Then, we build model (1) of the image. Following that, we use EDA to solve model (1) and obtain the optimal solution set. Finally, we obtain the fingertip positions by the optimal solution set.

The decision space of model (1) is all the pixels of the hand region. If the size of the decision space is large, the complexity of solving model (1) will be increased. Therefore, it is necessary to compress the above images. By sampling the pixels at each interval of 4 rows and 4 columns in each above image, we reduce the size of each image.

Additionally, we take the decision space as all non-boundary pixels of the hand region. We use the method of [45] to obtain the gravity center of the palm, O , in each compressed image. According to the distance between non-boundary pixel, x^i , and O , we obtain $Z_2(x^i)$ of x^i .

We use a relatively simple method to obtain the approximate position of I_i to reduce the complexity of calculating $Z_1(x^i)$. We first obtain the connection line, L_v , between hand-boundary pixel, P_v , and O . Then, we calculate the distance, d_v^i , between non-boundary pixel x^i and L_v . Finally, we select L_v with the minimal distance, and take its corresponding P_v as the approximate position of I_i . In this way, we obtain the approximate position of I_i .

According to the approximate position of I_i of x^i , we calculate its \bar{l}_i and m . If $m < 20$, we take $m = 20$. And then we obtain its $Z_1(x^i)$. Thus, we can obtain $Z_1(x^i)$ and $Z_2(x^i)$ of each non-boundary pixel of each compressed image.

We further reduce the size of the decision space of model (1) by the following method. According to each value of $Z_2(x^i)$, we take half of the maximum value of $Z_2(x^i)$ as the threshold, i.e., $Z_{th} = \max(Z_2(x^i))/2$. If $Z_2(x^i) \geq Z_{th}$, we take its x^i as a pixel of the deci-

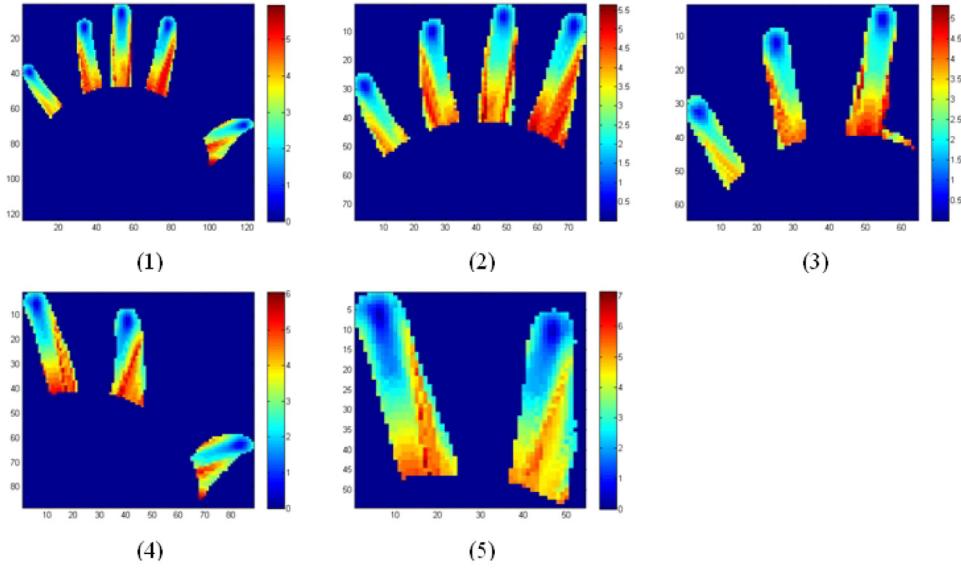


Fig. 4a. Distribution of $Z_1(x^i)$ values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sion space; otherwise, we remove its x^i from the decision space. So we remove a large number of pixels of the palm region from the decision space.

We compress each of the 630 images, calculate $Z_1(x^i)$ and $Z_2(x^i)$ of each non-boundary pixel of the hand region, and reduce the size of the decision space by the above method. In this way, we accomplish the preprocessing of each image.

According to model (1), the decision variable should represent several pixels. Thus, a vector of 60 dimensions is used as the decision variable in the following experiments. And decision variable x represents 30 pixels. We can obtain $Z_1(x^i)$ and $Z_2(x^i)$ of x , and obtain $f_1(x)$ and $f_2(x)$.

We use the proposed algorithm to solve model (1) of each preprocessed image, and then obtain one or more optimal solutions. Considering the efficiency of evolutionary computation, in each calculation of solving model (1), the population size is 30, and the number of evolutionary generations is 100. In this EDA, $\lambda_{\max} = 15$, $\lambda_{\min} = 4$. Due to randomness of the optimal solution set, we run 20 times evolutionary calculations when solving model (1). So we obtain 20 optimal solution sets and the selected pixels of the 20 optimal solution sets.

At last, we use the K -means algorithm [41] to divide the selected pixels of each preprocessed image into K categories, calculate the mean coordinates of all the pixels of each category, and obtain the K fingertip positions of the image.

The environment configuration in the experiments is Intel Core i3-3240 (3.40 GHz CPU, 1.85 GB RAM).

5.5. Experimental results and analysis

In the experiments of the proposed method, we preprocess the binary images of the hands, 1.png, 2.png, ..., 630.png, in order, and obtain $Z_1(x^i)$ and $Z_2(x^i)$. After preprocessing each image shown in Fig. 3 (1)–(5) in order, we obtain the binary image of the part of its hand region, i.e., Fig. 4a (1)–(5). $Z_1(x^i)$ and $Z_2(x^i)$ values of each image among Fig. 4a (1)–(5) are shown in Fig. 4a (1)–(5) and Fig. 4a (1)–(5) in order.

In each image of Fig. 4a (1)–(5), each vertical coordinate represents the number of rows of a pixel in its matrix image; each horizontal coordinate means the number of columns of a pixel in its matrix image. They are similar to each vertical or horizontal coordinate in each image of Fig. 4b (1)–(5). There is a color bar in

the right of each image of Fig. 4a (1)–(5). The color bar refers to the $Z_1(x^i)$ value of each pixel in each image. The color in the bar changes from dark red to dark blue when its $Z_1(x^i)$ value changes from large to small. They are similar to the color bar in each image of Fig. 4b (1)–(5), except that the color bar represents the $Z_2(x^i)$ values.

We run the proposed algorithm 20 times for each of the 630 preprocessed images, and obtain its 20 optimal-pixel sets. The 20 optimal-pixel sets of Fig. 4a (1) are shown in Fig. 1a of Appendix A. In Fig. 1a, the figures appearing from left to right in order in the first row are the results obtained from the first to the 4th run; the figures appearing from left to right in order in the second row are the results obtained from the 5th to the 8th run; ..., and the like. We show their populations at the last generation in the objective space in Fig. 1b. In Fig. 1b, the figures appearing from left to right in order in the first row are the results obtained from the first to the 4th run; the figures appearing from left to right in order in the second row are the results obtained from the 5th to the 8th run; ..., and the like.

By using the proposed method of this paper, we can also obtain the pixel sets of Fig. 4a (2)–(5) and their populations at the last generation in the objective space. They are shown in Figs. 2a, 3a, 4a, 5a, and 2b, 3b, 4b, 5b.

In each image shown in Fig. 1a, we use the 5-means algorithm to divide the selected pixels into 5 categories in order, calculate the mean coordinates of all the pixels of each category, and obtain the 5 fingertip positions of the hand. Similarly we obtain the K fingertip positions in each image of each of Figs. 2a, 3a, 4a, 5a. In Figs. 2a, 3a, 4a, 5a, $K = 4, 3, 3, 2$ in order.

A fingertip is a region covered by a fingernail in a finger, and the ratio of the length to the width of the fingernail is about 1. Thus the ratio of the length to the width of the fingertip region is about 1. We can manually obtain the center position of each fingertip region by the relationship between its length and width [17]. The steps are as follows. First, we take the average width of each finger as the width of the fingertip region; second, we segment the fingertip region from the top of the finger by the relationship between its length and width; finally, we calculate the average coordinates of the pixels of each fingertip region. The average coordinates can be taken as the real fingertip position.

The distance between each fingertip position obtained by each image of Fig. 1a and its corresponding real fingertip position is the

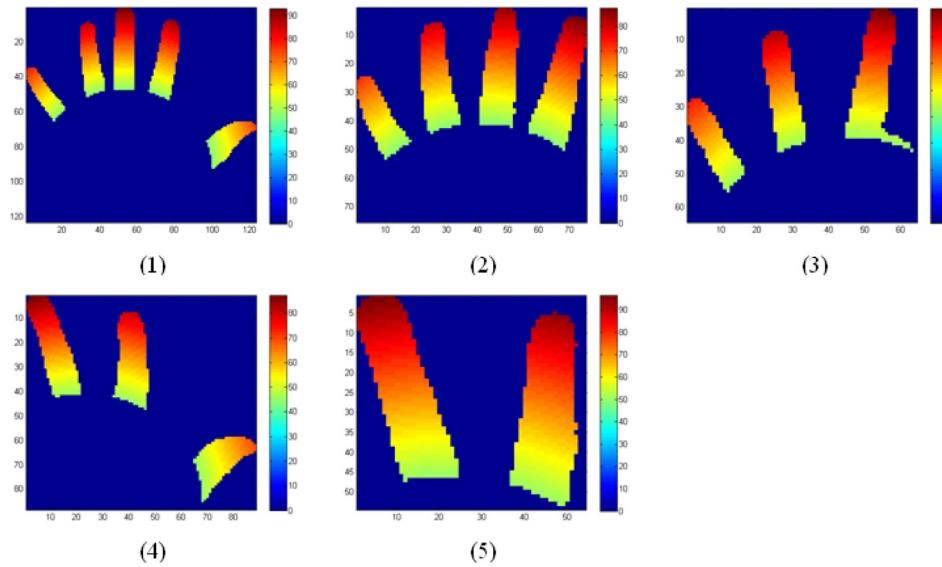


Fig. 4b. Distribution of $Z_2(x^i)$ values.

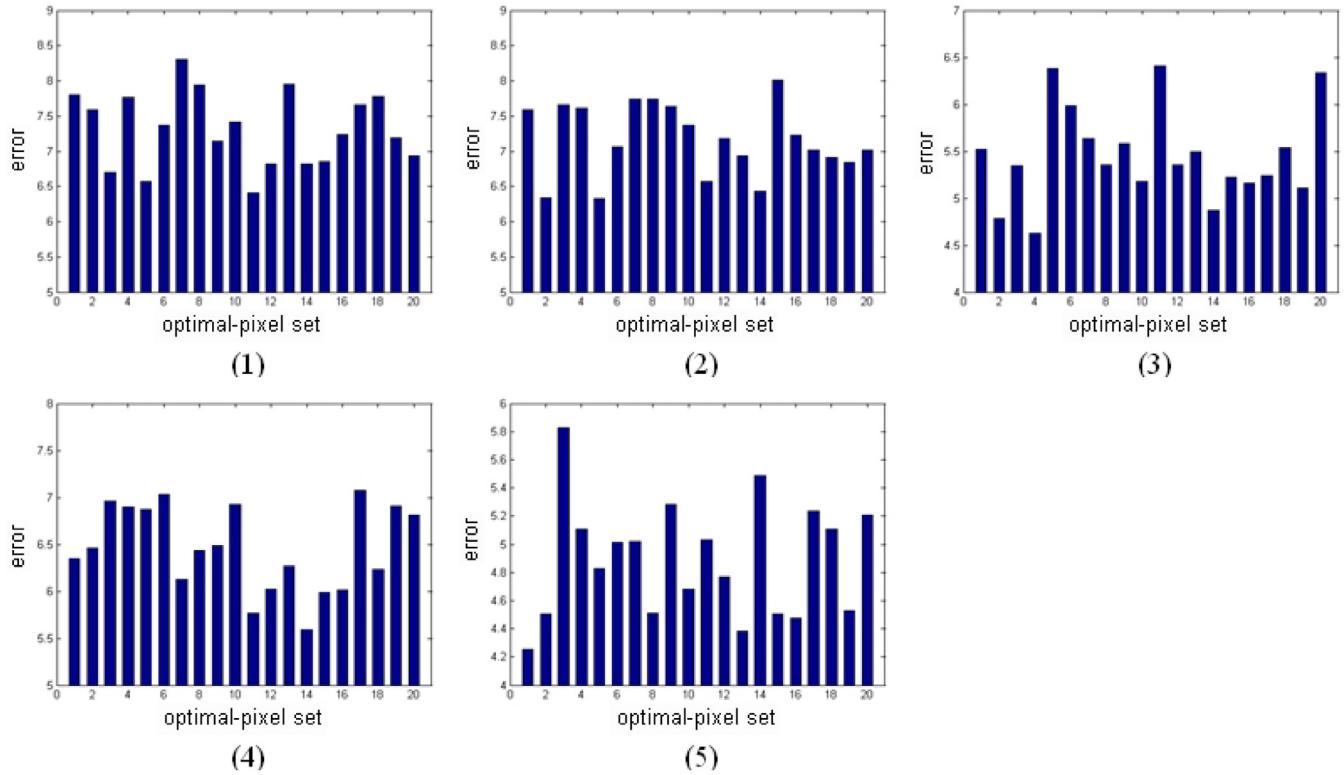


Fig. 5. The errors of Fig. 2a, 3a, 4a, 5a.

result error of the proposed method. Each image of Fig. 1a has 5 fingertip positions and 5 errors, so all 20 images have 100 errors. Fig. 1a shows 20 optimal-pixel sets, and the average value of the 5 errors of each above optimal-pixel set is shown in Fig. 5 (1) in turn. The horizontal coordinate of Fig. 5 (1) is the serial number of each optimal-pixel set of Fig. 1a. Similarly the average error of each optimal-pixel set of Figs. 2a, 3a, 4a, 5a is shown in Fig. 5 (2)–(5) in turn. The average value of the 100 errors of Fig. 1a is 7.45. Similarly the average error of each of Figs. 2a, 3a, 4a, 5a is 7.12, 5.47, 6.38, 5.04 in order, as shown in Table 1. The 630 average errors of the proposed method are shown in Fig. 6 (1), and their average value

Table 1
Average errors of Fig. 2a, 3a, 4a, 5a.

| 20 optimal-pixel sets | Average error |
|-----------------------|---------------|
| Fig. 1a | 7.45 |
| Fig. 2a | 7.12 |
| Fig. 3a | 5.47 |
| Fig. 4a | 6.38 |
| Fig. 5a | 5.04 |

is 6.25. This average value is small to the size of each fingertip region. Thus it indicates that the proposed model and algorithm

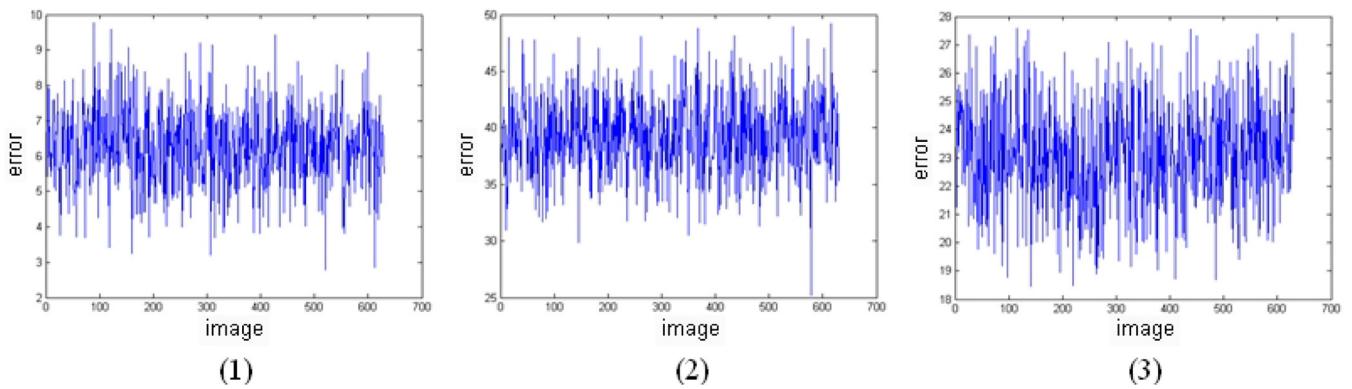


Fig. 6. The errors of the three methods.

Table 2
Average errors of Fig. 6 (1)–(3).

| Method | Average error |
|------------|---------------|
| Fig. 6 (1) | 6.25 |
| Fig. 6 (2) | 39.31 |
| Fig. 6 (3) | 23.12 |

can accurately seek the fingertip positions from a gesture image. In addition, the proposed method spends 4–5 s in accomplishing an experiment of fingertip localization of one image.

We apply NSGA-II to each of the 630 preprocessed images 20 times, and obtain 20 optimal pixel sets. For NSGA-II, the same parameter settings are adopted, that is, a vector of 60 dimensions is used as the decision variable, the population size is 30, and the maximal number of generations is 100. In addition, the same environment configuration is adopted. Based on these, we obtain the average error of 20 runs. In NSGA-II, the crossover and mutation probabilities are 0.9 and 0.1, respectively. Also, both the crossover and the mutation distribution indexes are 0.2. The 630 average errors of NSGA-II are shown in Fig. 6 (2), and their average value is 39.31.

By the method of [14], we locate the fingertips of image 1.png, 2.png,..., 630.png in order, and then obtain the fingertip positions and their average errors. The 630 average errors of the method of Candela et al. [14] are shown in Fig. 6 (3), and their average value is 23.12. The average error of each of Fig. 6 (1)–(3) is shown in Table 2 in turn.

The average error of the proposed EDA is lower than that of NSGA-II. This shows that the proposed EDA is superior to NSGA-II. The proposed EDA causes pixels of the temporary population to distribute around K points, where K is the number of fingertips. Thus, the proposed EDA is in accord with the correlation among the pixels of fingertip regions. NSGA-II does not take advantage of the correlation among the pixels of fingertip regions to generate a temporary population. Thus, the proposed EDA exceeds NSGA-II.

The average error of the proposed method is lower than that of the method of Candela et al. [14]. This shows that the proposed method is superior to the method of Candela et al. [14]. The proposed method selects the pixels of fingertip regions by the prominent position and the edge shape of each fingertip region, and then determines the fingertip positions in the hand region. But the method of Candela et al. [14] locates the fingertips on the edge curve of the hand region. Thus, the errors of the proposed method are lower than that of the existing method.

By the above experimental results and analysis, we can obtain the following conclusions: the proposed method can find the fingertip positions and excels the existing methods. So the proposed model and algorithm are effective.

5.6. Applications and limitations

The proposed method has numerous potential applications. Firstly, we can design a finger writing system where the movement locus of a fingertip is recognized as a letter or word. Secondly, we can design a virtual keyboard using the proposed method where a key will be pressed if a fingertip stays at the keys position for a certain time. Thirdly, we can develop a virtual reality environment where we catch, move or release an object according to the fingertip positions and their movements. Lastly, we can control a machine based on the proposed method, where we handle the switches and the buttons of the machine by the fingertip positions.

The proposed method has the following challenges and limitations. Firstly, the number of categories, K , in the proposed EDA is required to know before solving model (1), indicating that the number of fingertips is required to know, which is hard if it is not impossible. Secondly, due to the time complexity of evolutionary algorithms, the proposed algorithm spends a long time in locating the fingertips. As a result, obtaining a relatively accurate candidate of the optimization problem is at the cost of a high time complexity, suggesting that the proposed algorithm has a limited application for real time scenarios. Thirdly, due to the randomness of evolutionary algorithms, the proposed algorithm might obtain candidates with inferior quality, which can be reflected by the cases that a candidate might contain a number of pixels which are located at the non-fingertip regions. As a result, the proposed algorithm is generally unsuitable for the cases that have a high requirement in precision.

The hand of each image in the image database is from one of 5 people, and its image is shot from one of 5 viewpoints. In each image, 1, 2, ..., or 5 fingertips of a hand stretch out. Each of the 5 fingertips of a hand stretches out alone, or with the other fingertips. So the image database has a sufficient complexity to demonstrate the effectiveness of the proposed method.

The proposed method can behave as expected for real time situation if the following conditions are met. First, the number of fingertip positions is given before fingertip localization. Second, the time of accomplishing a process of fingertip localization of one image is allowed to be greater than 5 s.

6. Conclusions

A multi-objective optimization model of fingertip localization and its solution based on EDA are proposed. In the model, we take a set of pixels as the decision variable, the shape of the hand-edge curve corresponding to each of the pixels as one objective function, and the distance between each of the pixels and the gravity center of the palm as the other objective function. In the solution, we present a multi-objective EDA based on the correlation among the

pixels of the fingertip regions. We solve the model to get the best pixel set and obtain the fingertip positions. The proposed method is applied to the actual problem of fingertip localization of 630 images, and compared with two existing methods. The experimental results and analysis show the effectiveness of the proposed model and algorithm.

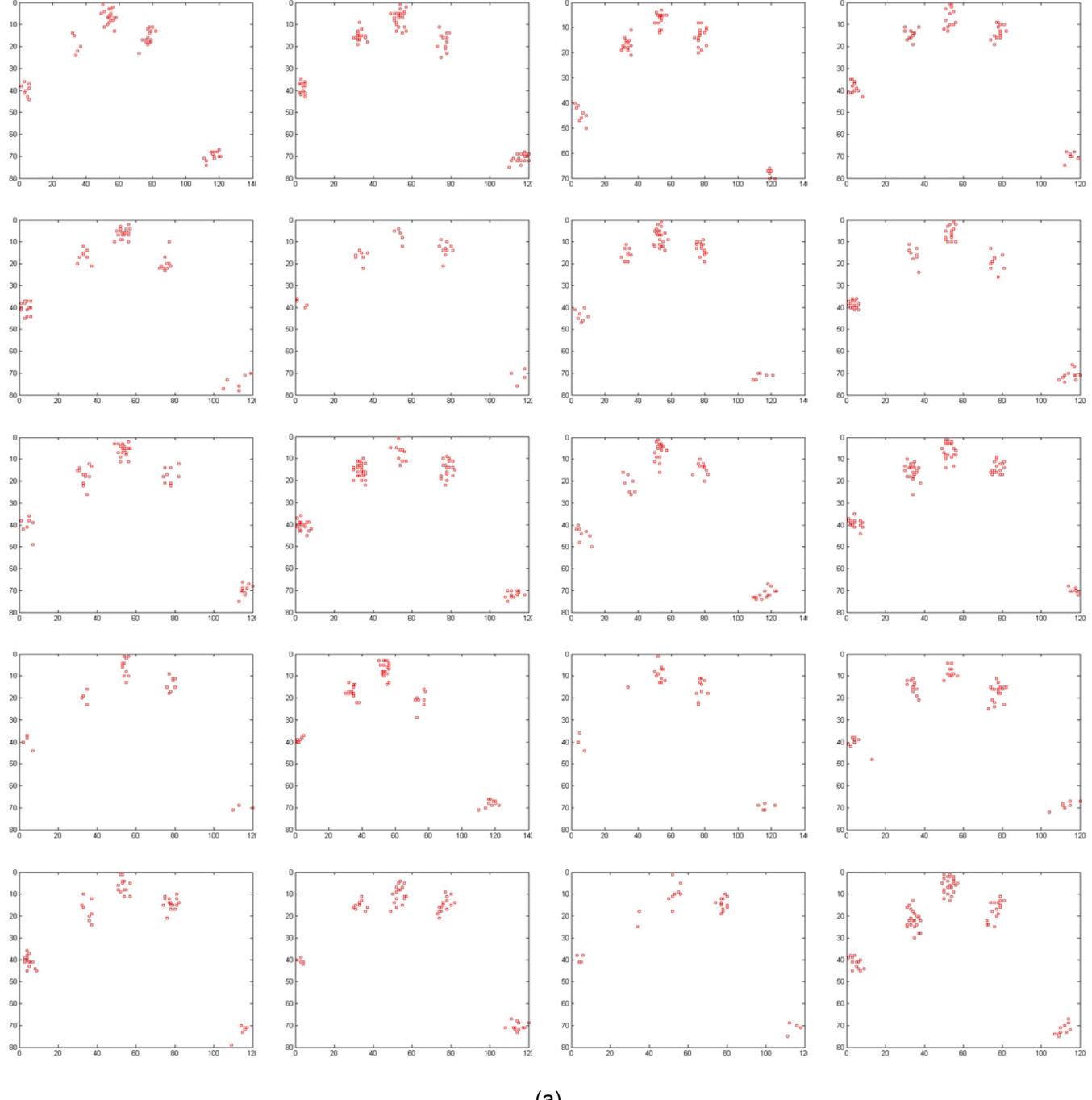
It should be pointed out that, due to the randomness of the experimental results of the proposed method, the accuracy of fingertip localization of the proposed method should be further improved. It is of necessity to study appropriate methods to improve the accuracy of fingertip localization.

Acknowledgments

This work was jointly supported by National Natural Science Foundation of China, Grant nos. 61403155, 61473299, 61375067, and 61773384. Thanks to Dr. Fanlin Meng, Senior Lecturer of The University of Edinburgh, Edinburgh, UK, for linguistic advice.

Appendix A

Figs. 1a, 2a, 3a, 4a, 5a and 1b, 2b, 3b, 4b, 5b are shown in this section.



(a)

Fig. 1a. The pixel sets of Fig. 4a (1) obtained by the proposed method.

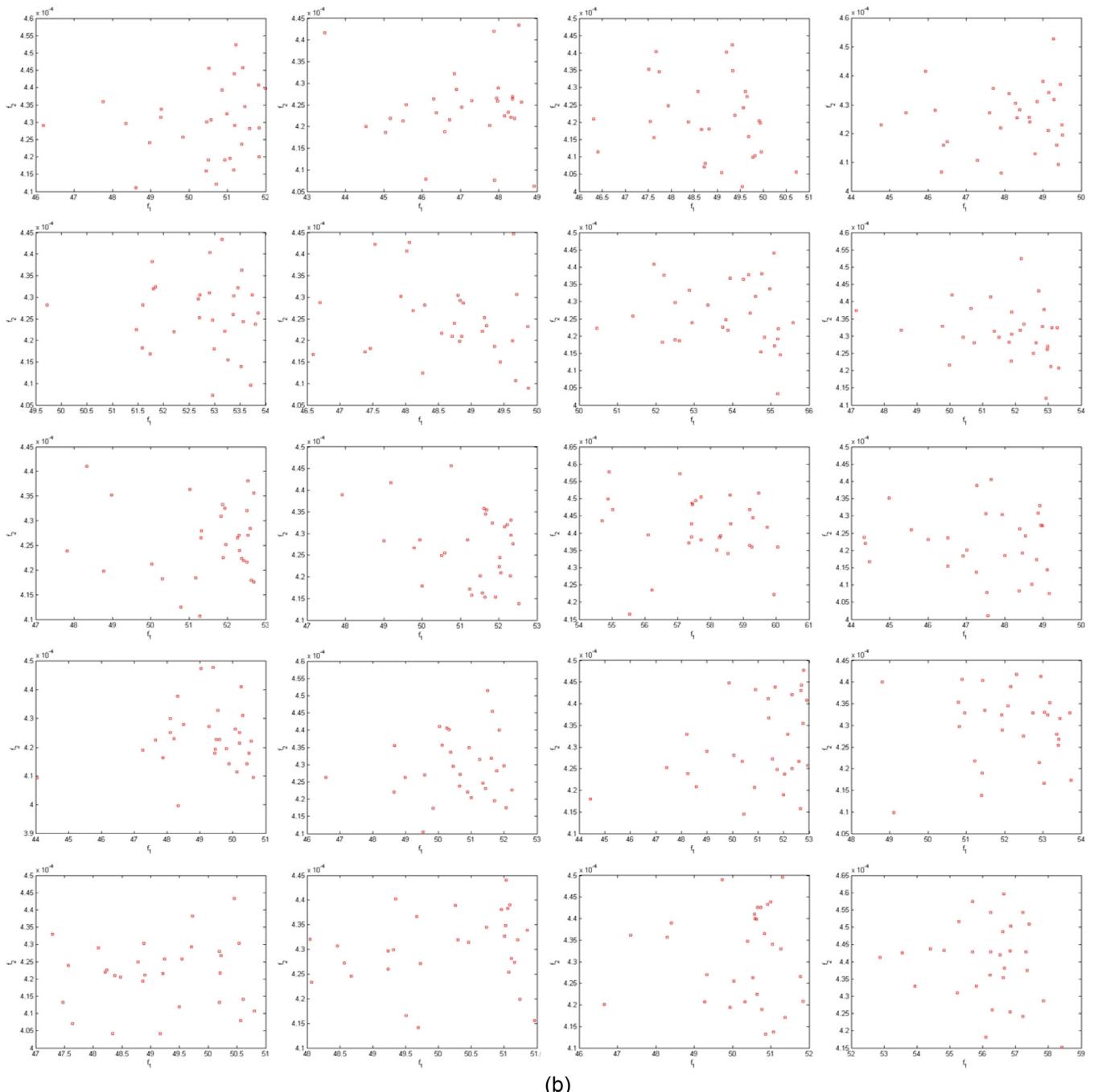


Fig. 1b. The populations at the last generation in the objective space of Fig. 4b (1).



(a)

Fig. 2a. The pixel sets of Fig. 4a (2) obtained by the proposed method.

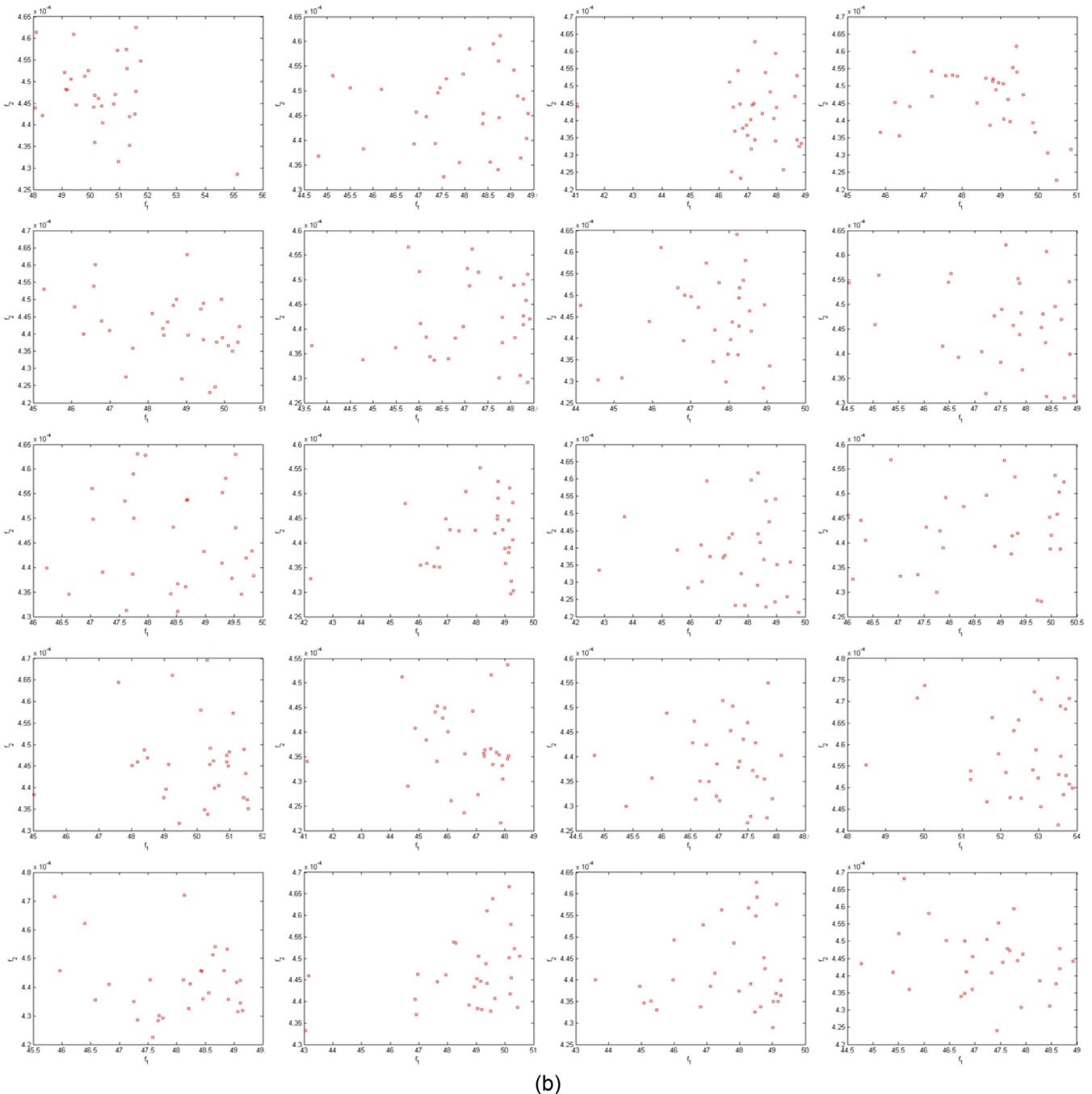
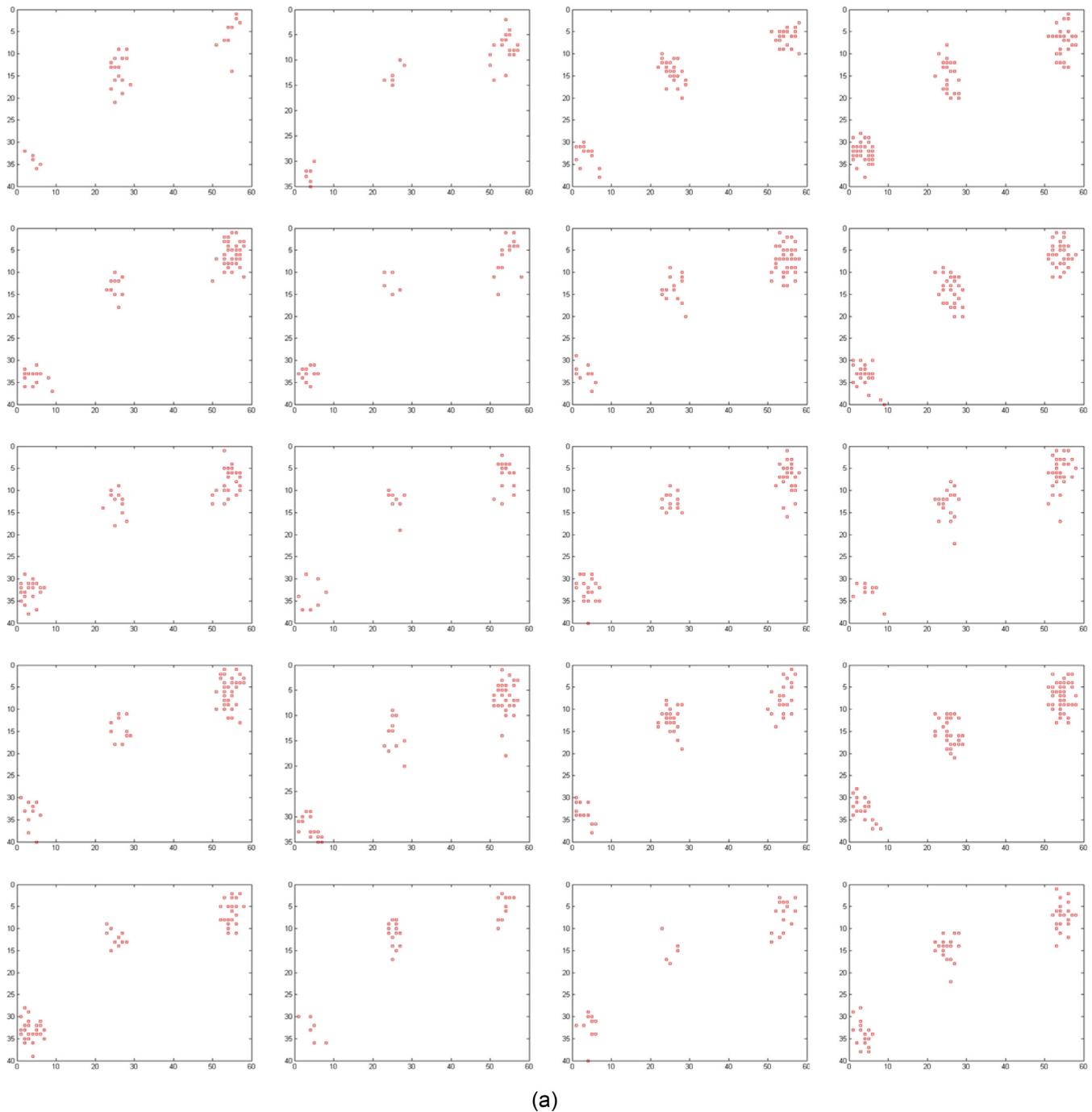


Fig. 2b. The populations at the last generation in the objective space of Fig. 4b (2).



(a)

Fig. 3a. The pixel sets of Fig. 4a (3) obtained by the proposed method.

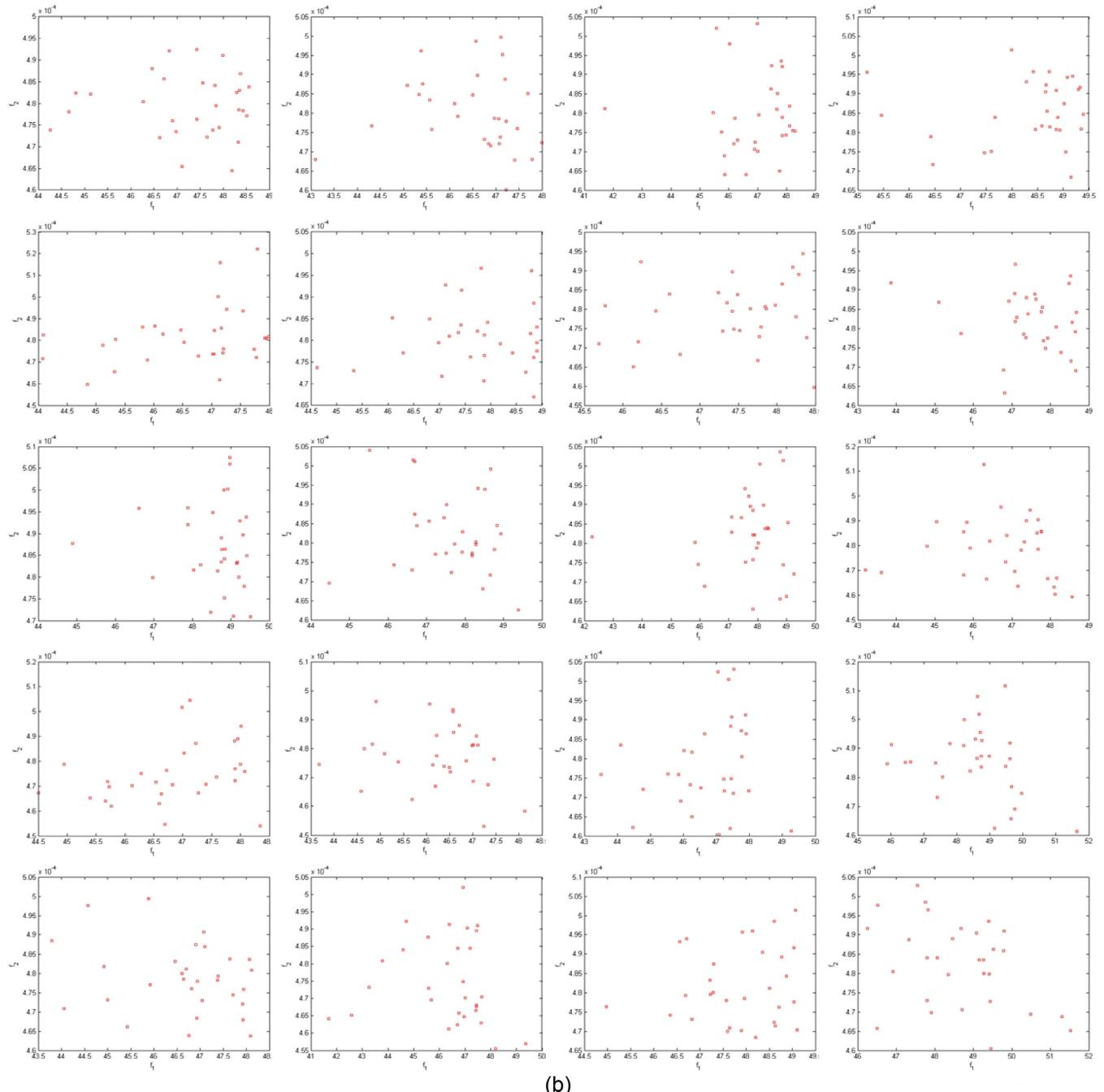


Fig. 3b. The populations at the last generation in the objective space of Fig. 4b (3).

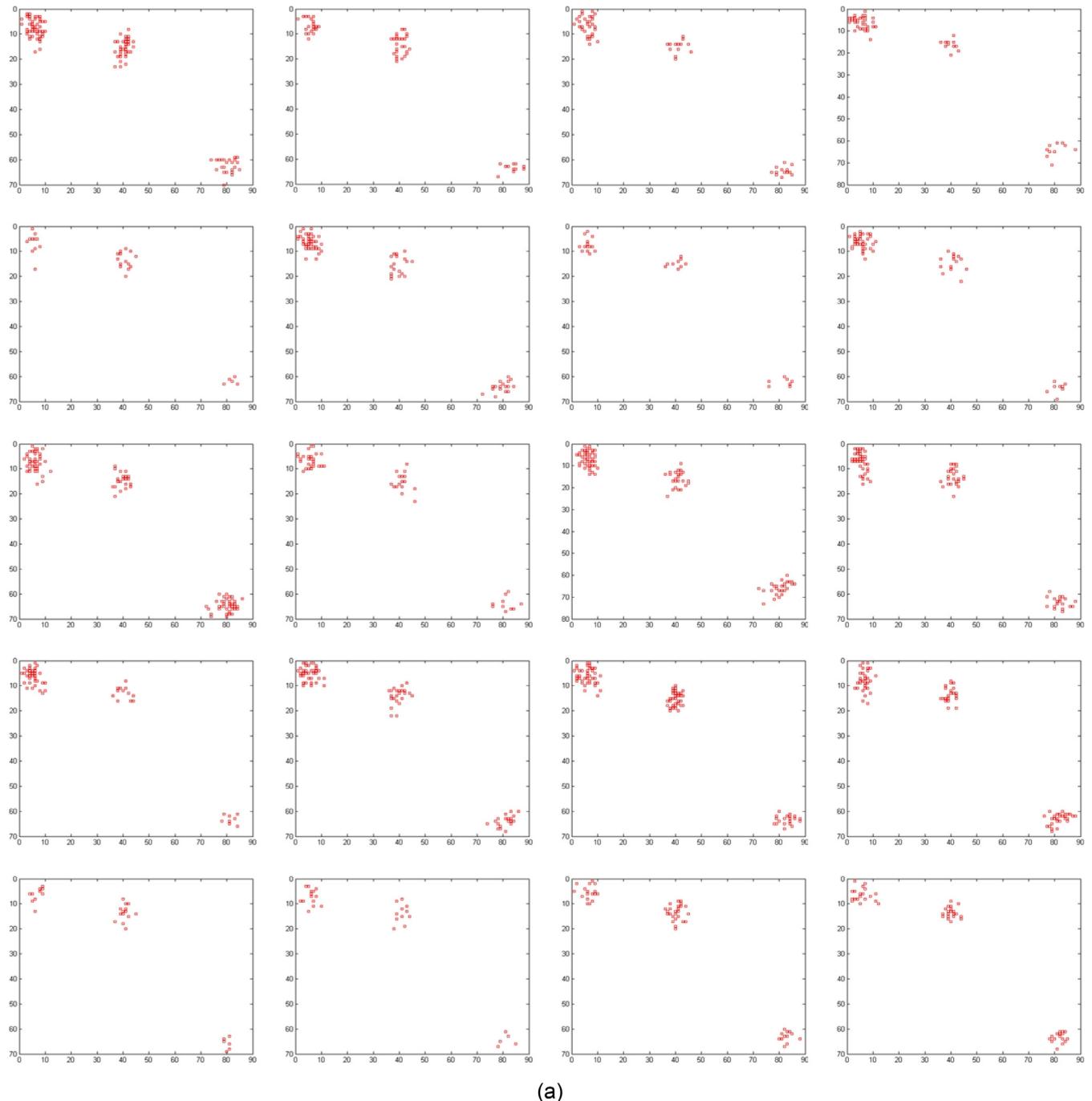


Fig. 4a. The populations at the last generation in the objective space of Fig. 4a (3).

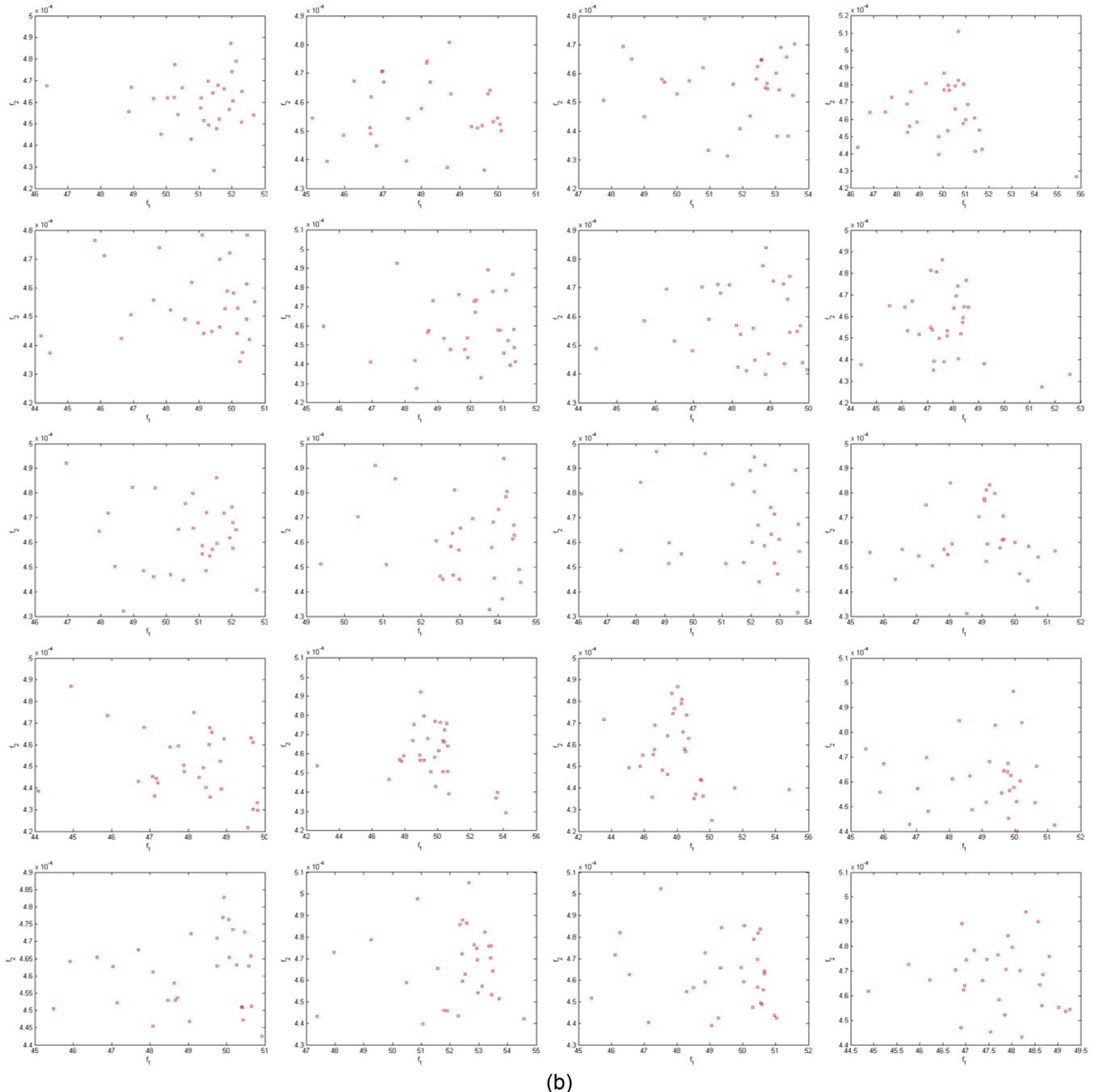


Fig. 4b. The populations at the last generation in the objective space of Fig. 4b(4).

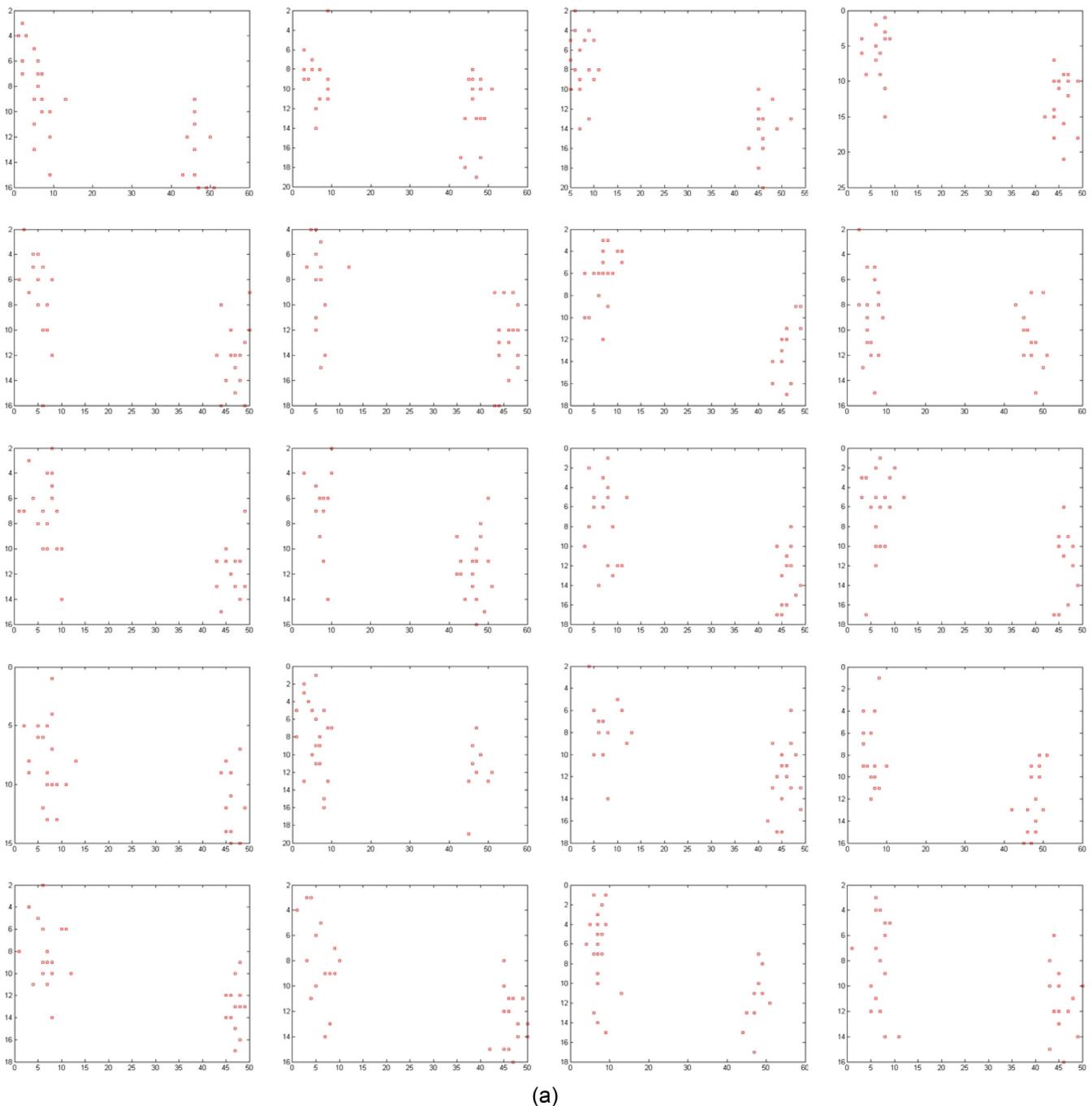


Fig. 5a. The pixel sets of Fig. 4a (5) obtained by the proposed method.

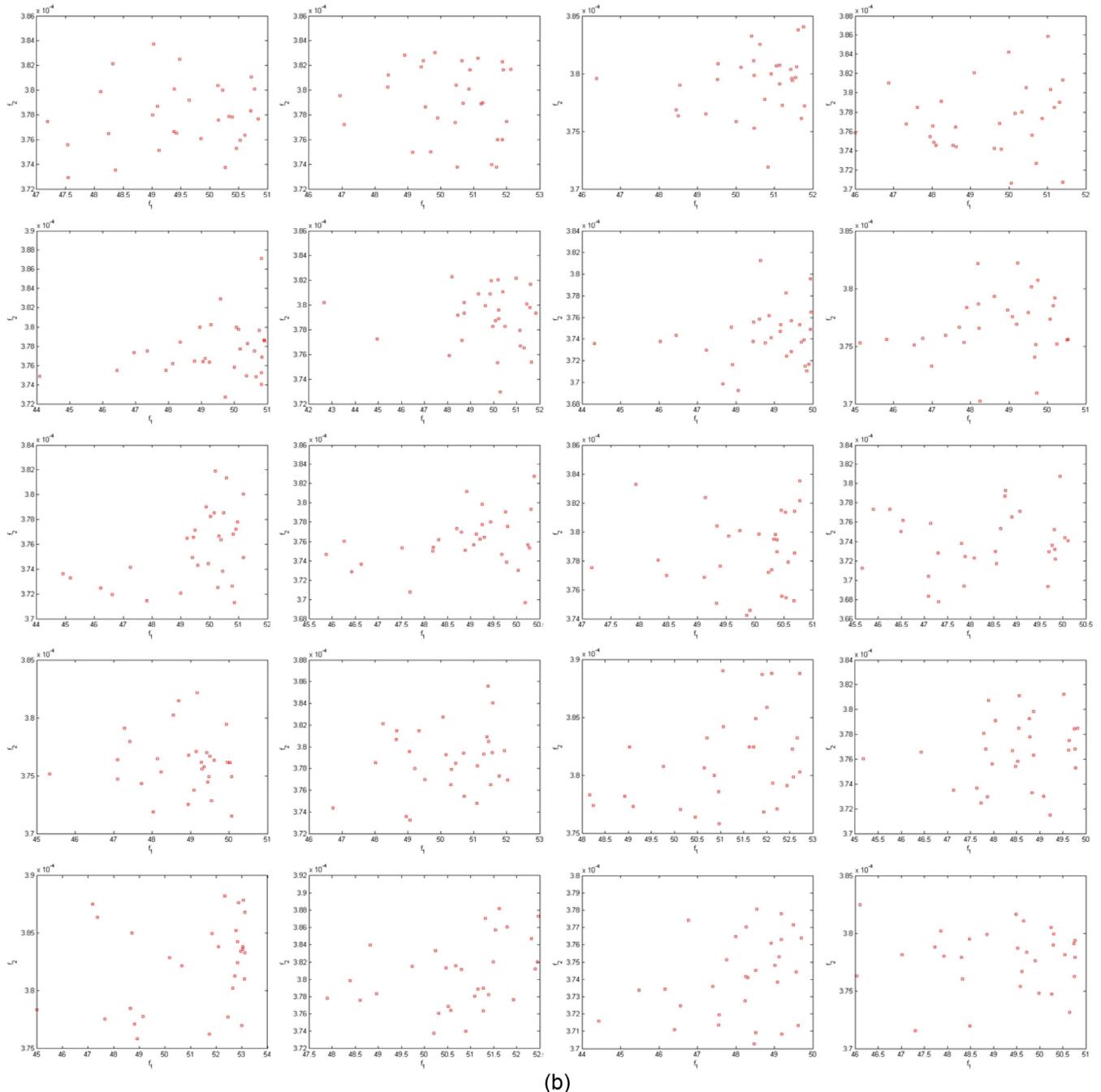


Fig. 5b. The populations at the last generation in the objective space of Fig. 4b (5).

References

- [1] J. Wan, G. Guo, S.Z. Li, Explore efficient local features from RGB-D data for one-shot learning gesture recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (8) (2016) 1626–1639.
- [2] D. Wu, L. Pigou, P. Kindermans, N. D. Le, L. Shao, J. Dambre, J. Odobez, Deep dynamic neural networks for multimodal gesture segmentation and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, 38(8), 1583–1597.
- [3] J. Letessier, F. Berard, Visual tracking of bare fingers for interactive surfaces, 2004, Proceedings of the Seventeenth ACM Annual Symposium on User Interface Software and Technology, 119–122.
- [4] S. Kim, Y. Park, K. Lim, Fingertips detection and tracking based on active shape models and an ellipse, 2009, Proceedings of the Tencor IEEE Region 10 Conference, 1–6.
- [5] K. Oka, Y. Sato, H. Koike, Real-time fingertip tracking and gesture recognition, *Comput. Gr. Appl.* 22 (6) (2002) 64–71.
- [6] Z.H. Chen, J.T. Kim, J. Liang, Real-time hand gesture recognition using finger segmentation, *Sci. World J.* 267872 (2014) 1–9.
- [7] M.K. Bhuyan, D.R. Neog, M.K. Kar, Fingertip detection for hand pose recognition, *Int. J. Comput. Sci. Eng.* 4 (3) (2012) 501–511.
- [8] J. Barrho, M. Adam, U. Kiencke, Finger localization and classification in images based on generalized hough transform and probabilistic models, in: Proceedings of the Ninth IEEE International Conference on Control, Automation, Robotics and Vision, 2006, pp. 1–6.
- [9] J. Park, B. Oh, K. Hong, A robust fingertip detection algorithm for mobile user interfaces, *Int. J. Bio-Sci. Bio-Technol.* 5 (3) (2013) 69–78.
- [10] G. Wu, W. Kang, Robust fingertip detection in a complex environment, *IEEE Trans. Multimed.* 18 (6) (2016) 978–987.
- [11] X. Suau, M. Alcoverro, A. Lopez-Mendez, J. Ruiz-Hidalgo, J. Casas, Real-time fingertip localization conditioned on hand gesture classification, *Image Vis. Comput.* 32 (8) (2014) 522–532.
- [12] P. Prasertsakul, T. Kondo, A fingertip detection method based on the top-hat transform, in: Proceedings of the Eleventh IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2014, pp. 1–5.
- [13] M. Wang, J. Lin, G.Q. Meng, Fingertip detection and gesture recognition based on contour approximation, *Int. J. Pattern Recognit. Artif. Intell.* 29 (7) (2015) 1–14.
- [14] E.S. Candela, M.O. Perez, C.M. Romero, Human top: a multi-object tracking tabletop, *Multimed. Tools Appl.* 70 (3) (2014) 1837–1868.
- [15] Y. Jang, S.T. Noh, H.J. Chang, 3D finger CAPE: clicking action and position estimation under self-occlusions in egocentric viewpoint, *IEEE Trans. Vis. Comput. Graph* 21 (4) (2015) 501–510.
- [16] G. Heo, D. Lee, H. Shin, H. Jeong, Hand segmentation and fingertip detection for interfacing of stereo vision-based smart glasses, in: Proceedings of the IEEE International Conference on Consumer Electronics, 2015, pp. 585–586.
- [17] C. Ho, C. Wu, Y. Chiang, F. Chang, Fingertip detection technique for a wearable projection device, *J. Comput. Inf. Syst.* 10 (24) (2014) 10839–10848.
- [18] S. Hashemi, S. Kiani, N. Noroozi, M.E. Moghaddam, An image contrast enhancement method based on genetic algorithm, *Pattern Recognit. Lett.* 31 (2010) 1816–1824.
- [19] Y.D. Fougerolle, J. Gielis, F. Truchetet, A robust evolutionary algorithm for the recovery of rational Gielis curves, *Pattern Recognit.* 46 (2013) 2078–2091.
- [20] R. Sarkhel, N. Das, A.K. Saha, M. Nasipuri, A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition, *Pattern Recognit.* 58 (2016) 172–189.
- [21] H. Xia, J. Zhuang, D. Yu, Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data, *Pattern Recognit.* 46 (2013) 2562–2575.
- [22] J.D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: Proceedings of the First International Conference on Genetic Algorithms and Their Application, 1985, pp. 93–100.
- [23] N. Srinivas, K. Deb, Multiobjective optimization using non-dominated sorting in genetic algorithms, *Evolut. Comput.* 2 (3) (1994) 221–248.
- [24] K. Deb, A. Pratap, S. Agarwal, A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans. Evolut. Comput.* 6 (2) (2002) 182–197.
- [25] B. Sanghamitra, M. Arpan, An algorithm for many-objective optimization with reduced objective computations: a study in differential evolution, *IEEE Trans. Evol. Comput.* 19 (3) (2015) 400–413.
- [26] F. Lobato, C. Sales, I. Araujo, V. Tadaiquesky, L. Dias, L. Ramos, A. Santana, Multi-objective genetic algorithm for missing data imputation, *Pattern Recognit. Lett.* 68 (2015) 126–131.
- [27] M. Arevalillo-Herraez, F.J. Ferri, S. Moreno-Picot, Improving distance based image retrieval using non-dominated sorting genetic algorithm, *Pattern Recognit. Lett.* 53 (2015) 109–117.
- [28] H. Karshenas, R. Santana, C. Bielza, Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables, *IEEE Trans. Evol. Comput.* 18 (4) (2014) 519–542.
- [29] K. Yang, L. Mu, D. Yang, Multiobjective memetic estimation of distribution algorithm based on an incremental tournament local searcher, *Sci. World J.* 836272 (2014) 1–21.
- [30] I. Oh, J. Lee, B. Moon, Hybrid genetic algorithms for feature selection, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (11) (2004) 1424–1437.
- [31] L. Silva, O. Bellon, K.L. Boyer, Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (5) (2005) 762–776.
- [32] M. Laszlo, S. Mukherjee, A genetic algorithm using hyper-quadtrees for low-dimensional k-means clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (4) (2006) 533–543.
- [33] C. Fang, R. Kolisch, L. Wang, C. Mu, An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem, *Flex. Serv. Manuf. J.* 27 (4) (2015) 1–21.
- [34] F. Zhao, Z. Shao, J. Wang, C. Zhang, A hybrid differential evolution and estimation of distribution algorithm based on neighbourhood search for job shop scheduling problems, *Int. J. Prod. Res.* 54 (4) (2016) 1039–1060.
- [35] Q.F. Zhang, A.M. Zhou, Y. Jin, RM-MEDA: a regularity model based multiobjective estimation of distribution algorithm, *IEEE Trans. Evol. Comput.* 12 (1) (2008) 41–63.
- [36] X. Liang, H. Chen, J. Lozano, A Boltzmann-based estimation of distribution algorithm for a general resource scheduling model, *IEEE Trans. Evol. Comput.* 19 (6) (2015) 793–806.
- [37] J. Ceberio, E. Irurzoki, A. Mendiburu, J. Lozano, A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem, *IEEE Trans. Evol. Comput.* 18 (2) (2014) 286–300.
- [38] J. Wang, K. Tang, J. Lozano, X. Yao, Estimation of the distribution algorithm with a stochastic local search for uncertain capacitated arc routing problems, *IEEE Trans. Evol. Comput.* 20 (1) (2016) 96–109.
- [39] S. Wang, L. Wang, An estimation of distribution algorithm-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem, *IEEE Trans. Syst. Man Cybern. Syst.* 46 (1) (2016) 139–149.
- [40] L. Wang, C. Fang, P. Suganthan, M. Liu, Solving system-level synthesis problem by a multi-objective estimation of distribution algorithm, *Expert Syst. Appl.* 41 (5) (2014) 2496–2513.
- [41] S. Zahra, M.A. Ghazanfar, A. Khalid, Novel centroid selection approaches for KMeans-clustering based recommender systems, *Inf. Sci. (NY)* 320 (1) (2015) 156–189.
- [42] G. Gan, M.K. Ng, K-Means clustering with outlier removal, *Pattern Recognit. Lett.* 90 (2017) 8–14.
- [43] L. Bai, X. Cheng, J. Liang, H. Shen, Y. Guo, Fast density clustering strategies based on the k-means algorithm, *Pattern Recognit.* 71 (2017) 375–386.
- [44] A.L. Barczak, N.H. Reyes, M. Abastillas, A new 2d static hand gesture colour image dataset for ASL gestures, *Res. Lett. Inf. Math. Sci.* 15 (2011) 12–20.
- [45] A. Rosenfeld, J.L. Pfaltz, Sequential operations in digital picture processing, *J. Ass. Comput. Mach.* 13 (4) (1966) 471–494.

Dunwei Gong is a professor in School of Information and Electrical Engineering, China University of Mining and Technology.

Ke Liu is a doctoral student in School of Information and Electrical Engineering, China University of Mining and Technology.