第 30 卷第 9 期
2013 年 9 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 30 No. 9
Sep. 2013

DOI: 10.7641/CTA.2013.12223

一种基于修正差分进化的虹膜定位算法

邹德旋[1†], 王 鑫[2], 段 纳[1]

(1. 江苏师范大学 电气工程及自动化学院, 江苏 徐州 221116; 2. 沈阳建筑大学 信息与控制工程学院, 辽宁 沈阳 110168)

摘要: 提出一种用于虹膜定位的差分进化算法(modified differential evolution, MDE). MDE和原始差分进化算法(differential evolution, DE)主要有3点不同: 第一, MDE采用了基于混沌序列的尺度因子和基于均匀分布的交叉率, 这有助于提高候选解的多样性; 第二, MDE使用中心解来修正最差解的变异操作, 这有助于提高候选解的质量; 第三, MDE使用最好解来帮助受困解摆脱局部最优点. 在搜索边缘前, 两种有效的去噪方法被用来减少虹膜图像中噪声的影响. 去噪后, 再使用MDE和其他4种方法来进行虹膜定位. 在中科院(Chinese Academy of Sciences Institute of Automation, CASIA)眼图数据库中选择200幅来自不同个体的虹膜图像来验证和比较MDE及其他4种方法的效率. 实验结果表明, 与其他4种方法相比, MDE使用更少的执行时间来定位瞳孔边缘和虹膜边缘.

关键词: 修正差分进化; 虹膜定位; 混沌序列; 变异操作; 中心解; 去噪

中图分类号: TP391.41　　　**文献标识码**: A

Iris location algorithm based on modified differential evolution algorithm

ZOU De-xuan[1†], WANG Xin[2], DUAN Na[1]

(1. School of Electrical Engineering and Automation, Jiangsu Normal University, Xuzhou Jiangsu 221116, China;
2. Information and Control Engineering faculty, Shenyang Jianzhu University, Shenyang Liaoning 110168, China)

Abstract: A modified differential evolution (MDE) algorithm is proposed for iris location. The MDE and the original differential evolution algorithm are different from three aspects: First, MDE adopts the scale factor based on chaotic sequences and the crossover rate based on uniform distribution, which is helpful to improve the diversity of candidate solutions. Second, MDE utilizes the center solution to modify the mutation operation of the worst solution, which is beneficial in improving the quality of candidate solutions. Third, MDE uses the best solution to help the trapped solutions to escape from local optima. Before searching boundaries, we use two kinds of efficient denoising methods to reduce the effects of noises on iris edge images. After denoising, the proposed MDE and the other four methods are applied to iris location. Some 200 iris images of different individuals are chosen from the Chinese Academy of Sciences Institute of Automation (CASIA) eye image database in investigating and comparing the efficiency of MDE with the other four methods. Experimental results show that MDE consumes less execution time to locate pupil and iris boundaries in comparison with the other four methods.

Key words: modified differential evolution; iris location; chaotic sequence; mutation operation; center solution; denoising

1 Introduction

Iris recognition plays a very important role in biological recognition, and this technology is helpful to preserve personal privacy, business information and state secrets etc. Iris location is one of the most difficult parts in iris recognition, because iris images are easily affected by noises including eyelashes, eyelids and reflection etc.

In recent decades, a variety of approaches have been used for iris location, and they have achieved good results. These iris location approaches include integrodifferential operator[1], Hough transform[2], active contours[3], novel integrodifferential constellation[4], cubic smoothing spline fitting[5], and so on.

In order to increase the efficiency of iris location, we proposed a modified differential evolution (MDE) algorithm. Our contributions are summarized as follows. In Section 3, MDE is introduced in detail, and it makes three improvements on the original differential evolution (DE) algorithm[6]. In Section 4, the working principle of pupil boundary detection by MDE is adequately explained. In Section 5, the working principle of iris boundary detection by MDE is briefly presented. In Section 6, 200 iris images from different individuals are chosen to investigate the efficiency of MDE for iris location. We end this paper with some conclusions in Section 7.

2 The original differential evolution algorithm (DE)

As is well known, Hough transform is virtually an enumeration method, but it is time-consuming, which will

Received 29 September 2012; revised 25 May 2013.
†Corresponding author. E-mail: zoudexuan@163.com.
This work was supported by the National Natural Science Foundation of China (No.61104222), Natural Science Fundamental Research Project of Jiangsu Colleges and Universities (No.11KJB510026), Science Fundamental Research Project of Jiangsu Normal University (No.9212812101).

strongly influence the efficiency of iris location. On the other hand, the differential evolution algorithm (DE)[6] is an efficient optimization technique, and it has a simple algorithm structure which is easy to study and master. Moreover, many improved versions of DE have been applied to a variety of complicated engineering problems including economic load dispatch optimization of power systems[7], pixel classification in remote sensing imagery[8], system identification[9], positioning of prototypes[10] and so on. Due to its excellent performance, DE can be considered for iris location. Generally speaking, DE works as follows:

Step 1 Initialization.

DE parameters are initialized in this step, and these parameters include scale factor F, crossover rate CR, population size M and the maximal iteration number K. Additionally, all candidate solutions are initialized randomly from a uniform distribution in the ranges $[\underline{x}_j, \bar{x}_j]$ $(j = 1, 2, \cdots, N)$, where, $\underline{x}_j$ and $\bar{x}_j$ are the upper bound and lower bound of the jth $(j = 1, 2, \cdots, N)$ variable, respectively, and N is the number of variables.

Step 2 Mutation.

For any trial vector v_i^{k+1}, it is generated by mutating a target vector. Usually, trial vector v_i^{k+1} is generated in terms of the following equation:

$$v_i^{k+1} = x_{i_3}^k + F(x_{i_1}^k - x_{i_2}^k). \tag{1}$$

Here, F is scale factor. i_1, i_2 and i_3 are different integers which are randomly selected from the set $\{1, 2, \cdots, M\}$.

Step 3 Crossover.

The variables of offspring vector u_i^{k+1} are the combination of parent vector x_i^k and trial vector v_i^k, thus they are calculated as follows.

$$u_{i,j}^{k+1} = \begin{cases} v_{i,j}^{k+1}, & \text{if } \mathrm{rand}(\cdot) < CR \text{ or } j = r_{1\sim N}, \\ x_{i,j}^k, & \text{otherwise}, \end{cases} \tag{2}$$

where, $\mathrm{rand}(\cdot)$ is a randomly generated number in the range $[0, 1]$, $r_{1\sim N}$ denotes a random integer in the range $[1, N]$, and $CR(CR \in [0, 1])$ represents crossover rate.

Step 4 Selection.

If $f(u_i^{k+1})$ is smaller (or better) than $f(x_i^k)$, offspring vector u_i^{k+1} is assigned to x_i^{k+1}, otherwise, parent vector x_i^k is assigned to x_i^{k+1}. Thus, the selection step can be given by the following expression:

$$x_i^{k+1} = \begin{cases} u_i^{k+1}, & \text{if } f(u_i^{k+1}) < f(x_i^k), \\ x_i^k, & \text{otherwise}. \end{cases} \tag{3}$$

Step 5 Judge stopping condition.

If the maximal iteration number (K) is reached, computation is stopped. Otherwise, Steps 3−4 are repeated.

3 A modified differential evolution algorithm

The performance of DE mainly depends on its control parameters including the mutation factor and the crossover probability[11−13], and there is no confirmable parameters which are suitable for all optimization problems. To suit DE to the problem of iris location, chaotic sequences[14−15] are adopted to adjust the mutation factor. In addition, several other improvements are also included in the modification of DE. In detail, our proposed modified differential evolution (MDE) algorithm are different from DE in three aspects as follows:

1) Setting algorithm parameters.

Tuning suitable parameters is really a difficult and problem-dependent work. Regarding the tuning of these parameters, some trials are usually made during several optimization runs, which is known as trial-and-error method. However, multiple attempts bring inconvenience, and decrease the efficiency of DE.

For the original DE, its scale factor F and crossover rate CR are set as constants, that is to say, all solutions adopt these two constants in the evolution process. Unlike DE, MDE adjusts scale factor F and crossover rate CR by using chaotic sequences[14−15] and uniform distribution, respectively, because both distributions are beneficial to increasing the diversity of candidate solutions and the randomness of searching. More specifically, F and CR are stated as follows:

$$F_i^k = \mu \times F_i^{k-1} \times (1 - F_i^{k-1}). \tag{4}$$

Where μ is control parameter, $(0 \leqslant \mu \leqslant 4)$, and k denotes the iteration number. In this paper, μ is set to 4. In addition, the initial values $F_i^0 (i = 1, 2, \cdots, M)$ are randomly generated in the range $[0, 1]$, and the values of $F_i^k (i = 1, 2, \cdots, M; k = 1, 2, \cdots, K)$ are generated in terms of Eq.(4). Chaotic sequences has many advantages such as ergodicity, stochastic properties, irregularity, and so on. Therefore, they enable MDE to escape easily from the local optimums.

$$CR^k \sim Ud(CR_{\min}, CR_{\max}). \tag{5}$$

Where $Ud(CR_{\min}, CR_{\max})$ represents uniform distribution in the range of $[CR_{\min}, CR_{\max}]$ with mean $(CR_{\min} + CR_{\max})/2$ and standard deviation $(CR_{\max} - CR_{\min})/2\sqrt{3}$.

The utilization of suitable randomness is harmless to the convergence of MDE. Furthermore, The randomness existing in scale factors and crossover rates is very necessary[13,16], because it can not only improve the diversity of the population, but also enhance the exploitation capacity of MDE.

2) Modifying the mutation operation of the worst solution.

For the original DE, all M solutions adopt the same mutation and crossover operations (As Eqs. (1) and (2)). On the other hand, Liu et al.[17] proposed a center particle swarm optimization (CPSO) algorithm, which generates a center position for the last particle by averaging the positions of the other particles. The position of center particle is a potential and promising alternative, and it often guides the search direction of the population which is beneficial to producing solutions of high quality. Inspired by this characteristic, MDE also adopts a central solution x_{center}^k at generation $k (k = 1, 2, \cdots, K)$, but it is a little different from that of CPSO. In detail, it is obtained by averaging all the other solutions except the worst solution $x_{i_{\text{worst}}}^k$. After generating x_{center}^k, it is assigned to the trial vector $v_{i_{\text{worst}}}^{k+1}$ corresponding to $x_{i_{\text{worst}}}^k$. Additionally, parameter 'step' is used to carry out small-scale searching near x_{center}^k. Thus we have

$$x^k_{\text{center}} = \frac{1}{M-1}\sum_{i\neq i_{\text{worst}}} x^k_i, \tag{6}$$

$$v^{k+1}_{i_{\text{worst}},j} = x^k_{\text{center},j} - \text{step} + 2\times\text{step}\times\text{rand}, \tag{7}$$

where 'rand' denotes a random number in $[0, 1]$, and i_{worst} represents the index of the worst solution $x^k_{i_{\text{worst}}}$. Although MDE excludes the mutation operation in the updating of trail vector $v^{k+1}_{i_{\text{worst}}}$, but it still uses crossover operation in the updating of the offspring solution $u^{k+1}_{i_{\text{worst}}}$ (As Eq.(2)).

3) Updating the trapped solutions according to the best solution.

In the early evolution process of MDE, many solutions have fast convergence rate. However, they may slow down or stagnate in the late evolution process. In this situation, they are likely to be trapped into the local optimums. To overcome this disadvantage, the best solution $x^k_{i_{\text{best}}}$ is used, and it can help the trapped solutions to get rid of the local optimums. In short, the updating formulas is given by

$$x^{k+1}_{i,j} = \begin{cases} x^{k+1}_{i,j} + \text{rand}(\cdot)\cdot(x^k_{i_{\text{best}},j} - x^{k+1}_{i,j}), & \text{if count}(i) = \text{SP},\\ x^{k+1}_{i,j}, & \text{otherwise.}\end{cases} \tag{8}$$

Where i_{best} denotes the index of the best solution $x^k_{i_{\text{best}},j}$. In addition, $\text{count}(i)$ is used to count the number of continuous stagnation for the ith solution. If it reaches the stagnation period SP, $x^{k+1}_{i,j}$ will be replaced with a random number between $x^{k+1}_{i,j}$ and $x^k_{i_{\text{best}},j}$.

According to the above specific explanations, the complete MDE procedure can be illustrated in Algorithm 1.

Algorithm 1 Procedure of MDE.

```
Begin
Set population size $M$; the maximal iteration number $K$;
$\underline{x} = (\underline{x}_1, \cdots, \underline{x}_N)$; $\bar{x} = (\bar{x}_1, \cdots, \bar{x}_N)$;
Set $CR_{\min} = 0.5; CR_{\max} = 1; P = 20$; control parameter $\mu = 4$; $\text{count}(i) = 0 (i = 1, 2, \cdots, M)$.
Generate a series of random numbers in [0,1] for $F^0_i (i = 1, 2, \cdots, M)$.
Initialize a random population $P$.
For $k = 1$ to $K$
    Find the worst solution $x^k_{i_{\text{worst}}}$ and the best solution $x^k_{i_{\text{best}}}$ in $P$.
    Calculate central solution
    $x^k_{\text{center}} = \frac{1}{M-1}\sum_{i\neq i_{\text{worst}}} x^k_i$.
    $CR^k \sim Ud(CR_{\min}, CR_{\max})$.
    For $i = 1$ to $M$
      If $i \neq i_{\text{worst}}$
        $F^k_i = \mu \times F^{k-1}_i \times (1 - F^{k-1}_i)$.
        Randomly generate three integers $i_1, i_2$ and $i_3$ in the range $[1, M]$, and $i_1 \neq i_2 \neq i_3 \neq i$.
        $v^{k+1}_i = x^k_{i_1} + F^k_i \times (x^k_{i_1} - x^k_{i_2})$.
      Else
        For $j = 1$ to $N$
          $v^{k+1}_{i_{\text{worst}},j} = x^k_{\text{center},j} - \text{step} + 2\times\text{step}\times\text{rand}$.
        End For
      End If
      Randomly generate a integer $j_{\text{rand}}$ in $[1, N]$.
      For $j = 1$ to $N$
        If rand $< CR^i$ or $j = j_{\text{rand}}$
          $u^{k+1}_{i,j} = v^{k+1}_{i,j}$.
        Else if
          $u^{k+1}_{i,j} = x^k_{i,j}$.
        End If
      End For
      If $f(u^{k+1}_i) < f(x^k_i)$
        $x^{k+1}_i = u^{k+1}_i$; $\text{count}(i) = 0$.
      Else
        $x^{k+1}_i = x^k_i$; $\text{count}(i) = \text{count}(i) + 1$.
      End If
      If $\text{count}(i)$=SP
        For $j = 1$ to $N$
          $x^{k+1}_{i,j} + \text{rand}(\cdot) \times (x^k_{i_{\text{best}},j} - x^{k+1}_{i,j})$.
        End For
        $\text{count}(i) = 0$.
      End If
    End For
End For
End
```

4 Detecting pupil boundary by MDE

Iris location is composed of pupil boundary detection and iris boundary detection, and MDE is used to find both boundaries in this paper. However, the image preprocessing approaches of both detections are different before using MDE. More specifically, the procedure of pupil boundary detection is illustrated as follows:

4.1 Finding a pixel in pupil

All iris images used are from the Chinese Academy of Sciences Institute of Automation (CASIA) eye image database[18]. According to the distribution characteristics of human eyes (As Fig.1), the gray level intensities of pupil are obviously lower than those of iris and scleral. Therefore, we can separate pupil from iris and scleral by binarizing iris image, and the binary image is shown in Fig.2.

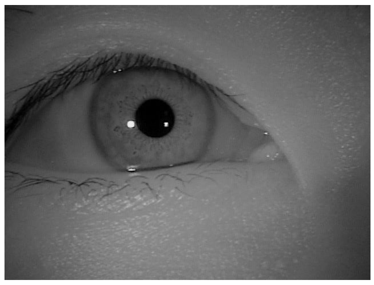
Fig. 1 Original iris image

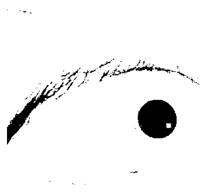
Fig. 2 The binary iris image

Unfortunately, the eyelash remains in the binary image, because the gray level intensities of eyelash are comparable to those of pupil. By observing the shape of eyelash and pupil, we can find that the size of pupil is larger than that of eyelash vertically. Therefore, we can find a pixel in pupil by detecting the longest line segment vertically in the binary image. The detailed steps can be stated in Algorithm 2.

Algorithm 2 Procedure of detecting the line segment.

```
Begin
Initialize the maximal length $l_{\max} = 0$
Set $x_{\rm start} = 3 \times {\rm rows}/20 + 1$; $x_{\rm end} = {\rm rows} - {\rm rows}/20$;
$y_{\rm start} = {\rm cols}/8 + 1$; $y_{\rm end} = {\rm cols} - {\rm cols}/8$.
For $j = y_{\rm start}$ to $y_{\rm end}$
    count $= 0$.
    For $i = x_{\rm start}$ to $x_{\rm end}$
        If $I_{\rm bin}(i, j) = 0$
            count $= count + 1$, $i_{\rm end} = i$.
        Else
            If count $> l_{\max}$
                $l_{\max} =$ count; $b_{\rm end} = b_{\rm start} = j$;
                $a_{\rm end} = i_{\rm end}$; $a_{\rm start} = i_{\rm end} -$ count $+ 1$.
            End If
            count $= 0$.
        End If
    End For
End For
End
```

Where 'rows' and 'cols' are the numbers of rows and columns of an iris image, respectively; $l_{\max}$ denotes the length of the longest black line segment vertically. $I_{\rm bin}$ denotes a binary iris image, and $I_{\rm bin}(i, j)$ denotes its pixel located in the ith row and the jth column. $(a_{\rm start}, b_{\rm start})$ and $(a_{\rm end}, b_{\rm end})$ are the coordinates of the top pixel and the bottom pixel of the detected line segment. The human eyes are usually captured in or near the center of a image, thus, we cut the binary image so as to reduce the searching region of the longest black line segment vertically. Specifically, the searching region is given by $\{(i, j)|x_{\rm start} \leqslant i \leqslant x_{\rm end}, y_{\rm start} \leqslant j \leqslant y_{\rm end}\}$. By using the method in Algorithm 2, we can detect the required line segment l which is shown in Fig.3.

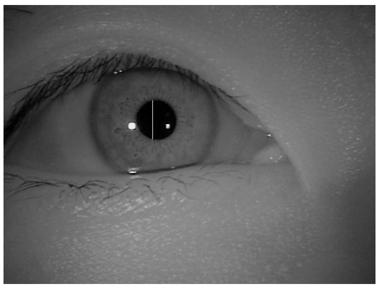

Fig. 3 The detected line segment in pupil

After detecting the line segment in pupil, we can determine the midpoint $P_{\rm mid}$ of line segment l. More specifically, suppose the coordinate of $P_{\rm mid}$ is $(x_{\rm mid}, y_{\rm mid})$, then we have $x_{\rm mid} = (a_{\rm start} + a_{\rm end})/2$ and $y_{\rm mid} = b_{\rm start}$.

4.2 Extracting edge and denoising

Canny operator[19] is a useful tool of edge extraction; therefore, we use this operator to extract the edge of iris image which is shown in Fig.4.

It is clear from Fig.4 that there are several noises in the edge image, and they are eyelash, eyelid, and so on. In order to overcome these noises, we determine a denoising region according to the length $(l_{\max})$ and the midpoint $(P_{\rm mid}(x_{\rm mid}, y_{\rm mid}))$ of line segment l. In other words, the preserved region used for iris inner boundary detection is stated by the following formulas:

$$\begin{aligned}&\{(x, y)|x_{\rm mid} - L \leqslant x \leqslant x_{\rm mid} + L,\\ &y_{\rm mid} - L \leqslant y \leqslant y_{\rm mid} + L\}.\end{aligned} \tag{9}$$

Where $L = 0.5l_{\max} + 0.2l_{\max} = 0.7l_{\max}$. As can be seen from Fig.3, the length $(l_{\max})$ of line segment l is close to the diameter of iris inner circle (pupil boundary). Roughly speaking, the region $\{(x, y)|x_{\rm mid} - 0.5l_{\max} \leqslant x \leqslant x_{\rm mid} + 0.5l_{\max}, y_{\rm mid} - 0.5l_{\max} \leqslant y \leqslant y_{\rm mid} + 0.5l_{\max}\}$ can cover almost the pupil boundary. Nevertheless, this approximation may miss out some important pixels of pupil boundary. In order to avoiding this situation, we enlarge this region by increasing L from $0.5l_{\max}$ to $0.7l_{\max}$. Based on this enlargement, we can obtain the preserved region which is shown in Fig.5.

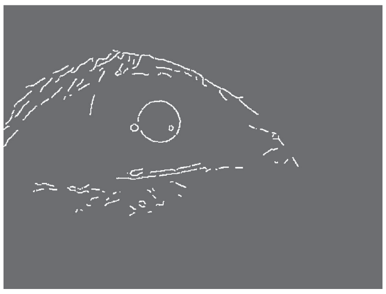

Fig. 4 Edge image based on canny operator

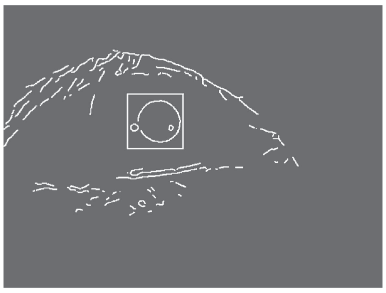

Fig. 5 The square region used to perform pupil boundary detection

As can be seen from Fig.5, the square region can cover the complete pupil boundary. Moreover, most noises can be overcome by eliminating the pixels outside this square region. According to the above denoising method, we can obtain the denoised edge image which is shown in Fig.6.

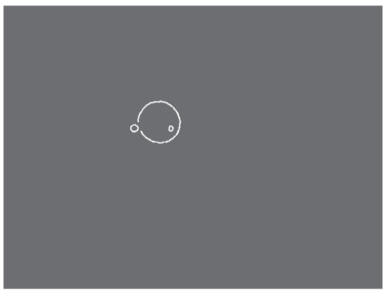

Fig. 6 The denoised edge image for pupil boundary detection

4.3 Using MDE to detect pupil boundary

After denoising the edge image, we will use MDE to find the optimal inner circle which can fit the pupil boundary. Most importantly, MDE takes the function f related to the accumulator array of Hough transform as its objective function. In detail, the accumulator array is given by

$$H(x_{\mathrm{c}}, y_{\mathrm{c}}, r) = \sum_{j=1}^{n} h(x_j, y_j, x_{\mathrm{c}}, y_{\mathrm{c}}, r), \tag{10}$$

where

$$h(x_j, y_j, x_{\mathrm{c}}, y_{\mathrm{c}}, r) = \begin{cases} 1, \text{ if } (x_j - x_{\mathrm{c}})^2 + (y_j - y_{\mathrm{c}})^2 - r^2 = 0, \\ 0, \text{ otherwise.} \end{cases} \tag{11}$$

Where r denotes the radius a circle, and $(x_{\mathrm{c}}, y_{\mathrm{c}})$ represents its center coordinate. According to Eqs.(10) and (11), the parameter combination $(x_{\mathrm{c}}, y_{\mathrm{c}}, r)$ which maximizes $H(x_{\mathrm{c}}, y_{\mathrm{c}}, r)$ is commonly chosen as the parameters of the inner circle of iris. It should be noticed that Hough transform is actually a maximization problem. In order to use MDE, we only need transform this maximization problem to a minimization problem, and the objective function f is given by

$$f = \min_{x_{\mathrm{c}}, y_{\mathrm{c}}, r} -H(x_{\mathrm{c}}, y_{\mathrm{c}}, r). \tag{12}$$

Before applying MDE to inner location, the ranges of problem variables should be determined. For inner location, the abscissa of the center of inner circle is in the range of $[x_{\mathrm{mid}} - \Delta x, x_{\mathrm{mid}} + \Delta x]$, and the ordinate of the center of inner circle is in the range of $[y_{\mathrm{mid}} - \Delta y, y_{\mathrm{mid}} + \Delta y]$. In addition, the radius of inner circle is in the range of $[r_{\min}, r_{\max}]$. Based on these settings, we can obtain the accurate inner location image by using MDE, and it is shown in Fig.7.

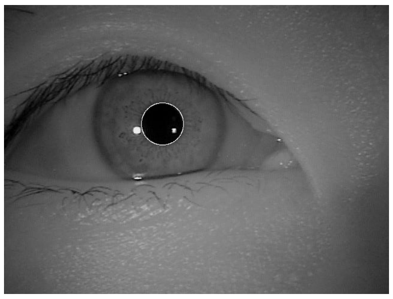

Fig. 7 The accurate inner location image

5 Detecting iris boundary by MDE

5.1 Eliminating noises

Extract the iris edge image using canny operator[19], and the edge image is shown in Fig.8.

According to Fig.8, there exist three noises in the above edge image, and they are eyelash, iris inner boundary and iris texture, respectively. These noises are harmful to the accuracy and speed of outer location. Thus, it is necessary to adopt useful methods to eliminate these noises and decrease their impacts on outer location. More specifically, two kinds of noise areas are are determined in terms of the parameters (x_{inner}, y_{inner} and r) of the inner circle o, and they are illustrated as follows:

The first noise area is outside the iris outer boundary. Suppose this area is represented by set Ω, then its opposite area is represented by set $\bar{\Omega}$ which is given by

$$\bar{\Omega} = \{(x, y) | o_{\mathrm{top}} \leqslant x \leqslant o_{\mathrm{bottom}}, o_{\mathrm{left}} \leqslant y \leqslant o_{\mathrm{right}}\}. \tag{13}$$

Here, $o_{\mathrm{top}} = x_{\mathrm{inner}} - (r + R_{\max})/2$, $o_{\mathrm{bottom}} = x_{\mathrm{inner}} + (r + R_{\max})/2$, $o_{\mathrm{left}} = y_{\mathrm{inner}} - 1.1R_{\max}$, $o_{\mathrm{right}} = y_{\mathrm{inner}} + 1.1R_{\max}$. Some noises including eyelash and eyelid can be eliminated in set Ω. Regarding the value of $1.1R_{\max}$, if it is set to a small value such as $1.01R_{\max}$, $1.02R_{\max}$ etc., the iris outer boundary is likely to be eliminated. If it is set to a large value such as $1.2R_{\max}$, $1.3R_{\max}$ etc., the noises beside iris outer boundary will not be overcome well. Therefore, the value of $1.1R_{\max}$ is suitable for keeping a balance between eliminating iris texture and preserving iris outer boundary.

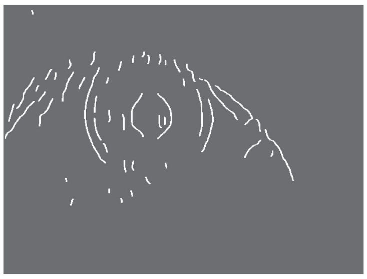

Fig. 8 The edge image

The second noise area is inside the iris outer boundary. Suppose this area is represented by set $\varPhi$, then it is given by

$$\bar{\varPhi} = \{(x, y) | 1 \leqslant x \leqslant cols, y_{\mathrm{left}} \leqslant y \leqslant y_{\mathrm{right}}\}. \tag{14}$$

Here, $y_{\mathrm{left}} = y_{\mathrm{inner}} - (r + R_{\min})/2$, $y_{\mathrm{left}} = y_{\mathrm{inner}} + (r + R_{\min})/2$. This area are mainly composed of eyelash, iris inner boundary and iris texture.

By eliminating the noises in the above two kinds of areas, we can obtain the denoised iris edge image, and it is shown in Fig.9.

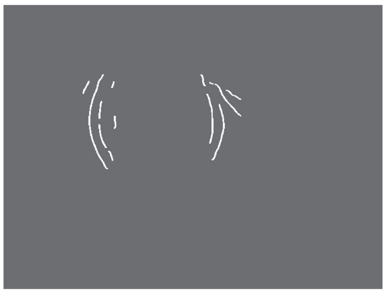

Fig. 9 The denoised iris edge image

According to Fig.9, most noises including eyelash, iris texture and iris inner boundary have been eliminated from the iris edge image. Therefore, the above denoising procedure plays a significant role in increasing the speed and accuracy of outer location.

5.2 Accurate location of iris boundary by using MDE

After denoising the iris edge image, we will rely on MDE to find the optimal outer circle which can fit the iris boundary. Moreover, MDE chooses the function f related to the accumulator array of Hough transform as its objective function (As Eq. (12)). Before applying MDE to outer location, the ranges of problem variables should be determined. For outer location, the abscissa of the center of inner circle is in the range of $[x_{\text{inner}} - \Delta X, x_{\text{inner}} + \Delta X]$, and the ordinate of the center of inner circle is in the range of $[y_{\text{inner}} - \Delta Y, y_{\text{inner}} + \Delta Y]$. In addition, the radius of outer circle is in the range of $[R_{\min}, R_{\max}]$. Therefore, we can find accurate iris boundary by using MDE, and it is shown in Fig.10.

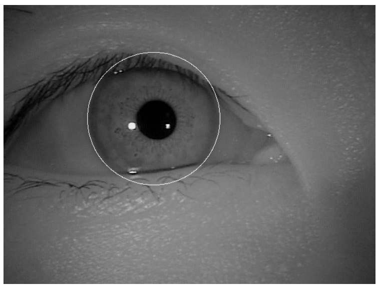
Fig. 10 The accurate outer location image

6 Experimental results and analysis

The Chinese Academy of Sciences Institute of Automation (CASIA) eye image database[18] is utilized to perform the following experiments. Furthermore, 200 iris images from different individuals are chosen to investigate the efficiency of MDE on performing iris location. The iris location algorithm based on MDE is executed in MATLAB on the Intel(R) Core(TM)2 2.30 GHz PC. To be more specific, the MDE parameters include population size $M = 30$; the maximal number of iterations $K = 100$; control parameter $\mu = 4$; stagnation period SP=20; the minimal crossover rate $CR_{\min} = 0.5$; the maximal crossover rate $CR_{\min} = 1$. In the mean time, problem parameters are determined as follows: the problem dimension $N = 3$; For inner location, $\Delta x = 20$; $\Delta y = 20$; $r_{\min} = 20$; $r_{\max} = 65$; For outer location, $\Delta X = 20$; $\Delta Y = 20$; $R_{\min} = 85$; $R_{\max} = 125$; Additionally, four other iris location algorithms are selected to compare with ILA-MDE, and they are Hough transform (HT)[2], differential evolution algorithm based on self-adapting control parameters (SADE)[16], opposition-based differential evolution (ODE)[20] and adaptive differential evolution with optional external archive (JADE)[13], respectively. The above five methods are used to implement iris location for 200 iris, and the experimental results are recorded in Table 1.

Table 1 The computational times of HT, SADE, ODE, JADE and MDE for iris location

Problem	Algorithm	$t_{\min}$/s	$t_{\max}$/s	t_{mean}/s	t_{std}/s	PTS/%	SR/%
Inner location	HT	1.7559	5.8383	2.8774	0.6934	76.48	98
	SADE	0.6376	0.8204	0.6826	0.0335	0.84	98
	ODE	0.7450	1.2538	0.8523	0.0872	20.58	98
	JADE	0.6459	0.7777	0.6809	0.0251	0.59	98
	MDE	0.6170	0.8279	0.6769	0.0325	—	98
Outer location	HT	3.7023	9.0476	5.5577	0.9373	85.89	97.5
	SADE	0.7228	0.9490	0.8015	0.0474	2.16	97
	ODE	0.9796	1.6930	1.2265	0.1190	36.06	97
	JADE	0.7568	1.0929	0.8518	0.0670	7.94	97.5
	MDE	0.6746	0.9485	0.7842	0.0393	—	97.5
Complete location	HT	6.2522	11.5756	8.4351	1.1064	82.68	96.5
	SADE	1.3747	1.6676	1.4841	0.0610	1.56	96.5
	ODE	1.7966	2.4605	2.0788	0.1448	29.72	96.5
	JADE	1.4113	1.7688	1.5327	0.0721	4.68	96.5
	MDE	1.3295	1.6317	1.4610	0.0526	—	96.5

Here, $t_{\min}$ and $t_{\max}$ represent the minimal time and maximal time of iris location, respectively. t_{mean} represents the average time, and t_{std} represents the standard deviation. The term SR stands for success rate. For measuring the improvement, PTS (percentage of time saving) is defined to measure the time saving of iris location based on MDE to the other four methods, and it is given by

$$PTS = (t_{\text{other}} - t_{\text{MDE}})/t_{\text{other}}, \tag{15}$$

where t_{MDE} denotes the time of iris location based on MDE and t_{other} denotes the time of iris location based on any other method.

According to four criterions ($t_{\min}$, $t_{\max}$, t_{mean} and t_{std}), MDE performs better than the other four methods for iris location in most cases. To be more specific, MDE achieves the smallest values of $t_{\min}$, $t_{\max}$, t_{mean} and t_{std}, and they are 0.6746 s, 0.9485 s, 0.7842 s and 0.0393 s, respectively, for outer location. Moreover, the values of $t_{\min}$, $t_{\max}$, t_{mean} and t_{std} obtained using MDE are also smaller than those of the other four methods for complete location, and they are 1.3295 s, 1.6317 s, 1.4610 s and 0.0526 s, respectively. Furthermore, the average execution times of HT, SADE, ODE, JADE and MDE are 8.4351 s, 1.4841 s, 2.0788 s, 1.5327 s and 1.4610 s per image, whose size is 480×640, thus, MDE is the fastest. According to the term PTS, MDE saves the computational time by 82.68%

of the HT's, 1.56% of the SADE's, 29.72% of the ODE's, and 4.68% of the JADE's in average. Although the superiority of MDE over SADE and JADE is small in terms of the criterion PTS, even a little time saving is critical to the efficiency of iris location. Additionally, the success rate of MDE is the same as those of the other four methods for the inner location, and it is equal to 98%. Regarding the outer location, the success rate of MDE is the same as those of HT and SADE, and it is higher than those of ODE and JADE. With respect to the complete location, all the five methods have the same success rate, which is equal to 96.5%. In fact, the success rate of each iris location algorithm does not only depend on the searching method such as HT, SADE, ODE, JADE and MDE, etc., but also depends on the tools of edge extraction such as canny operator[19], sobel operator[21] etc.. As is well known, the difference between the average gray level intensity of pupil and that of iris is notable. Therefore, the inner boundary can be easily obtained by using canny operator. In contrast, the difference between the average gray level intensity of iris and that of scleral is smaller, and the outer boundary extracted by canny operator usually loses its upper and lower segments because of the masking of eyelashes and eyelids. Thus, the outer boundary could not be extracted easily by canny operator in some cases.

7 Conclusions

In this paper, a modified differential evolution (MDE) algorithm is proposed for iris location. MDE makes three improvements on the original DE algorithm. First, MDE adjusts scale factor F and crossover rate CR by using chaotic sequences and uniform distribution, respectively, because both distributions are beneficial to increasing the diversity of candidate solutions and the randomness of searching. Second, MDE modifies the mutation operation of the worst solution in terms of center solution, which provides a promising searching direction for global searching. Third, MDE updates the trapped solutions according to the best solution, which is helpful to enhance its convergence. In addition, we introduce two kinds of denoising methods to decrease the effects of noises on inner and outer locations. Finally, correct results of iris location are obtained by using MDE and the other four methods in most cases. According to experimental results, our proposed MDE has demonstrated higher efficiency in saving execution time when compared to the other four methods for iris location. In this paper, our main work contains denoising edge image and searching inner and outer circles. In order to further improve the success rate of iris location algorithm, our subsequent work will concentrate on the modification of the canny operator, which is beneficial to improving the quality of edge image.

References:

[1] DAUGMAN J G. High confidence visual recognition of persons by a test of statistical independence [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993, 15(11): 1148 – 1160.

[2] WILDES R. Iris recognition: an emerging biometric technology [J]. *Proceedings of the IEEE*, 1997, 85(9): 1348 – 1363.

[3] DAUGMAN J. New methods in iris recognition [J]. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 2007, 37(5): 1167 – 1175.

[4] TAN T N, HE Z F, SUN Z N. Efficient and robust segmentation of noisy iris images for non-cooperative iris recognition [J]. *Image and Vision Computing*, 2010, 28(2): 223 – 230.

[5] HE Z F, TAN T N, SUN Z A, et al. Toward accurate and fast iris segmentation for iris biometrics [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009, 31(9): 1670 – 1684.

[6] STORN R, PRICE K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces, technical report TR–95–012 [R]. Berkeley, USA: International Computer Science Institute, 1995.

[7] WANG Y, LI B, WEISE T. Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems [J]. *Information Sciences*, 2010, 180(12): 2405 – 2420.

[8] MAULIK U, SAHA I. Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery [J]. *Pattern Recognition*, 2009, 42(9): 2135 – 2149.

[9] YOUSEFI H, HANDROOS H, SOLEYMANI A. Application of differential evolution in system identification of a servo-hydraulic system with a flexible load [J]. *Mechatronics*, 2008, 18(9): 513 – 528.

[10] TRIGUERO I, GARCÍA S, HERRERA F. Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification [J]. *Pattern Recognition*, 2011, 44(4): 901 – 916.

[11] GÄMPERLE R, MÜLLER S D, KOUMOUTSAKOS P. A parameter study for differential evolution [C] //*WSEAS International Conferenece on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*. Greece: IEEE, 2002: 293 – 298.

[12] ZHANG J, SANDERSON A C. An approximate Gaussian model of differential evolution with spherical fitness functions [C] //*IEEE Congress on Evolutionary Computation*. Singapore: IEEE, 2007: 2220 – 2228.

[13] ZHANG J Q, SANDERSON A C. JADE: adaptive differential evolution with optional external archive [J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945 – 958.

[14] COELHO L S, LEE C S. Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches [J]. *International Journal of Electrical Power & Energy Systems*, 2008, 30(5): 297 – 307.

[15] LIU S S, WANG M, HOU Z J. Hybrid algorithm of chaos optimisation and SLP for optimal power flow problems with multimodal characteristic [C] //*Generation, Transmission and Distribution, IEE Proceedings*, 2003, 150(5): 543 – 547.

[16] BREST J, GREINER S, BOŠKOVIĆ B, et al. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems [J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(6): 646 – 657.

[17] LIU Y, QIN Z, SHI Z W, et al. Center particle swarm optimization [J]. *Neurocomputing*, 2007, 70(4–6): 672 – 679.

[18] CASIA-IrisV4. http://www.sinobiometrics.com/casiairis.htm.

[19] CANNY J A. Computional approach to edge detection [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, 8(6): 679 – 698.

[20] RAHNAMAYAN S, TIZHOOSH H R, SALAMA M M A. Opposition-based differential evolution [J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(1): 64 – 79.

[21] KITTLER J. On the accuracy of the Sobel edge detector [J]. *Image and Vision Computing*, 1983, 1(1): 37 – 42.

作者简介:

邹德旋 (1982–), 男, 博士, 讲师, 目前研究方向为数字图像处理、最优化, E-mail: zoudexuan@163.com;

王　鑫 (1979–), 男, 博士研究生, 讲师, 目前研究方向为通信系统、数字图像处理, E-mail: wangxin7988@163.com;

段　纳 (1981–), 女, 博士, 副教授, 目前研究方向为控制理论, E-mail: superxuan23@126.com.