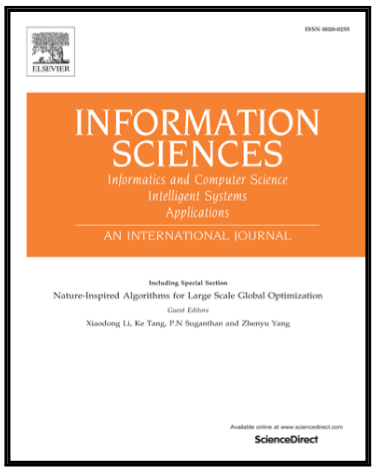

Gesture segmentation based on a two-phase estimation of distribution algorithm

Ke Liu, Dunwei Gong, Fanlin Meng, Huanhuan Chen, Gai-Ge Wang

PII: S0020-0255(17)30496-6
DOI: 10.1016/j.ins.2017.02.021
Reference: INS 12744

To appear in: *Information Sciences*

Received date: 19 April 2016
Revised date: 7 February 2017
Accepted date: 10 February 2017

Please cite this article as: Ke Liu, Dunwei Gong, Fanlin Meng, Huanhuan Chen, Gai-Ge Wang, Gesture segmentation based on a two-phase estimation of distribution algorithm, *Information Sciences* (2017), doi: 10.1016/j.ins.2017.02.021

Gesture segmentation based on a two-phase estimation of distribution algorithm

Ke Liu[a], Dunwei Gong[a,b,*], Fanlin Meng[c], Huanhuan Chen[d], Gai-Ge Wang[e]

[a]*School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China*
[b]*School of Electrical Engineering and Information Engineering, LanZhou University of Technology, Lanzhou 730050, China*
[c]*School of Engineering and Computing Sciences, Durham University, Durham DH1 3LE, UK*
[d]*School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China*
[e]*School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116, China*

Abstract

A multi-objective optimization model for the problem of gesture segmentation is formulated, and a method of solving the model based on a two-phase estimation of distribution algorithm is presented. When building the model, the positions of a series of pixels are taken as the decision variable, and the differences between the colors of pixels and those of a hand are taken as objective functions. A method of gesture segmentation based on a two-phase estimation of distribution algorithm is proposed according to the correlation among the positions of pixels. The method divides the solution of the problem based on evolutionary optimization into two phases, and uses different estimation of distribution algorithms in different phases. In the first phase, the probability model of candidates is formulated by a number of intervals given the fact that the positions of hand pixels distribute in several intervals. In the second phase, the probability model of candidates is built through a series of segments since the positions of

*Corresponding author at: School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China. Tel.: +86 516 83995312; fax: +86 516 83995312.

Email addresses: liu791018@126.com (Ke Liu), dwgong@vip.163.com (Dunwei Gong), flmeng086@gmail.com (Fanlin Meng), hchen@ustc.edu.cn (Huanhuan Chen), gaigewang@163.com (Gai-Ge Wang)

hand pixels further distribute around curves. A series of pixels constituting a hand region are obtained based on sampling by the above probability models. The proposed method is applied to 2515 problems of gesture segmentation, and is compared with the existing methods. The experimental results demonstrate the effectiveness of the proposed method.

Keywords: Gesture segmentation, Estimation of distribution algorithm, Two-phase, Probability model, Sampling

1. Introduction

Human-computer interaction is an important way to fulfill complex tasks. In the process of human-computer interaction, making a computer understand the behavior of a human is of considerable importance. Among various behaviors, the behavior based on human gestures expresses rich information. Therefore, correctly identifying a human gesture is very helpful for the cooperation of a human and a computer.

Segmenting a hand region from a background image is a precondition for correctly identifying a human gesture, which forms the problem of gesture segmentation. Image segmentation is one of the popular methods for solving this problem.

A color of a pixel can be expressed by a color space such as YCbCr or RGB. YCbCr is a representative color space in skin color segmentation [26]. In this space for skin color, component Y represents luminance, the value of which can be arbitrary. However, Cb or Cr refers to chrominance with its values obeying its own Gaussian distribution. It is further observed that the value of Cb or Cr of a skin color falls in an interval, and the best Cb or Cr value of a skin color is 109.38 or 152.02 respectively [9]. As a result, we can select pixels whose Cb or Cr values are closest to 109.38 or 152.02 from the pixels of a gesture image. Following that, we then take the selected pixels as the hand pixels, and obtain their Cb and Cr values. In this way, we can segment the hand region from the gesture image. Essentially, selecting the hand pixels whose Cb or Cr values are

closest to 109.38 or 152.02 is a multi-objective optimization problem. This is why gesture segmentation can be formulated as a multi-objective optimization problem.

The purpose of gesture segmentation is essentially to obtain a hand region from a gesture image. The existing methods of gesture segmentation are mainly implemented based on known skin colors. However, the color of a hand region is actually unknown and the obtained hand region via known skin colors might be inaccurate. For instance, the Cb or Cr value of the color of the obtained hand region is closest to 109.38 or 152.02. But the color may be regarded as a non-skin color according to known skin colors. Since gesture segmentation is formulated as a multi-objective optimization problem, we can obtain the hand region based on its own color. As a result, formulating gesture segmentation as a multi-objective optimization problem can obtain the hand region, which cannot be obtained by the existing methods.

To fulfill the task of image segmentation based on skin colors [7,15], a color space is first selected, a model of skin colors is then built based on a training sample set, and employed to distinguish pixels belonging to either skin colors or not. The distribution of skin colors in the color space is random and previous methods of segmenting skin colors have limitations for hand colors with a low probability of occurrence.

Estimation of Distribution Algorithm (EDA) is a population-based evolutionary algorithm guided by the theory of statistical learning. The algorithm expresses the distribution of candidates in a search space by a probability model. EDA first builds a probability model that reflects the distribution of candidates by using statistical learning, and then produces a temporary population by sampling the probability model. Finally, it generates an offspring population by selecting superior individuals from the united population formed based on the father population and the temporary population. EDA has been successfully applied in various fields [22,39,45].

Building the probability model of candidates is the core of EDA. When building the probability model, the following two aspects should be investigated, i.e.,

how to obtain information for building the probability model and how to express the probability model. For the first, information is the precondition of building a probability model, and precise information is helpful to building a reasonable probability model. The second is the base of generating new candidates by sampling, and a reasonable probability model is beneficial to the production of high quality candidates.

In this study, a multi-objective optimization model for the problem of gesture segmentation is formulated, and a method of solving the model based on a two-phase estimation of distribution algorithm is presented. When building the model, the positions of a series of pixels are taken as the decision variable, and the differences between the colors of pixels and those of a hand are taken as the objective functions. A method of gesture segmentation based on the two-phase estimation of distribution algorithm is proposed according to the correlation among the positions of pixels. The method divides the process of solving the problem based on evolutionary optimization into two phases, and uses different estimation of distribution algorithms in different phases. In the first phase, the probability model of candidates is formulated as a number of intervals given the fact that the positions of hand pixels distribute in several intervals. In the second phase, the probability model of candidates is built as a series of segments since the positions of hand pixels further distribute around curves. A series of pixels constituting a hand region are obtained based on sampling by the probability models. Having solved the formulated model, the best pixel set for forming a human gesture is obtained.

An evolutionary algorithm has the capability to solve a multi-objective optimization problem. The objective values of the multi-objective optimization problem are obtained by the color values of the pixels, and therefore are not continuous. As a result, the proposed evolutionary algorithm needs to deal with this discrete problem. Since the positions of hand pixels have correlations among themselves, we can guide the population evolution process to obtain accurate solutions in a limited time by using the above correlation characteristic. To this purpose, we propose an EDA to deal with the gesture segmentation problem

and quickly obtain accurate solutions. In the literature, although the famous evolutionary algorithm NSGA-II has been shown to deal with such a multi-objective optimization problem for gesture segmentation with discrete objective values, it does not take full advantage of the above correlation characteristic, and therefore its solutions are inferior to our proposed EDA.

In summary, this paper provides a feasible way to solve the problem of gesture segmentation and the following three contributions: (1) building a multi-objective optimization model for the problem of gesture segmentation, (2) proposing a two-phase estimation of distribution algorithm based on the correlation among the positions of hand pixels, and (3) verifying the effectiveness of the proposed model and algorithm by a series of experiments.

The remainder of this paper is arranged as follows. Section 2 reviews related work. A multi-objective optimization model for the problem of gesture segmentation is built in Section 3. Section 4 proposes a two-phase estimation of distribution algorithm based on the correlation among the positions of hand pixels. The applications of the proposed model and algorithm in real-world problems of gesture segmentation are provided in Section 5. Finally, Section 6 concludes the whole paper, and points out topics to be further studied.

2. Related work

In this study, we focus on the problem of gesture segmentation, formulate its multi-objective optimization model, and present a two-phase EDA to solve the model. Therefore, work related to this paper includes the following two main parts: methods of gesture segmentation based on skin colors and evolutionary algorithms for multi-objective optimization problems.

2.1. Gesture segmentation based on skin colors

Gesture segmentation based on skin colors can generally be divided into the following two steps: transforming a color space and modeling skin colors.

Different color spaces represent different features of a skin color. The distribution of skin colors is concentrated in a variety of color spaces, and is impacted by luminance. The color spaces that easily distinguish luminance from chrominance, such as YCbCr and HSV, can be employed to reduce the negative influence of luminance [18,29].

In order to build a model of skin colors, a mathematical expression of the skin colors based on a given sampling data set is formulated. Previous methods of modeling skin colors can be mainly divided into the following three categories, restricting the range of the values of each skin color, building the Gaussian distribution model of skin colors, and depicting the histogram that reflects the probability distribution of skin colors.

Restricting the range of the values of each skin color can be done using relatively simple methods. In reference [7], the values of Cb and Cr of skin colors are restricted to meet the following conditions: $77 \leq Cb \leq 127$ and $133 \leq Cr \leq 173$. In reference [15], the values of Cb and Cr of skin colors are in specific ranges related to Y, and a pixel can be taken as a skin pixel if the values of its HSV vector are in a tri-dimensional space.

The Gaussian distribution model of skin colors includes the case with a single peak [9,24] and the case with a mixed Gaussian distribution [13,23]. The Gaussian distribution model is built based on the fact that the values of skin colors in a color space are continuous, and obey the Gaussian distribution. Based on the Gaussian distribution model with a single peak, the similarity between the color vector of a pixel and that of the best skin color can be calculated. The model with the mixed Gaussian distribution consists of several models with a single peak [23], and can approximate any distribution model in an arbitrary shape. Compared with the model with a single peak, the mixed counterpart is a more accurate reflection of the distribution of skin colors.

The probability distribution histogram of skin colors depicts the probability of which skin colors emerge in each unit of a color space. To obtain this histogram, the probability of which a sampling data set emerges in each unit is calculated. The methods of seeking pixels of skin colors based on the above

histogram can be divided into the following two categories, Bayesian classifiers and regularized query tables [3,8].

In recent years, there have been many research achievements with regard to gesture segmentation. In [21], Karishma et al. proposed an algorithm incorporating a skin color model and background subtraction that yields robust output in the presence of drastic illumination changes. In [42], Zhang et al. presented a hand segmentation method based on the fuzzy C-means clustering algorithm and the mixed skin-color model. In this method, images handled by the mixed skin-color model and the clustering algorithm are processed by the morphological process and the logic operation. In [43], Zhang et al. proposed a method based on YCbCr and the k-means clustering algorithm for gesture segmentation. Based on this method, a hand binary image is obtained by clustering the values of Cb and Cr, and gesture segmentation is fulfilled by conducting the morphological process of a hand binary image. In [20], Ju et al. proposed a modified expectation-maximization algorithm to segment gestures in the RGB-Depth database. They proposed an approach to refine the edge of the tracked gesture by applying this algorithm based on Bayesian networks. The proposed approach has potential to improve the performance of gesture recognition. Dawod et al. employed contrast adjustment and motion detection analysis to obtain the start and the end points of each individual moving gesture from continuous gestures [10]. Mahmoodi et al. obtained skin regions by using a fusion feedback mechanism, a modified Bayesian classifier, and ternary-based human motion detection [27]. Gupta et al. built a system with three robust algorithms based on different color spaces for skin classification [17]. Wu et al. proposed a semi-supervised hierarchical dynamic framework based on a Hidden Markov Model for gesture segmentation [38].

2.2. Estimation of distribution algorithms

In real-world applications, one often encounters problems with simultaneously optimizing multiple objectives under specific conditions, also known as multi-objective optimization problems. In a multi-objective optimization prob-

lem, its objectives often conflict with each other. That is, the improvement of one objective is at the cost of the deterioration of 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, many scholars have combined evolutionary algorithms with multi-objective optimization, developed a variety of methods, and formed a popular research topic, i.e., multi-objective evolutionary optimization.

Rosenberg first employed evolutionary algorithms to solve multi-objective optimization problems. Deb et al. proposed a competent evolutionary algorithm, NSGA-II [12], for solving multi-objective optimization problems. Later a variety of efficient multi-objective evolutionary algorithms have been proposed based on NSGA-II [4,11].

Jin et al. pointed out that using Pareto-based multi-objective optimization is a good choice to tackle machine learning problems, particularly due to the great success of multi-objective evolutionary optimization and other population-based optimization [19]. Pareto-based multi-objective learning has been successful in addressing a variety of topics on machine learning, such as feature selection, clustering, generalization, ensemble, and knowledge extraction. Multi-objective learning can gain a deep insight into a problem by analyzing the Pareto front. An overview on Pareto-based multi-objective learning and case studies to illustrate its advantages were provided in [19]. Albukhanajer et al. adopted multi-objective evolutionary optimization to tackle functions in the Trace Transform for extracting image features robust to noise and invariant to geometric deformations such as rotation, scale, and translation (RST) [2]. To this end, they employed sample images with noise and RST distortion when optimizing the Trace Transform with multi-objective evolutionary optimization, termed evolutionary Trace Transform with noise (ETTN). Experimental studies demonstrate that the proposed ETTN is very promising in that it is computationally efficient, invariant to RST deformation, robust to noise, and generalizable.

Compared with genetic algorithms based on the micro mode in the search space, Estimation of Distribution Algorithm is based on the macro counterpart

in the search space [24,34], and has a stronger capability for exploration and a more rapid convergence speed [14,40]. Zhang et al. proposed the regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) [31,41], 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 (m-1)-dimensional manifold in the objective space. Here, m 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 that was originally used 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.

In multi-objective evolutionary optimization, how to generate new trial solutions has not been well studied. Crossover and mutation are directly used in most of the current multi-objective evolutionary algorithms. This could be one of the major reasons why these algorithms do not perform well on multi-objective optimization problems with variable linkages. In reference [41], the regularity property of continuous multi-objective optimization problems was used as a basis for an estimation of distribution algorithm when dealing with variable linkages. RM-MEDA models a promising area in the search space by a probability model whose centroid is a continuous manifold. Continuous two-objective evolutionary optimization uses the 1-dimensional local principal component analysis to build its probability model of several segments.

In reference [45], Zhou et al. suggested combining an EDA with cheap and expensive local search methods to make use of both individual location information and global statistical information. This EDA is used to solve a continuous single-objective optimization problem. In this EDA, parts of new solutions are generated by refining a parent solution through a cheap local search method that does not need any function evaluation, and the rest is sampled from

a modified univariate histogram probabilistic model. The modified univariate histogram probabilistic model is the histogram with variable width, and is based on the distribution of each component of the optimal solution of the continuous single-objective optimization problem. An expensive local search method is used to improve a promising solution found when the population converges.

Other research achievements of EDA have been obtained in recent years. Reference [25] proposed a Boltzmann-based EDA for resource scheduling. This EDA is based on an approximation of the Boltzmann distribution. Such an approximation method is a tradeoff between solution accuracy and complexity. Reference [6] introduced a hybrid approach consisting of a variable neighborhood search and a new EDA to deal with the flow-shop scheduling problem. This new EDA has the ability to discover promising regions in the search space. Reference [35] 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. The stochastic local search procedure avoids excessive fitness evaluations in the local search. Reference [37] 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, a novel selective-enhancing sampling mechanism for generating new solutions by sampling the probability model is proposed. The EDA and a local search are incorporated within the memetic algorithm framework. Reference [36] 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, a Pareto archive is used to preserve the non-dominated solutions, and another archive is used to preserve the solutions for updating the probability model. In addition, a specific updating mechanism and a sampling mechanism are provided for the probability model to track the most promising search area.

Zhou et al. presented an EDA to minimize makespan in a no-wait flow-shop scheduling problem with two batch processing machines [46]. Valdez et

al. presented a structure-control design method based on an EDA for simultaneously optimizing both the mechanical structure and the control system of a parallelogram linkage robot [33]. Wan et al. used an EDA to solve a special class of nonlinear bilevel programming problems in which the lower level problem is a convex programming problem for each given upper level decision [34]. Muelas et al. presented a comparative study for a distributed univariate EDA and a multivariate version over a wide set of parameters and problems, and compared two alternative methods for exchanging information [28]. Sun et al. presented a multi-cycled sequential memetic computing structure for constrained optimization, and applied an EDA to explore the search space until convergence at each cycle [32]. Alberto et al. compared several variation operators based on Pareto efficiency, extracted from EDAs, differential evolution, evolutionary programming, and evolution strategies to determine whether they increase the performance of the non-Pareto based versions or not [1]. Giagkiozis et al. created a method based on an EDA with low-order statistics, and generalized decomposition to solve many-objective optimization problems [16].

3. A multi-objective optimization model for gesture segmentation

A color of a pixel is composed of red, green and blue (RGB), and each of them has 256 gradations of brightness in image processing. RGB is a basic color space, and it can be transformed to other color spaces such as YCbCr. The color of each pixel in a gesture image can be expressed by Y, Cb and Cr with different gradations. Luminance and chrominance are separated in YCbCr. A color with different luminance can be expressed by the color plane, i.e., CbCr. According to [15], the skin colors with different luminance form a region in the color plane CbCr, but the shape of that region is uncertain. Due to the randomness of the distribution of skin colors, it is impossible to propose a specific expression for the distribution region of all skin colors in the CbCr plane.

YCbCr is a commonly used color space for segmenting skin colors. According to [15], the values that have the best match with those of Cb and Cr of skin

colors are 109.38 and 152.02, respectively. Therefore, one can select a series of pixels with the closest values to those of the best skin color from a gesture image.

To fulfill this task, the following multi-objective optimization model is formulated:

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

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 $x = (x^1, x^2, \cdots, x^n),\ x^i = (x_i, y_i),\ i = 1, 2, \cdots, n$. Here, n is the number of the selected pixels. In addition, $x_i \in [\underline{x}, \overline{x}], y_i \in [\underline{y}, \overline{y}]$ where $\underline{x}$ is the lower bound of x_i, $\overline{x}$ is the upper bound of x_i, $\underline{y}$ is the lower bound of y_i, and $\overline{y}$ is the upper bound of y_i.

In addition, $f_1(x)$ means the distance between the values of Cb of a set of pixels and that of the best skin color, whose expression is provided as follows:

$$f_1(x) = \frac{\sum_{i=1}^{n} |Cb(x^i) - \overline{Cb}|}{n} \tag{2}$$

where $Cb(x^i)$ is the value of Cb of x^i, and $\overline{Cb} = 109.38$ represents that of the best skin color.

$f_2(x)$ reflects the distance between the values of Cr of a set of pixels and that of the best skin color, with its expression as follows:

$$f_2(x) = \frac{\sum_{i=1}^{n} |Cr(x^i) - \overline{Cr}|}{n} \tag{3}$$

where $Cr(x^i)$ is the value of Cr of x^i, and $\overline{Cr} = 152.02$ means that of the best skin color.

One can obtain a series of pixels by solving (1), and form the ranges of the values of Cb and Cr of these pixels. If the values of both Cb and Cr of a pixel lie in the above ranges, the pixel will belong to a hand. In this way, the hand can be segmented from a specific gesture image.

4. The proposed algorithm

4.1. Ideas

In the gesture image, the hand region is a continuous and closed region, and its size is limited. Therefore, the horizontal coordinate x_i and the longitudinal coordinate y_i of the hand pixel x^i fall in bounded but unknown intervals respectively. A normal hand consists of five fingers and one palm where the fingers and the palm can each bend and rotate. When a finger is straight, its front or back projection on the plane can be represented by a rectangle. Since the rectangle's length is much larger than its width, the finger's projection can approximately define a line segment. However, when the finger bends and rotates, its projection on the plane will define a curve segment with the hand pixel x^i distributing around the curve segment. Further, when the palm is opened, its front or back projection on the plane can be similarly represented by several rectangles, and its projection defines several line segments. In contrast, when the palm bends and rotates, its projection defines several curve segments with the hand pixel x^i distributing around several curve segments.

Based on the above analysis, the hand pixel x^i has the following two features: (1) its horizontal coordinate x_i and its longitudinal coordinate y_i are in bounded intervals respectively; (2) x^i distributes around several curve segments.

If x is an optimal solution of model (1), each x^i in x represents the hand pixel, and each component of x meets the following conditions: (1) each x_i or y_i is bounded in an interval with a feasible range $[\underline{x}, \overline{x}]$ or $[\underline{y}, \overline{y}]$, respectively; (2) each $x^i = (x_i, y_i)$ distributes around a curve segment. In the following, we firstly build two probability models of the components of a candidate solution according to the above two conditions, and then propose two sampling methods for the above two established probability models. Finally we obtain all components of the candidate solution.

The distribution of x_i or y_i in an interval with the feasible range is uneven and irregular following condition (1). However, if we select the pixel whose x_i and y_i are both in their respective intervals, the selected pixel will fall into the

hand region, or the rectangular region determined by the hand region. To this end, we first build one probability model following condition (1) and obtain candidate solutions by sampling the probability model. After this step, we can obtain the coordinate ranges of the hand region. Secondly, we build the second probability model following condition (2), and obtain the candidate solutions by sampling the probability model. With this second step, we can eliminate non-hand pixels from the above rectangular region.

4.2. Algorithm framework

The evolutionary solution process of model (1) can be divided into two phases, which can be solved by two EDAs respectively. In the first phase, based on the fact that the position coordinates of the hand pixels distribute in the intervals, several intervals are built as the probability model of candidate components. The probability model of intervals causes the pixels of a population to locate in the rectangular region determined by the hand coordinate ranges, which is suitable for the randomly generated initial population. In the second phase, several line segments are built as the probability model of candidate components in order to eliminate non-hand pixels from the rectangular region, following the fact that the hand pixels distribute around several curve segments. This second probability model of line segments causes the population's pixels to locate in the hand region, and it is suitable for the population which has evolved.

In each generation t, after the iteration of the algorithm, the size of the population $P(t)$ is N. The algorithm works as follows.

Step 1: Set $t = 0$, and randomly generate an initial population $P(0)$.

Step 2: If the stopping condition is met, go to Step 9.

Step 3: If $t > T$, go to Step 6.

Step 4: Build the probability model of candidate component intervals by the non-dominated solution set of $P(t)$.

Step 5: Sample and obtain a temporary population, $Q(t)$, by the probability model of intervals, and then go to Step 8.

Step 6: Build the probability model of candidate component segments by the non-dominated solution set of $P(t)$.

Step 7: Sample and obtain a temporary population, $Q(t)$, by the probability model of segments.

Step 8: Select individuals and obtain the offspring population, $P(t+1)$, from $P(t) \cup Q(t)$ by the fast non-dominated sorting [12]. Set $t = t+1$, and go to Step 2.

Step 9: Return the non-dominated solutions of $P(t)$ and end the algorithm.

4.3. The probability model of candidate component intervals

We use the fast non-dominated sorting [12] to select all the non-dominated solutions of $P(t)$. According to condition (1), each x_i or y_i of an optimal solution of model (1) falls into an interval, i.e., $[a_1, a_{M-1})$ or $[b_1, b_{M-1})$, respectively. The distribution of x_i or y_i is uneven and irregular in $[a_1, a_{M-1})$ or $[b_1, b_{M-1})$. In order to express the distribution of x_i or y_i more accurately, we divide $[a_1, a_{M-1})$ or $[b_1, b_{M-1})$ into $M-2$ small intervals with the same width, i.e., $[a_{m-1}, a_m)$ or $[b_{m-1}, b_m), m = 2, 3, \cdots, M-1$. Therefore, the distribution of x_i or y_i is more even in each of these intervals. Following the above, we can use a histogram to express the distribution of x_i or y_i.

The distribution probability of x_i or y_i in $[\underline{x}, a_1)$ or $[\underline{y}, b_1)$ is 0. However, the distribution probability of x_i or y_i in each interval, $[a_{m-1}, a_m)$ or $[b_{m-1}, b_m), m = 2, 3, \cdots, M-1$ is not 0, and all the $M-2$ intervals are different from each other in general. Note that the distribution probability of x_i or y_i in $[a_{M-1}, \overline{x}]$ or $[b_{M-1}, \overline{y}]$ is 0. As a result, the distribution of x_i is regular, and can be represented by M intervals among which the first and the M-th intervals are $[\underline{x}, a_1)$ and $[a_{M-1}, \overline{x}]$, respectively, and the m-th $(m = 2, 3, \cdots, M-1)$ interval is $[a_{m-1}, a_m)$. Similarly, the distribution of y_i can also be represented by M intervals among which the first and the M-th intervals are $[\underline{y}, b_1)$ and $[b_{M-1}, \overline{y}]$, respectively, and the m-th $(m = 2, 3, \cdots, M-1)$ interval is $[b_{m-1}, b_m)$.

For each of the non-coincident pixels of the optimal solution set of model (1), the distribution of its x_i or y_i can also be expressed by a histogram. Thus,

based on each of the non-coincident pixels of the optimal solution set of the current population, $P(t)$, we can employ the histogram with variable width [45] to build the M intervals of its x_i or y_i. Following above, we can obtain the minimal and the sub-minimal values of x_i, i.e., $x^1_{\min}, x^2_{\min}$, and the maximal and the sub-maximal values of x_i, i.e., $x^1_{\max}, x^2_{\max}$. Similarly, we can obtain the minimal and the sub-minimal values of y_i, i.e., $y^1_{\min}, y^2_{\min}$, and the maximal and the sub-maximal values of y_i, i.e., $y^1_{\max}, y^2_{\max}$. With $x^1_{\min}, x^2_{\min}, x^1_{\max}, x^2_{\max}$ known, we can obtain M intervals of x_i, i.e., $[\underline{x}, a_1), [a_1, a_2), \cdots, [a_{m-1}, a_m), \cdots, [a_{M-1}, \overline{x}]$. Similarly, we can obtain M intervals of y_i, i.e., $[\underline{y}, b_1), [b_1, b_2), \cdots, [b_{m-1}, b_m), \cdots, [b_{M-1}, \overline{y}]$. In the above intervals,

$$a_1 = \max\left\{x^1_{\min} - 0.5(x^2_{\min} - x^1_{\min}), \underline{x}\right\},$$

$$a_{M-1} = \min\left\{x^1_{\max} + 0.5(x^1_{\max} - x^2_{\max}), \overline{x}\right\},$$

$$b_1 = \max\left\{y^1_{\min} - 0.5(y^2_{\min} - y^1_{\min}), \underline{y}\right\},$$

$$b_{M-1} = \min\left\{y^1_{\max} + 0.5(y^1_{\max} - y^2_{\max}), \overline{y}\right\}.$$

Each interval between the second and the $(M-1)$-th interval of x_i or y_i has the same width, i.e.,

$$a_m - a_{m-1} = \frac{1}{M-2}(a_{M-1} - a_1)$$

or

$$b_m - b_{m-1} = \frac{1}{M-2}(b_{M-1} - b_1), (m = 2, 3, \cdots, M-1).$$

The width of either the first or the M-th interval changes with $P(t)$, while the width of each of the rest intervals changes with the width of either the first or the M-th interval. If $a_1 = \underline{x}$ or $a_{M-1} = \overline{x}$, the width of the first or the M-th interval of $[\underline{x}, \overline{x}]$ is 0. Similarly, if $b_1 = \underline{y}$ or $b_{M-1} = \overline{y}$, the width of the first or the M-th interval of $[\underline{y}, \overline{y}]$ is 0.

For each of the non-coincident pixels of the optimal solution set of the current population, $P(t)$, the number of its x_i or y_i falling into the m-th

$(m = 2, 3, \cdots, M-1)$ interval is denoted as $A_{q,m}, q = x$ or y. We further define

$$B_{q,m} = A_{q,m} + 1, \text{if } m = 2, 3, \cdots, M-1; B_{q,m} = 0.1, \text{if } m = 1, M.$$

In case the width of either the first or the M-th interval of the feasible range is 0, its $B_{q,m}$ is 0.

Following the above, the probability of candidate component x_i or y_i from the m-th interval is defined as

$$P'_{q,m} = \frac{B_{q,m}}{\sum_{m=1}^{M} B_{q,m}}.$$

For each of the non-coincident pixels of the optimal solution set of $P(t)$, its x_i distributes in one of the M intervals, i.e.,

$$[\underline{x}, a_1), [a_1, a_2), \cdots, [a_{m-1}, a_m), \cdots, [a_{M-1}, \overline{x}]$$

by probability $P'_{x,m}$ while its y_i distributes in one of the M intervals, i.e.,

$$[\underline{y}, b_1), [b_1, b_2), \cdots, [b_{m-1}, b_m), \cdots, [b_{M-1}, \overline{y}]$$

by probability $P'_{y,m}$. As a result, we can obtain x_i of the hand pixels by sampling $[\underline{x}, a_1), [a_1, a_2), \cdots, [a_{m-1}, a_m), \cdots, [a_{M-1}, \overline{x}]$ with probability $P'_{x,m}$, and y_i of the hand pixels by sampling $[\underline{y}, b_1), [b_1, b_2), \cdots, [b_{m-1}, b_m), \cdots, [b_{M-1}, \overline{y}]$ with probability $P'_{y,m}$. Finally, the above M intervals of x_i or y_i are the probability model for generating candidate components, and we call it the probability model of candidate component intervals.

Note that the value of M is mainly determined by the size of the hand region. If the value of M is small the accuracy of the probability model will be low, which is not beneficial to the evolution of the population. Otherwise, the computational complexity of the evolution will be high.

4.4. Sampling based on intervals

By sampling the probability model of candidate component intervals in the decision space, a candidate solution can be obtained. Candidate component x_i

or y_i should meet the following conditions: (1) the probability of generating x_i or y_i based on the m-th ($m = 1, 2, \cdots, M$) interval is $P'_{x,m}$ or $P'_{y,m}$, and (2) the distribution of x_i or y_i is even in the m-th interval. In addition, each pixel of a candidate is different from that of the others based on the definition of x in model (1).

We initially set $x = \phi$, and use the following steps to generate candidate x that meets the above conditions.

Step 1: Select the m-th interval from M intervals with the probability of $P'_{x,m}$. If $m = 1$ or M, the m-th interval is $[\underline{x}, a_1)$ or $[a_{M-1}, \overline{x}]$. If $m = 2, 3, \cdots, M-1$, the m-th interval is $[a_{m-1}, a_m)$. Generate a x_i by evenly sampling in the m-th interval.

Step 2: Select the m-th interval from M intervals with the probability of $P'_{y,m}$. If $m = 1$ or M, the m-th interval is $[\underline{y}, b_1)$ or $[b_{M-1}, \overline{y}]$. If $m = 2, 3, \cdots, M-1$, the m-th interval is $[b_{m-1}, b_m)$. Generate a y_i by even sampling in the m-th interval.

Step 3: If $x^i = (x_i, y_i) \notin x$, set $x = x \cup \{x^i\}$.

Step 4: If $|x| = n$, end the algorithm. Otherwise, go to Step 1.

The above steps are the process of obtaining a candidate by sampling. We execute the above process N times, and obtain N candidates. In this way, a temporary population $Q(t)$ can be obtained and its size will be N.

The histogram with variable width is used to build an EDA to solve a continuous single-objective optimization problem [45]. In model (1), the values of each decision component are continuous, but the values of each objective component are discrete. As a result, model (1) is a discrete multi-objective optimization problem. Each component of the optimal solution of a continuous single-objective optimization problem is in an interval, which also applies to each optimal solution of model (1). Therefore, we can also use the histogram with variable width to build an EDA to solve model (1). In the probability model of candidate component intervals, the number of the intervals of components of each optimal solution is two, i.e., $[a_1, a_{M-1})$ and $[b_1, b_{M-1})$. Instead, in the probability model of candidate components in reference [45], the number of

the intervals of components of the optimal solution equals the number of components. The rest of the probability model of candidate component intervals is the same as that in [45]. Furthermore, apart from the fact that in our sampling method of candidate components, each pixel of a candidate is different from the others, the rest of our sampling method is the same as that in [45].

4.5. The probability model of candidate component segments

After several iterations, many x^i of the population enter or approach the hand region. At each generation after T iterations, we build the probability model of candidate component segments based on the non-dominated solution set of $P(t)$, and obtain a temporary population $Q(t)$, until the population ends the evolution.

For x^i of the population at generation $T+1$, its x_i distributes in the range of x_i of the hand region, and its y_i distributes in the range of y_i of the hand region. That is, x^i distributes in a rectangular region determined by the hand region. However, x^i might distribute out of the hand region. According to condition (2) in section 4.1, all the hand pixels distribute around several curves. Following [41], a curve can be approximated by one or more segments. That is, all x^i of an optimal solution of model (1) distribute around several segments. Since all the non-coincident pixels of the optimal solution set of model (1) distribute in the hand region, and therefore distribute around several segments, we can use the 1-dimensional local principal component analysis algorithm [41] to build the probability model of candidate component segments.

We use the fast non-dominated sorting [12] to select all the non-dominated solutions of $P(t)$. We divide all the non-coincident x^i of the non-dominated solution set of $P(t)$ into L categories, with each corresponding to one segment. The L segments are denoted as $l_1, l_2, \cdots, l_L$, and the sets of the pixels distributing around the line segments are denoted as $C_1, C_2, \cdots, C_L$, respectively.

We employ the 1-dimensional local principal component analysis algorithm [41] to obtain $C_1, C_2, \cdots, C_L$ of all the non-coincident x^i in the non-dominated solution set of $P(t)$. The pixels are thereafter clustered into several categories,

and the algorithm works as follows.

Step 1: Randomly select L pixels, i.e., $x^{i_1}, x^{i_2}, \cdots, x^{i_L}$, from all the non-coincident x^i of the non-dominated solution set of $P(t)$; and generate L lines, i.e., $l_1, l_2, \cdots, l_L$, based on the L pixels, with each passing through one pixel.

Step 2: Cluster all the non-coincident x^i of the non-dominated solution set of $P(t)$ into L categories, i.e., $C_1, C_2, \cdots, C_L$, based on the distance between each x^i and each of the L lines, with $C_j, j = 1, 2, \cdots, L$, being

$$C_j = \{x^i | D(x^i, l_j) \leq D(x^i, l_k), \forall k \neq j, k \in \{1, 2, \cdots, L\}\}.$$

where $D(x^i, l_j)$ is the distance between x^i and l_j.

Step 3: Let the mean value and the covariance of the pixel positions of C_j be c_j and V_j, respectively, which are given as follows.

$$c_j = \frac{1}{|C_j|} \sum_{x^i \in C_j} x^i,$$

$$V_j = \frac{1}{|C_j| - 1} \sum_{x^i \in C_j} (x^i - c_j)(x^i - c_j)^T.$$

where $|C_j|$ is the number of pixels in C_j. Further l_j has the following expression

$$l_j = \{l \in R^2 | l = c_j + \theta u_j, \theta \in R\}$$

where u_j is the feature vector corresponding to the largest eigenvalue of V_j.

Step 4: If $C_j, j = 1, 2, \cdots, L$, does not change, end the algorithm; otherwise, go to Step 2.

With the above algorithm, we can obtain the pixels of the hand by sampling around $l_1, l_2, \cdots, l_L$. Second, we consider the relationship between l_j and C_j to obtain the pixels of the hand. The projection of x^i in C_j on l_j is denoted as x_l^i. According to [41], we define the following parameter, d^i, based on the difference vector between c_j and x_l^i:

$$d^i = \left(x^i - c_j\right)^T u_j$$

According to the expression of d^i, the absolute value of d^i is equal to the distance between x_l^i and c_j. Further, we obtain the minimal and the maximal values of d^i

for each pixel in C_j, i.e., $\underline{d_j}$ and $\overline{d_j}$, respectively. So the projection of each pixel in C_j on l_j is on segment φ_j. According to [41], φ_j has the following expression

$$\varphi_j = c_j + \theta u_j,\ \underline{d_j} \leq \theta \leq \overline{d_j}.$$

All the non-coincident pixels of the non-dominated solution set of $P(t)$ distribute around $\varphi_1, \varphi_2, ..., \varphi_L$. We obtain the pixels of the hand by sampling $\varphi_1, \varphi_2, ..., \varphi_L$. They form the probability model of the components of x.

We lengthen $\varphi_1, \varphi_2, ..., \varphi_L$ to expand the search scope of the probability model, and obtain the probability model, ψ_j, $j = 1, 2, \cdots, L$, as follows.

$$\psi_j = c_j + \theta' u_j,\ \underline{d_j} - 0.25\left(\overline{d_j} - \underline{d_j}\right) \leq \theta' \leq \overline{d_j} + 0.25\left(\overline{d_j} - \underline{d_j}\right).$$

We call it the probability model of candidate component segments.

The value of L is mainly determined by the size of the hand region. If the value is small the accuracy of the probability model will be low, which is not beneficial to the evolution of the population. Otherwise, the computational complexity of the evolution will be high.

4.6. Sampling based on segments

We can obtain a candidate, x, by sampling the probability model, ψ_j. Component $x^i, i = 1, 2, \cdots, n$ of the candidate should meet the following conditions: (1) the probability of generating x^i by ψ_j, is equal to the ratio of the number of pixels in C_j to the total number of pixels, (2) x^i distributes evenly in the direction of ψ_j, and (3) x^i distributes around ψ_j. In addition, the pixels of different candidates are different from each other, according to the definition of x in model (1).

We set $x = \phi$. According to $|C_j|$, the probability of generating a pixel by ψ_j is $p_j = \frac{|C_j|}{\sum_{k=1}^{L} |C_k|}$, $j = 1, 2, \cdots, L$. In addition, we can also obtain the second largest eigenvalue of V_j, i.e., λ_j, which reflects the deviation between the pixels in C_j and their centers.

We employ the following steps to generate candidate x which meets the above conditions.

Step 1: Select ψ_j from L segments with the probability of p_j.

Step 2: Generate point (x'_i, y'_i) by evenly sampling ψ_j. Obtain the disturbance of (x'_i, y'_i), i.e., $(\Delta x'_i, \Delta y'_i)$, by Gaussian sampling (the sampling variance is λ_j). So $x^i = (x'_i + \Delta x'_i, y'_i + \Delta y'_i)$ is the obtained pixel by sampling ψ_j.

Step 3: If x^i is beyond the allowed scope, regenerate x^i by the boundary value of the scope.

Step 4: If $x^i \notin x$, set $x = x \cup \{x^i\}$.

Step 5: If $|x| = n$, end the algorithm; otherwise, go to Step 1.

The above steps are the process of obtaining a candidate by sampling. We execute the above process N times and obtain N candidates. Thus, a temporary population, $Q(t)$, is obtained and its size is N.

The 1-dimensional local principal component analysis is used to build an EDA to solve a continuous two-objective optimization problem [41]. Model (1) is a discrete multi-objective optimization problem. All optimal solutions of a continuous bi-objective optimization problem are around several segments, and all pixels of each optimal solution of model (1) are also around several segments. Therefore, we will use the 1-dimensional local principal component analysis to build an EDA to solve model (1). The probability model of candidate component segments is built for each component of a candidate whereas the probability model of candidates in reference [41] is built for each candidate. The rest of the probability model of candidate component segments is the same as that in [41]. Apart from in our sampling method of pixels, each pixel of a candidate is different from the others, the rest of our sampling method of pixels is the same as that in [41].

4.7. Selection

The population at the t-th generation is $P(t)$, and the temporary population obtained by the above method is $Q(t)$. In order to produce the offspring population, i.e., $P(t+1)$, we first combine $P(t)$ and $Q(t)$. Then, we obtain the offspring population based on the fast non-dominated sorting selection [12]. Interested

readers are suggested to refer to [12] for details of the fast non-dominated sorting selection.

5. Experiments

In this section, the effectiveness of the proposed model and algorithm is verified by a series of experiments. The problems to be verified are first proposed. Following that, the hand images used in the experiments and the compared algorithms are provided. Next, the performance index for the comparison among algorithms is given. Having described the experimental process, the experimental results are presented and analyzed.

5.1. Problems to be verified

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

(1) Can the proposed method select hand pixels from a gesture image? We investigate this by processing actual gesture images with approximate skin-color regions.

(2) Is the proposed algorithm better than the compared algorithms? We compare the proposed algorithm with three state-of-the-art algorithms, and demonstrate the effectiveness of the proposed algorithm by showing that it has a higher accuracy rate.

5.2. Benchmark gesture images

In this paper, we use gesture images from the standard American Sign Language image database [5] to conduct the experiments. The image database contains 2515 gesture images of five people in various light conditions. Thus, the gesture images are representative.

We sort the gesture images by their file names, and use their serial numbers to rename them. In this way, their file names are changed from hand1-0-bot-seg-1-cropped.png, hand1-0-bot-seg-2-cropped.png, $\cdots$, hand5-z-dif-seg-5-cropped.png into 1.png, 2.png, $\cdots$, 2515.png.

Gesture images 133.png, 1024.png, 1838.png, 2063.png, and 2506.png from 5 people are shown in Fig. 1 (1) - (5) in order.

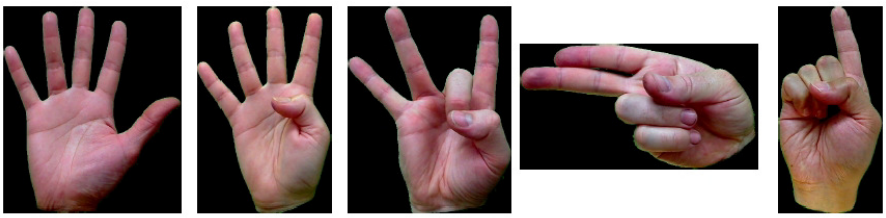

Fig. 1. Gesture images.

5.3. Three compared algorithms

We compare EDA of candidate component intervals, EDA of candidate component segments, and NSGA-II [12] with our proposed EDA.

Firstly, EDA of candidate component intervals works as follows.

Step 1: Set $t = 0$, and randomly generate an initial population $P(0)$.

Step 2: If the stopping condition is met, go to Step 6.

Step 3: Build the probability model of candidate component intervals by the non-dominated solution set of $P(t)$.

Step 4: Sample and obtain a temporary population, $Q(t)$, by the probability model of intervals.

Step 5: Select individuals and obtain the offspring population, $P(t+1)$, from $P(t) \cup Q(t)$ by the fast non-dominated sorting [12]. Set $t = t+1$, and go to Step 2.

Step 6: Return the non-dominated solutions of $P(t)$ and end the algorithm.

Note that the above algorithm only employs candidate component intervals to generate the temporary population, $Q(t)$, which is different from that of our proposed EDA. Besides the above difference, the rest of the above algorithm is the same as the proposed EDA.

Secondly, EDA of candidate component segments works as follows.

Step 1: Set $t = 0$, and randomly generate an initial population $P(0)$.

Step 2: If the stopping condition is met, go to Step 6.

Step 3: Build the probability model of candidate component segments by the non-dominated solution set of $P(t)$.

Step 4: Sample and obtain a temporary population, $Q(t)$, by the probability model of segments.

Step 5: Select individuals and obtain the offspring population, $P(t+1)$, from $P(t) \cup Q(t)$ by the fast non-dominated sorting [12]. Set $t = t+1$, and go to Step 2.

Step 6: Return the non-dominated solutions of $P(t)$ and end the algorithm.

Note that the above algorithm only employs candidate component segments to generate the temporary population, $Q(t)$, which is different from that of our proposed EDA. Apart from the above difference, the rest is the same as our proposed EDA.

Thirdly, NSGA-II utilizes the crossover and mutation operators to obtain a temporary population, Q. Since the feasible range $[\underline{x}, \overline{x}]$ or $[\underline{y}, \overline{y}]$ of the decision component x_i or y_i of model (1) is a continuous interval, 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 $Q(t)$. However, the method of generating the temporary population, $Q(t)$, in NSGA-II is different from the proposed EDA.

All the above three algorithms can solve model (1), and all employ the fast non-dominated sorting selection to obtain the offspring population, $P(t+1)$. As a result, we can demonstrate the effectiveness of our proposed EDA by comparing it with these three algorithms.

Although the proposed EDA consists of two stages and all the above compared algorithms have only one stage, we have a unified baseline during comparison. Since the computational complexity of an evolutionary algorithm is mainly determined by the evaluations of individuals, which is equal to the population size multiplied by the number of generations, we should guarantee that each compared algorithm and the proposed one have the same evaluations of individuals when comparing. To fulfill this task, we set the population sizes of

all the algorithms to be the same. Additionally, the maximal number of generations of the proposed EDA, which is the sum of the number of generations in the first stage and that in the second stage, is equal to that of each compared algorithm. In this way, the comparison of the four algorithms is guaranteed to be reasonable and fair.

5.4. The performance index

We use the proposed method to process a gesture image, and obtain the optimal solution set and the corresponding pixels. Based on the ranges of Cb and Cr values of the above pixels, we can obtain the ranges of Cb and Cr values of the hand region of the gesture image, and segment the hand region from the gesture image. If the pixel set of the optimal solution set contains non-hand pixels, the obtained range of Cb or Cr values of the hand region will be inaccurate, and therefore the hand region cannot be segmented. As a result, whether the pixel set of the optimal solution set contains non-hand pixels or not is crucial when evaluating the capability of the proposed algorithm.

We solve model (1) of a gesture image once and select the pixels of the optimal solutions. Each selected pixel has its Cb and Cr values, and its CbCr vector is a point in the CbCr plane. All the above points in the CbCr plane form a set of CbCr vectors, and each CbCr vector expresses the Cb and Cr values of a selected pixel of the optimal solutions. If the selected pixels are the majority of pixels in the hand region, the set of CbCr vectors of the optimal solutions is said to be similar to that of the hand region in the gesture image. The CbCr vectors of the hand region are different from those of the rest of the gesture image, and therefore we can segment the hand region from the image by its CbCr vectors. Following the above, we can fulfill gesture segmentation by the Cb and Cr values of the selected pixels.

We run the proposed EDA once for solving the model and obtain a set of Pareto optimal solutions. Following that, we obtain the pixels contained by the above Pareto optimal solutions. If each obtained pixel is a hand pixel, the set of Pareto optimal solutions is regarded as correct, otherwise, it is incorrect.

Similarly, we can judge the correctness of the set of Pareto optimal solutions obtained by NSGA-II. Therefore, we can compare the two sets of Pareto optimal solutions according to whether they are correct or not. If both are correct, or neither is correct, we think that they have no difference. If only one is correct, we consider the correct one as superior, and the other as inferior.

Due to the randomness of evolutionary algorithms, the pixels of the optimal solution set obtained by the proposed method might contain non-hand pixels. To overcome this, we utilize the proposed algorithm to independently solve model (1) several times, and obtain several optimal solution sets. If only a minority of the optimal solution sets contain non-hand pixels, we can take pixels with more times of repeated selection as the hand pixels, and eliminate the non-hand pixels. Therefore, whether there are a minority of sets which contain non-hand pixels among the optimal solution sets is of importance. Furthermore, we can obtain the ratio of the number of the optimal solution sets which do not contain non-hand pixels to the total number of optimal solution sets, and define it as the accuracy rate. According to the above analysis, the accuracy rate can be taken as the performance index of evaluating the capability of the proposed algorithm.

We use each of the three compared algorithms to independently solve model (1) several times, and obtain several optimal solution sets. Their accuracy rates can also be taken as the performance indexes of evaluating the capability of the three compared algorithms. Thus, we can demonstrate the effectiveness of the proposed algorithm by comparing these accuracy rates.

We run one of the above four algorithms once to solve model (1) and obtain a set of optimal solutions. If the obtained optimal solutions are the true optimal solutions of model (1), their pixels are in the hand region. Due to the randomness of these algorithms, we run each one several times and obtain several sets of optimal solutions. A pixel can be selected several times by the above sets, and the pixel with more selections has a lower error in general. Therefore we should further select the pixels with more selections from the selected pixels as the hand pixels. If all the selected pixels of the above sets are hand pixels, we need to further select them. If only a part of selected pixels are hand pixels, it

is difficult to further select hand pixels and remove non-hand pixels according to the selections of each pixel. This is because once a non-hand pixel is further selected, there will be a non-hand region in the obtained region, and therefore the obtained hand region is inaccurate. As a result, we define the accuracy rate of the optimal solution sets to evaluate them. The accuracy rate can be used to evaluate the method of gesture segmentation and compare the above four algorithms.

5.5. The experimental process

We fill the outside part of each hand region of the image database with a background image to study the accuracy rate of the proposed algorithm. The values of Cb and Cr of each background image are 60 and 210, respectively, which approach those of Cb and Cr of a skin color. The size of each gesture image is 800×800, and the hand region is in the middle of the image with a filled background around it.

For the proposed method, we first preprocess the gesture image, and obtain the values of Cb and Cr of each pixel. Following that, we build model (1). Finally, we use the two-phase EDA to solve model (1), and obtain one or more non-dominated solutions.

The decision space of model (1) is composed of all the pixels of the gesture image. The number of coordinates of the selected pixels is the dimension of the decision variable. If the dimension is too large, both the computation complexity and the time consumption for solving the model will increase. Therefore, it is necessary to compress the gesture images. By sampling the pixels at each interval of 8 rows and 8 columns, we can reduce the size of each gesture image. The size of each compressed gesture image is 100×100. We express each compressed gesture image with YCbCr, and obtain the values of Cb and Cr of each pixel.

According to model (1), the decision variable should represent several pixels. Thus, a vector of 100 dimensions is utilized as the decision variable in the following experiments, with decision variable x representing 50 pixels. We can

obtain their values of Cb and Cr, and obtain their $f_1(x)$ and $f_2(x)$.

Considering the efficiency of evolutionary computation, the population size is 50, the maximal number of generations is 100, $M = 12$, $L = 10$, and $T = 50$. Due to the randomness of the optimal solution set, we independently run each algorithm 20 times, and obtain 20 optimal solution sets, that is, 20 pixel sets. We compare each of the 20 pixel sets with that of the actual hand pixels to obtain the number of optimal solution sets not containing non-hand pixels, and calculate the accuracy rates of 20 runs.

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

5.6. Experimental results and analysis

Each of the hand regions of the image database has a green edge, with its values of Cr less than 124, and therefore is removed from each hand region. In this way, we can obtain 2515 accurate hand regions. We fill the accurate hand regions with background images. Having removed green edges from the hand regions shown in Fig. 1 and filled them with background images, we obtain Fig. 2(1)-(5). The values of Cb and Cr of each pixel of Fig. 2(1)-(5) are shown in (1)-(5) of Fig. 2(a) and 2(b) in order.

Fig. 2 (a) (1)-(5) show the distribution of Cb values of Fig. 2 (1)-(5) in order. Fig. 2 (b) (1)-(5) show the distribution of Cr values of Fig. 2 (1)-(5) in order. In each image of Fig. 2 (a) (1)-(5), each vertical coordinate represents the number of rows of a pixel in each matrix image of Fig. 2 (1)-(5); each horizontal coordinate means the number of columns of a pixel in each matrix image of Fig. 2 (1)-(5). They are similar to each vertical or horizontal coordinate in each image of Fig. 2 (b) (1)-(5). There is a color bar in the right of each image of Fig. 2 (a) (1)-(5). The color bar refers to the Cb value of each pixel in each image of Fig. 2(1)-(5). The color in the bar changes from dark red to dark blue when its Cb value changes from large to small. They are similar to the color bar in each image of Fig. 2 (b) (1)-(5), except that the color bar represents the Cr values.

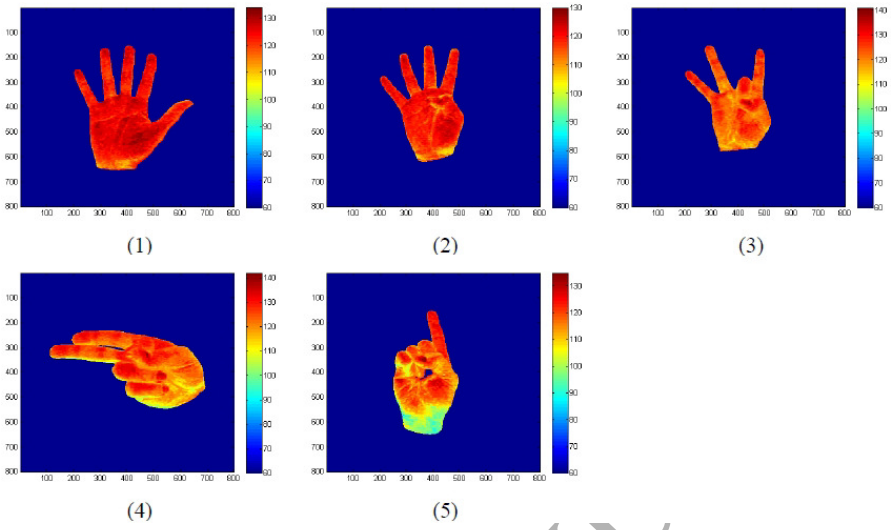

Fig. 2(a). Distribution of Cb values.

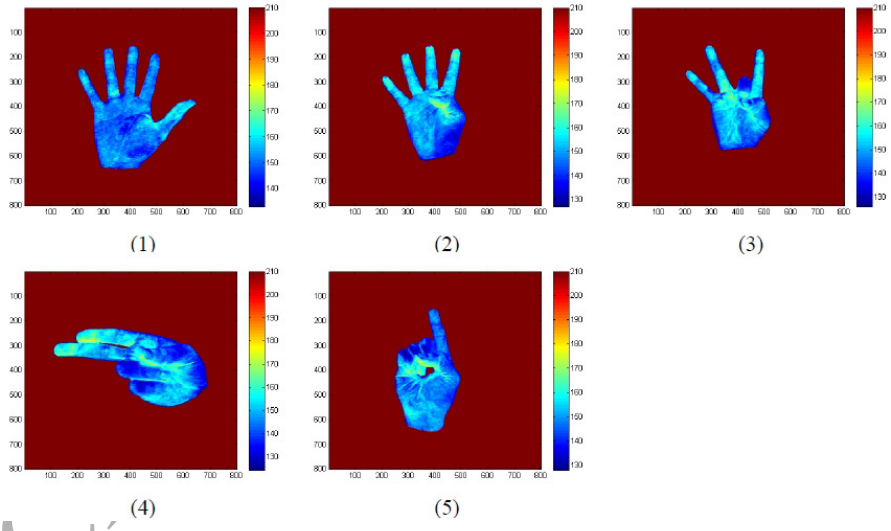

Fig. 2(b). Distribution of Cr values.

We run the proposed algorithm 20 times for each of the 2515 preprocessed images, and obtain 20 optimal pixel sets. The obtained pixel sets of Fig. 2 (1) are shown in Fig. 3 (1). In Fig. 3 (1), the figures from left to right in order in the first row are the results obtained from the 1st to the 5th run. The ones from left to right in order in the second row are the results obtained from

the 6th to the 10th run, and similarly for the others. By using the proposed method, we can also obtain the pixel sets of Fig. 2 (2)-(5) shown in Fig. 3 (2)-(5) respectively.

The 20 images in Fig. 3 (1) show 20 optimal pixel sets obtained from Fig. 2 (1). They are similar to Fig. 3 (2)-(5), except that the pixel sets are obtained from Fig. 2 (2)-(5) in order. In each image of Fig. 3 (1), each red dot means the pixel obtained from Fig. 2 (1). The vertical coordinate of the above red dot represents its number of rows in the matrix of Fig. 2 (1), and the horizontal coordinate of the above red dot refers to its number of columns in the matrix of Fig. 2 (1). They are similar to the red dot in each image of Fig. 3 (2)-(5).

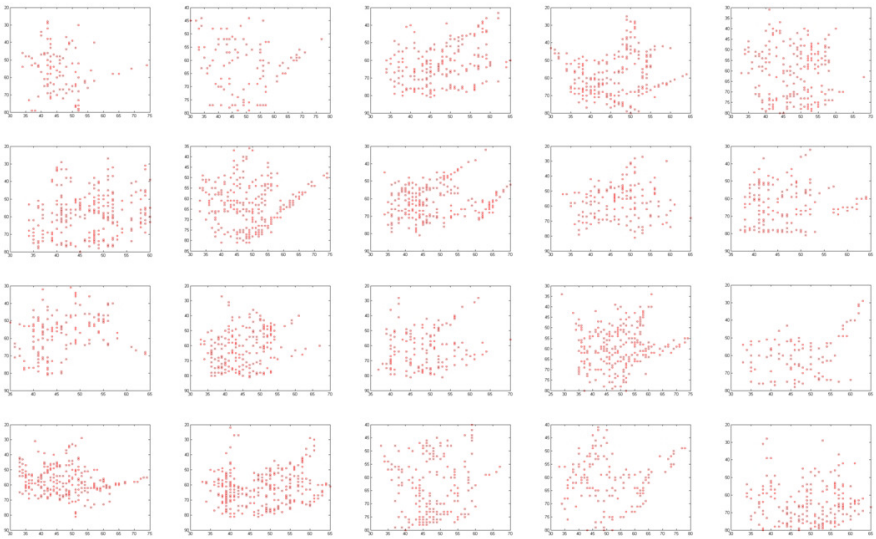

Fig. 3(1). The pixel sets of Fig. 2 (1) obtained by the proposed method.

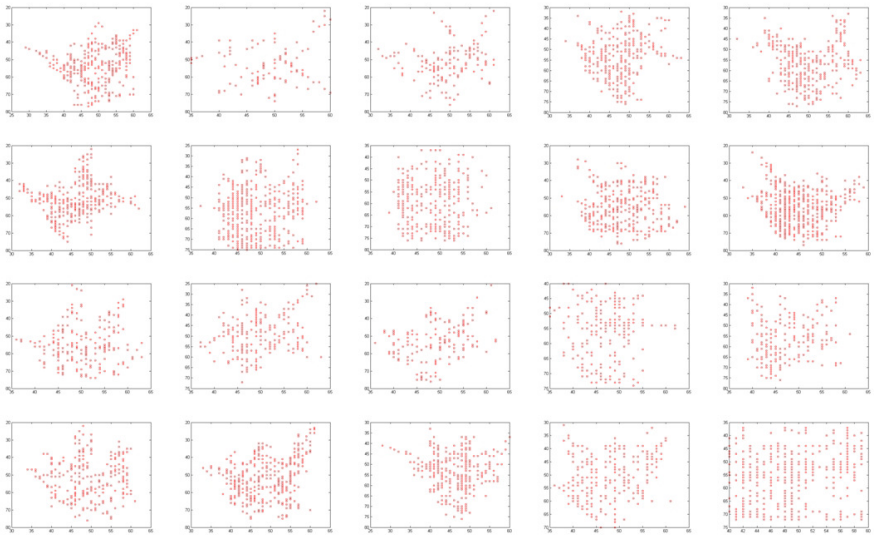

Fig. 3(2). The pixel sets of Fig. 2 (2) obtained by the proposed method.

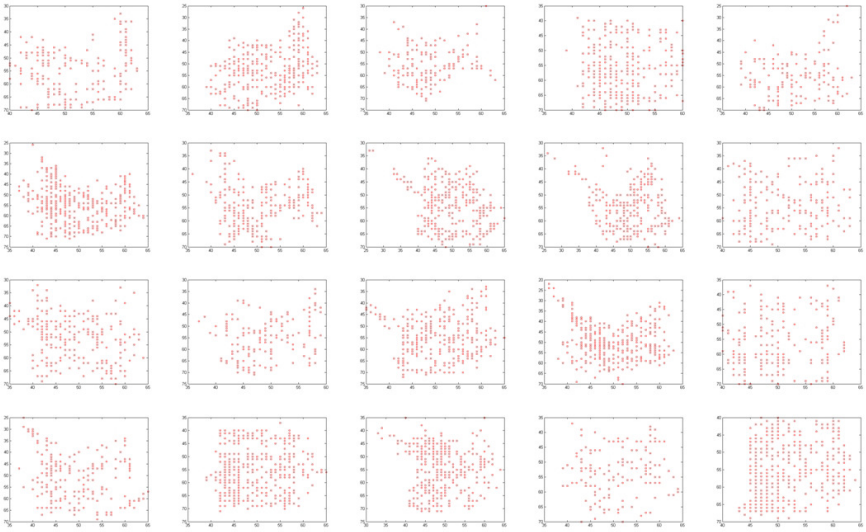

Fig. 3(3). The pixel sets of Fig. 2 (3) obtained by the proposed method.

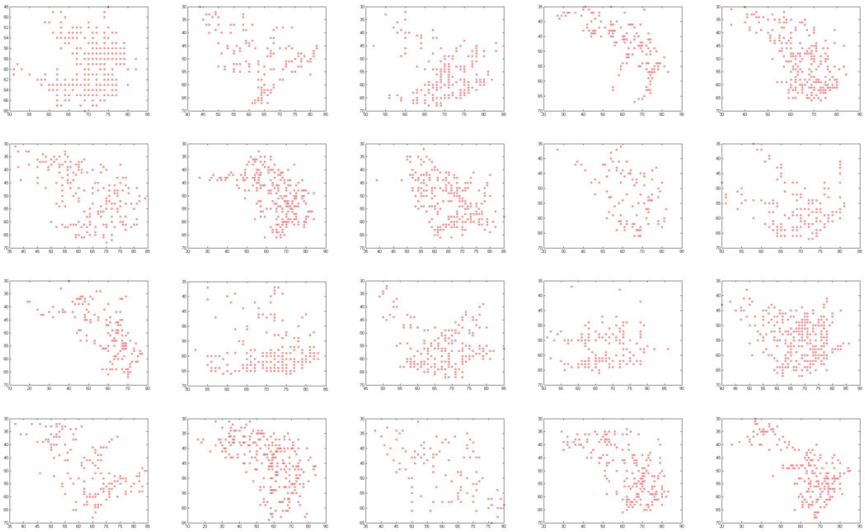

Fig. 3(4). The pixel sets of Fig. 2 (4) obtained by the proposed method.

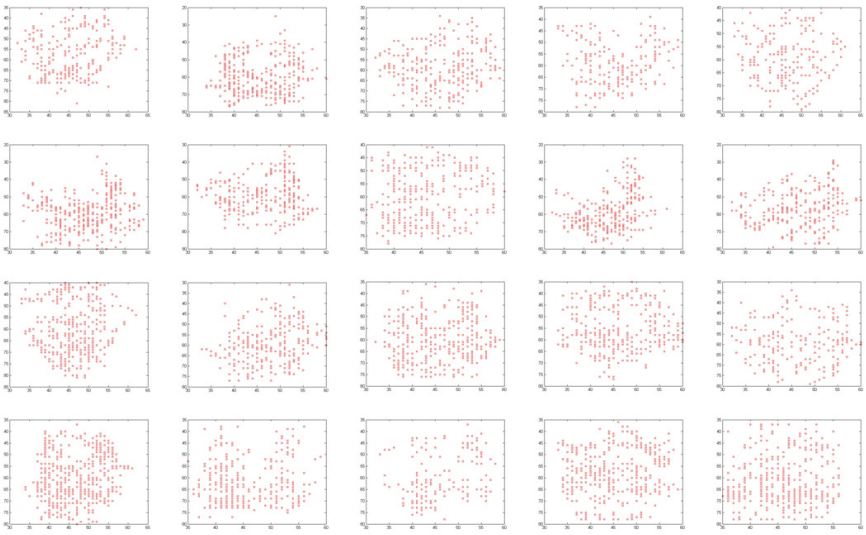

Fig. 3(5). The pixel sets of Fig. 2 (5) obtained by the proposed method.

The 5 accuracy rates of Fig. 2(1)-(5) obtained by the proposed algorithm are all 100%. The accuracy rates of the 2515 preprocessed images obtained by the proposed algorithm are shown in Fig. 4(1), and their average value is 92.24%. This indicates that the proposed algorithm can accurately select hand

pixels from a gesture image. In addition, the proposed method spends 3-4s in processing an image.

We apply each of the three compared algorithms to each of the 2515 preprocessed images 20 times, and obtain 20 optimal pixel sets. For each compared algorithm, the same parameter settings are adopted, that is, the population size is 50, and the maximal number of generations is 100. In addition, the same environment configuration is adopted. Based on these, we obtain the accuracy rate of each run. Furthermore, in EDA of candidate component intervals, $M = 12$. In EDA of candidate component segments, $L = 10$. In NSGA-II, the crossover and mutation probabilities are 0.9 and 0.1, respectively. In addition, both the crossover and the mutation distribution indexes are 0.2. The 2515 accuracy rates of EDA of candidate component intervals are shown in Fig. 4(2), and their average value is 25.11%. The 2515 accuracy rates of EDA of candidate component segments are shown in Fig. 4(3), and their average value is 7.55%. All the 2515 accuracy rates of NSGA-II are 0. This indicates that the proposed algorithm exceeds the three compared counterparts.

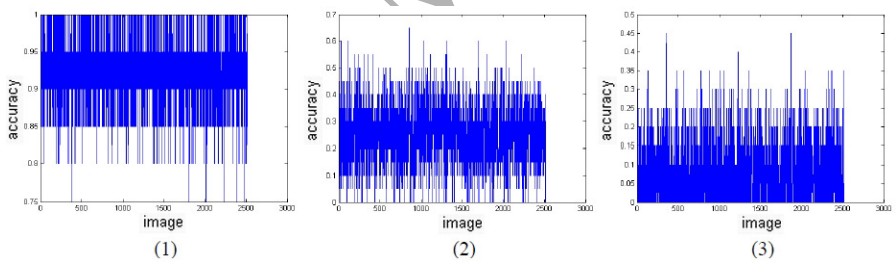

Fig. 4. The accuracy rates of different algorithms.

Fig. 4(1) shows the accuracy rates of the 2515 gesture images obtained by the proposed algorithm. In this figure, the horizontal coordinate, image, is the file name of each gesture image, and the vertical coordinate, accuracy, is the accuracy rate of the gesture image. Fig. 4(2) and 4(3) show the accuracy rates obtained by EDA of candidate component intervals and EDA of candidate component segments, respectively, and their coordinates have the same meanings as those of Fig. 4(1).

The proposed algorithm employs candidate component intervals and candidate component segments to build the probability models and to generate the temporary population. This is in accordance with the correlation among the positions of hand pixels, whereas EDA of candidate component intervals only causes pixels of the temporary population to distribute in a rectangular region, and EDA of candidate component segments only causes pixels of the temporary population to distribute around several line segments. As a result, neither of them is in accordance with the correlation among the positions of hand pixels, and they are inferior to our proposed method. NSGA-II does not take advantage of the correlation among the positions of hand pixels to generate a temporary population, and therefore is inferior compared with our method.

From the above experimental results and analysis, we can obtain the following conclusions: the proposed method can select hand pixels from a gesture image with a high accuracy rate, and exceeds the other. Thus, both the proposed model and algorithm are effective.

6. Conclusions

We have formulated a multi-objective optimization model for the problem of gesture segmentation, and presented a method of solving the above model based on a two-phase estimation of distribution algorithm. When building the model, we take the positions of a series of pixels as the decision variable, and the differences between the colors of pixels and those of a hand as the objective functions. The proposed method of gesture segmentation based on a two-phase estimation of distribution algorithm is based on the correlation among the positions of pixels. The method divides the process of solving the problem based on evolutionary optimization into two phases, and adopts different estimation of distribution algorithms in different phases. In the first phase, we formulate the probability model of candidates as a number of intervals given the fact that the positions of hand pixels distribute in several intervals. In the second phase, we build the probability model of candidates as a series of segments since the

positions of hand pixels further distribute around curves. We obtain a series of hand pixels based on sampling by the above probability models. The experimental results of actual gesture images demonstrate that the proposed method can select hand pixels from a gesture image and exceeds the three compared algorithms.

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

Acknowledgments

This work was jointly supported by National Natural Science Foundation of China, Grant No. 61403155, 61473299, and 61375067, and Graduate Student Research and Innovation Program of Jiangsu Province of China, Grant No. 2014KYLX1388. Thanks to Dr. Edward C. Mignot and Dr. Richard Mealing for linguistic advice.

References

[1] I. Alberto, C.A.C. Coello, P.M. Mateo, A comparative study of variation operators used for evolutionary multi-objective optimization, Information Sciences 273 (8) (2014) 33-48.

[2] W.A. Albukhanajer, J.A. Briffa, Y. Jin, Evolutionary multi-objective image feature extraction in the presence of noise, IEEE Transactions on Cybernetics 45 (9) (2015) 1757-1768.

[3] B.D. Avinash, D.K. Ghosh, S. Ari, Color hand gesture segmentation for images with complex background, in: Proceedings of 2013 International Conference on Circuits, Power and Computing Technologies, IEEE, 2013, pp. 1127-1131.

[4] S. Bandyopadhyay, A. Mukherjee, An algorithm for many-objective optimization with reduced objective computations: a study in differential evolution,

IEEE Transactions on Evolutionary Computation 19 (3) (2015) 400-413.

[5] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, T. Susnjak, A new 2D static hand gesture colour image dataset for ASL gestures, Res. Lett. Inf. Math. Sci. 15 (2011) 12-20.

[6] J. Ceberio, E. Irurozki, A. Mendiburu, J. Lozano, A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem, IEEE Transactions on Evolutionary Computation 18 (2) (2014) 286-300.

[7] D. Chai, K. Ngan, Face segmentation using skin color map in videophone applications, IEEE Transactions on Circuits and Systems for Video Technology 9 (4) (1999) 551-564.

[8] D.S. Chen, Z.K. Liu, A survey of skin color detection, Chinese Journal of Computers 29 (2) (2006) 194-207 (in Chinese).

[9] Z. Chen, J. Kim, J. Liang, J. Zhang, Y. Yuan, Real-time hand gesture recognition using finger segmentation, The Scientific World Journal 267872 (2014) 1-9.

[10] A.Y. Dawod, M.J. Nordin, J. Abdullah, Gesture segmentation: automatic continuous sign language technique based on adaptive contrast stretching approach, Middle East Journal of Scientific Research 24 (2) (2016) 347-352.

[11] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints, IEEE Transactions on Evolutionary Computation 18 (4) (2014) 577-601.

[12] K. Deb, A. Pratap, S. Agarwal, T. Mrystivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182-197.

[13] M. Fakhravari, M. Dadvar, Skin color segmentation in RGB color space by adaptive network based fuzzy inference system (ANFIS), Ciencia e Natura 37 (2) (2015) 264-275.

[14] 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, Flexible Services and Manufacturing Journal 27 (4)

(2015) 1-21.

[15] C. Garcia, G. Tziritas, Face detection using quantized skin color regions merging and wavelet packet analysis, IEEE Transactions on Multimedia 1 (3) (1999) 264-277.

[16] I. Giagkiozis, R.C. Purshouse, P.J. Fleming, Generalized decomposition and cross entropy methods for many-objective optimization, Information Sciences 282 (2014) 363-387.

[17] A. Gupta, A. Chaudhary, Robust skin segmentation using color space switching, Pattern Recognition and Image Analysis 26 (1) (2016) 61-68.

[18] D. Hong, L. Yang, A method of gesture segmentation based on skin color and background difference method, in: Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering, IEEE, 2013, pp. 355-357.

[19] Y. Jin, B. Sendhoff, Pareto-based multi-objective machine learning: An overview and case studies, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38 (3) (2008) 397-415.

[20] Z. Ju, Y. Wang, W. Zeng, H. Cai, H. Liu, A modified EM algorithm for hand gesture segmentation in RGB-D data, in: Proceedings of 2014 IEEE International Conference on Fuzzy Systems, IEEE, 2014, pp. 1736-1742.

[21] S.N. Karishma, V. Lathasree, Fusion of skin color detection and background subtraction for hand gesture segmentation, International Journal of Advanced Trends in Computer Science and Engineering 3 (1) (2014) 13-18.

[22] H. Karshenas, R. Santana, C. Bielza, P. Larranaga, Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables, IEEE Transactions on Evolutionary Computation 18 (4) (2014) 519-542.

[23] P. Lam, B. Abdesselam, C. Douglas, Skin segmentation using color pixel classification: analysis and comparison, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (1) (2005) 148-154.

[24] W. Li, X. He, F. Jiao, Skin region segmentation based on the average preprocessed image of multicolor face image sequence, Journal of Engineering

and Technological Sciences 46 (4) (2014) 381-393.

[25] X. Liang, H. Chen, J. Lozano, A Boltzmann-based estimation of distribution algorithm for a general resource scheduling model, IEEE Transactions on Evolutionary Computation 19 (6) (2015) 793-806.

[26] A. Licsar, T. Sziranyi, User adaptive hand gesture recognition system with interactive training, Image and Vision Computing 10 (2) (2005) 1102-1114.

[27] M.R. Mahmoodi, S.M. Sayedi, F. Karimi, Color-based skin segmentation in videos using a multi-step spatial method, Multimedia Tools and Applications 5 (2016) 1-17.

[28] S. Muelas, A. Mendiburu, A. Latorre, J.M. Pena, Distributed estimation of distribution algorithms for continuous optimization: how does the exchanged information influence their behavior? Information Sciences 268 (1) (2014) 231-254.

[29] O. Ozturk, A. Aksac, T. Ozyer, R. Alhajj, Boosting real-time recognition of hand posture and gesture for virtual mouse operations with segmentation, Applied Intelligence 43 (4) (2015) 786-801.

[30] R. Perez-Rodriguez, A. Hernandez-Aguirre, S. Jons, A continuous estimation of distribution algorithm for the online order-batching problem, International Journal of Advanced Manufacturing Technology 79 (1-4) (2015) 569-588.

[31] O. Soliman, E. Elgendi, Niching estimation of distribution algorithm based on fuzzy clustering for multi-mode resource-constrained project scheduling problems, International Journal of Scientific and Engineering Research 5 (7) (2014) 183-189.

[32] J. Sun, J.M. Garibaldi, Y. Zhang, A. Al-Shawabkeh, A multi-cycled sequential memetic computing approach for constrained optimisation, Information Sciences 340C341 (2016) 175-190.

[33] S.I. Valdez, E. Chavezconde, E.E. Hernandez, M. Ceccarelli, Structure-control design of a mechatronic system with parallelogram mechanism using an estimation of distribution algorithm, Mechanics Based Design of Structures and Machines 44 (1-2) (2016) 58-71.

[34] Z. Wan, L. Mao, G. Wang, Estimation of distribution algorithm for a

class of nonlinear bilevel programming problems, Information Sciences 256 (1) (2014) 184-196.

[35] 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 Transactions on Evolutionary Computation 20 (1) (2016) 96-109.

[36] L. Wang, C. Fang, P. Suganthan, M. Liu, Solving system-level synthesis problem by a multi-objective estimation of distribution algorithm, Expert Systems with Applications 41 (5) (2014) 2496-2513.

[37] S. Wang, L. Wang, An estimation of distribution algorithm-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem, IEEE Transactions on systems, man, and cybernetics: systems 46 (1) (2016) 139-149.

[38] D. Wu, L. Pigou, P.J. Kindermans, L.E. Nam, L. Shao, Deep dynamic neural networks for multimodal gesture segmentation and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (8) (2016) 1583-1597.

[39] K. Yang, L. Mu, D. Yang, F. Zou, L. Wang, Multiobjective memetic estimation of distribution algorithm based on an incremental tournament local searcher, The Scientific World Journal 836272 (2014) 1-21.

[40] P. Yang, K. Tang, X. Lu, Improving estimation of distribution algorithm on multimodal problems by detecting promising areas, IEEE Transactions on cybernetics 45 (8) (2015) 1438-1449.

[41] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm, IEEE Transactions on Evolutionary Computation 12 (1) (2007) 41-63.

[42] Q.Y. Zhang, J.C. Lu, M.Y. Zhang, H.X. Duan, F. Lai, Hand gesture segmentation based on mixed skin-color model and FCM algorithm, Journal of Information and Computational Science 12 (9) (2015) 3527-3536.

[43] Q.Y. Zhang, J.C. Lu, M.Y. Zhang, H.X. Duan, L. Lv, Hand gesture segmentation method based on YCbCr color space and k-means clustering, Interna-

tional Journal of Signal Processing, Image Processing and Pattern Recognition 8 (5) (2015) 105-116.

[44] 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, International Journal of Production Research 54 (4) (2016) 1039-1060.

[45] A. Zhou, J. Sun, Q. Zhang, An estimation of distribution algorithm with cheap and expensive local search methods, IEEE Transactions on Evolutionary Computation 19 (6) (2015) 807-822.

[46] S. Zhou, X. Li, H. Chen, C. Guo, Minimizing makespan in a no-wait flowshop with two batch processing machines using estimation of distribution algorithm, International Journal of Production Research 54 (16) (2016) 4919-4937.