

文章编号 1004-924X(2014)04-1056-08

基于改进粒子群的虹膜定位算法

邹德旋^{1*}, 王 鑫², 陈传虎¹, 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:提出了一种改进的粒子群优化算法(IPSO)来解决虹膜定位问题。该算法采用两种速度更新策略来增强种群多样性并提高算法自身的收敛速度,并提出一种变异操作以阻止 IPSO 陷入局部最优。对虹膜内进行边缘定位时,通过搜索 6 条直线与虹膜内边缘的交点来获得 12 个边缘点;另外建立了与这 12 个点有关的目标函数,并用 IPSO 来优化该函数。根据 IPSO 在该函数上的应用,找到一个最合适的圆来拟合虹膜内边缘。进行虹膜外定位时,设计了一个模板来提取虹膜外边界,然后从外边界中选择 12 个边缘点,并同样使用 IPSO 找到一个最合适的圆来拟合虹膜外边缘。为了验证基于改进粒子群优化算法的虹膜定位方法(ILA-IPSO)的性能,从中国科学院自动化研究所的数据库中选择了不同个体的 108 幅虹膜图像。实验结果表明,ILA-IPSO 算法要好于其它两种方法,该算法利用最少的定位时间获得了最高的成功率。

关键词:改进粒子群优化;虹膜定位;速度更新;变异操作;内边界;外边界

中图分类号:TP391.41 **文献标识码:**A **doi:**10.3788/OPE.20142204.1056

Iris location algorithm based on improved particle swarm optimization

ZOU De-xuan^{1*}, WANG Xin², CHEN Chuan-hu¹, DUAN Na¹

(1. School of Electrical Engineering and Automation, Jiangsu Normal University, Xuzhou 221116, China;

2. Information and Control Engineering Faculty, Shenyang Jianzhu University, Shenyang 110168, China)

* Corresponding author, E-mail: zoudexuan@163.com

Abstract: An Improved Particle Swarm Optimization (IPSO) algorithm was proposed to solve iris localization. The IPSO adopted two velocity updating strategies to diversify the swarm and to improve its convergence rate. In addition, a mutation operation was used to prevent the IPSO from being trapped into the local optimums. For inner location, twelve edge points were determined by finding the intersection points of six straight lines with the inner boundary. A function related to these twelve points was constructed, and this function was optimized by IPSO. By applying IPSO to this function, a most suitable circle could be found to fit the inner boundary of the iris. With respect to outer location, a template was designed to extract the outer boundary of the iris, and twelve edge points were equally selected from the outer boundary of the iris. By applying IPSO to a function related to these

收稿日期:2013-05-24;修订日期:2013-07-16.

基金项目:江苏师范大学基金资助项目(No. 9212812101)

twelve points, a most appropriate circle could be obtained to fit the outer boundary of the iris. 108 iris images of different individuals were chosen from the eye image database of Institute of Automation of Chinese Academy of Sciences (CASIA) to test the performance of our proposed Iris Location Algorithm based on Improved Particle Swarm Optimization (ILA-IPSO) algorithm. Experimental results reveal that the ILA-IPSO performs better than the other two approaches for iris localization. This method uses the least time of iris localization, and obtains the highest success rate.

Key words: Improved Particle Swarm Optimization(IPSO); iris location; velocity updating; mutation operation; inner boundary; outer boundary

1 引言

虹膜识别^[1-2]是一种重要的生物识别技术,该技术可用于寻找和识别不同个体,对于保护个人隐私、商业信息以及国家机密起着至关重要的作用。作为虹膜识别技术中的一个非常重要的步骤——虹膜定位已经受到了越来越多的关注,该技术的改进有利于提高后续步骤(特征提取^[3]和匹配^[4])的质量。

很多学者提出了一些经典的虹膜定位方法。Daugman^[5]引入了一种微积分算子,通过搜索微积分的最大值,即可找到虹膜内、外边缘。Wildes^[6]使用了一种 Hough 变换(Hough Transformation, HT)来进行虹膜定位,这种方法中的累加器数组取得最大值的参数组合通常作为虹膜的内/外圆参数。Huang 等人^[7]提出了一种新的非极大值抑制方法来保留环形边缘,并消除辐射边缘;他们通过使用一种阈值操作来消除独立的边缘,并产生最终的二值化边缘图像;最后,他们设计了一种自适应的虹膜检测方法来获得虹膜边缘,该方法比传统方法具有更高的成功率。Basit 和 Javed^[8]提出了一种基于亮度值分析的虹膜定位方法,他们使用两种方法来计算虹膜的内边缘数据,再通过寻找不同方向的最大梯度点来获得虹膜外边缘;实验结果表明,他们的方法较为有效和准确。Pundlik 等人^[9]介绍了一种图切割的方法,并将其用于非理想虹膜分割中;他们通过综合考虑基于带有图像梯度、眼睛几何启发式的亮度和纹理的区域来实现虹膜分割;该方法的一个优点是能够解决虹膜亮度和背景区域的多模型分布。吴建华等人^[10]提出了一种基于小范围搜索的虹膜识别算法(Iris Location Algorithm based on Small-Scale Searching, ILA-SSS),该算法缩

小了虹膜内、外圆圆心的搜索范围,节省了计算时间。

为了提高虹膜定位的精度和速度,本文提出了一种基于改进粒子群优化算法的虹膜定位方法(Iris Location Algorithm based on Improved Particle Swarm Optimization, ILA-IPSO)。本文的研究内容包括两个方面,第一,对原始粒子群优化算法(Particle Swarm Optimization, PSO)^[11-12]的改进,即提出了一种改进的粒子群优化算法(Improved Particle Swarm Optimization algorithm, IPSO);第二,使用 IPSO 算法进行虹膜内、外边缘的定位。最后,通过实验比较了 ILA-IPSO 算法和 ILA-SSS 以及 HT 算法的性能。

2 一种改进的粒子群优化

PSO 算法^[11]是一种强有力的优化方法,且一些 PSO 变种已被用于解决各种复杂问题,如瓷砖制造^[13]、太阳能光伏发电系统^[14]、结构设计优化^[15]等。但它实际是一种枚举法,非常耗时。鉴于 PSO 良好的优良性能,本文使用该方法来进行虹膜定位。

由于原始 PSO 算法收敛性较差,全局搜索能力较弱,本文提出了一种改进的粒子群优化算法(IPSO)来克服这些缺点,具体步骤如下:

(1) 初始化算法参数及问题参数

IPSO 的参数包括种群大小 M , 最大迭代次数 K , 停滞周期(Stagnation Period, SP), 高斯分布的标准差 $\delta=2$; 问题参数包括变量的数目 N , 第 $j(j=1, \dots, N)$ 个变量的下限 \underline{x}_j 和上限 \bar{x}_j 。

(2) 产生 M 个粒子

在 $[\underline{x}_j, \bar{x}_j]$ 内为第 $i(i=1, \dots, M)$ 个粒子的第 j 个变量随机产生一个值,表示如下:

$$x_j^i = \underline{x}_j + \text{rand}() \times (\bar{x}_j - \underline{x}_j), \quad (1)$$

式中: $rand()$ 代表一个在 $[0, 1]$ 内服从均匀分布的随机数。

(3) 更新速度和位置

PSO 算法的速度更新和位置更新公式分别为:

$$v_{i,j}^{k+1} = v_{i,j}^k + c_1 r_1 (pb_{i,j} - x_{i,j}^k) + c_2 r_2 (gb_j - x_{i,j}^k), \quad (2)$$

$$x_{i,j}^{k+1} = x_{i,j}^k + v_{i,j}^{k+1}, \quad (3)$$

其中: i 表示粒子的索引; j 表示变量的索引; k 是当前迭代次数; pb 和 gb 分别为个体最好位置和全局最好位置; 系数 c_1 和 c_2 都是加速常数, r_1 和 r_2 为 $[0, 1]$ 内服从均匀分布的随机数。

PSO 算法在寻优初期具有较快的收敛速度, 但到了寻优后期, 它的收敛速度会减慢。为了提升 PSO 算法的收敛性, 本文采用以下的速度更新公式:

$$v_{i,j}^{k+1} = \begin{cases} v_{i,j}^k + c_1 r_1 (pb_{i,j} - x_{i,j}^k), & \text{If } sig(i) = 1 \\ v_{i,j}^k + c_2 r_2 (gb_j - x_{i,j}^k), & \text{Otherwise} \end{cases}, \quad (4)$$

其中: r_1 和 r_2 为 $[0, 1]$ 内服从均匀分布的随机数。 $sig(i)$ 是一种标记, 它被用于确定采用哪种速度更新策略。 $sig(i)$ 被初始化为 1 时, 粒子 i 更喜欢采用第一种策略, 即根据它经过的个体最好位置来更新速度, 这有助于 IPSO 算法执行局部搜索。如果粒子 i 在 SP 内持续处于停滞状态, 则将 $sig(i)$ 设为零。在这种情况下, 粒子 i 更喜欢采用第二种速度更新策略, 即根据它全局最好位置来更新速度, 这有助于 IPSO 算法执行全局搜索。类似地, 如果带有 $sig(i) = 0$ 的粒子 i 在 SP 内持续处于停滞状态, 则将 $sig(i)$ 设为 1。总之, 这两种速度更新方程不仅能提高 IPSO 算法的收敛速度, 还能改进候选解的质量。

如果只采用式(4)中的第二种速度更新公式, 则意味着所有粒子都要根据当前种群中的最好位置来进行搜索, 这样很容易导致算法早熟, 也就是陷入局部最优, 因此不利于算法收敛。通过引入个体最优位置可以增加粒子更新速度的方式, 拓展算法的搜索空间, 从而能有效避免算法陷入局部最优。虽然这两种速度更新策略在寻优前期会影响最优解的产生, 但它却兼顾了拓展搜索空间和避免陷入局部最优这两个方面。而且, 通过使

用这两种策略会在寻优中期和后期逐渐改善解的质量, 使算法最终收敛于全局最优解。

(4) 变异

变异操作能增加整个种群的多样性, 且能有效避免 IPSO 算法发生早熟现象。总之, 该变异操作可用公式表示如下:

$$x_{i,j}^{k+1} = Nd(x_{i,j}^{k+1}, \delta), \quad (5)$$

其中: Nd 表示高斯分布, δ 表示标准差。而且, 该变异运算是小概率事件, 它出现的概率为:

$$p_m = p_{\min} + (p_{\max} - p_{\min}) \times k/K, \quad (6)$$

其中: p_m 定义为变异率, p_{\min} 、 p_{\max} 分别代表最小、最大变异率。在寻优初期, p_m 保持了较小值, 这对算法的收敛性影响较小。在寻优后期, 变异率保持了较大值。此时, 粒子具有很强的逃离局部最优点的能力, 并能朝着全局最优优点运动。

(5) 判断终止条件

若当前迭代次数 k 大于 K , 则终止 IPSO 算法, 并输出最优结果。否则, 重复步骤(3)和(4)。

PSO 在解决大规模优化问题时, 一般需要较多的计算时间, 而它在解决小规模问题(本文所研究的问题维数为 3)时, 所需的时间较短。另外, PSO 通过所有粒子的共同学习和相互作用来找到全局最优解, 该方法不需要列举出所有的可能情况, 因此, 能够节省时间。相反, Hough 算法在解决虹膜定位时要考虑所有的可能情况, 比较费时。因此, PSO 在提高定位速度方面比 Hough 算法具有更大的优势。IPSO 是 PSO 的一种改进版本, 它的运行时间和 PSO 的差不多, 但它的优化性能要更好, 也就是说, 基于 IPSO 的虹膜定位的精度要高于基于 PSO 的虹膜定位的精度。

3 使用 IPSO 算法进行虹膜内定位

虹膜定位包括内边缘检测和外边缘检测, 且内定位和外定位的预处理方法明显不同, 以下将给出详细的介绍:

3.1 虹膜粗定位

在虹膜图像中, 瞳孔的灰度级要比虹膜和巩膜的都低, 如图 1 所示。但眼睫毛的灰度级与瞳孔的比较接近, 这会影响灰度投影法的准确性, 所以在使用灰度投影法之前必须采取有效措施来减少眼睫毛的影响。消除眼睫毛等干扰的步骤及灰

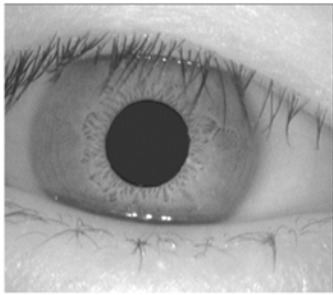


图 1 原始虹膜图像
Fig. 1 Original iris image

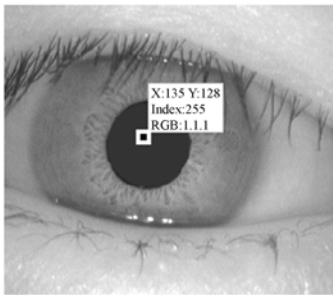


图 2 内圆粗定位圆心的位置
Fig. 2 Rough location of inner circle center

度投影法的使用可参见文献[16]。最终可以获得粗定位圆心 $M(x_{rough}, y_{rough})$,如图 2 所示。

3.2 确定 12 个边缘点

为了找到合适的圆来拟合虹膜内边缘,需要从虹膜内边缘上选择一些像素点,步骤如下:

(1)提取图像边缘

使用 Canny 算子^[17]来检测虹膜内边缘,基于 Canny 算子的边缘图像如图 3 所示。



图 3 基于 Canny 算子的边缘图像
Fig. 3 Edge image based on Canny operator

(2)在虹膜内边缘上确定 12 个边缘点

确定 6 条直线 l_1, l_2, l_3, l_4, l_5 和 l_6 。它们具有不同的倾角,分别为 $\frac{\pi}{7}, \frac{2\pi}{7}, \frac{3\pi}{7}, \frac{4\pi}{7}, \frac{5\pi}{7}$ 和 $\frac{6\pi}{7}$,且

这 6 条直线都经过 M 点(如图 4 所示)。

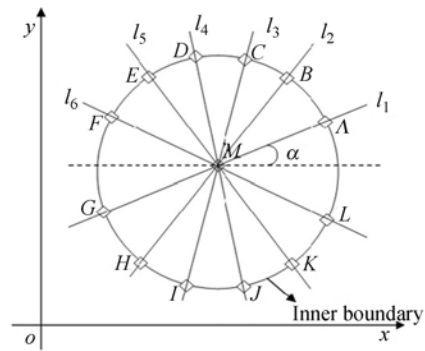


图 4 找到虹膜内的 12 个边缘点
Fig. 4 Determination of twelve edge points on inner boundary of iris

在图 4 中,直线 l_1 的倾角设为 $\alpha = \frac{\pi}{7}$,相应的直线方程表示为:

$$y = \tan \alpha \times (x - x_{rough}) + y_{rough}. \quad (7)$$

确定直线 l_1 的方程后,即可求出它与虹膜内边界的 2 个交点 A 和 G。用同样的方法可以求出其它 10 个边缘点的坐标。

3.3 基于 IPSO 算法的虹膜内边缘的精定位

设内圆 o 的圆心坐标为 (x, y) ,半径为 r ,且 12 个边缘点 A, B, ..., L 的坐标分别为 $(x_1, y_1), (x_2, y_2), \dots, (x_{12}, y_{12})$ 。在理想情况下,这 12 个边缘点都经过一个圆,且圆的方程可表示为:

$$(x - x_i)^2 + (y - y_i)^2 = r^2. \quad (8)$$

实际上,这 12 个点并非全在同一个圆上,本文试图找到一种最优的 (x, y, r) 组合,使这 3 个参数能够满足以下的条件:

$$f = \min_{x, y, r} \sum_{i=1}^{12} |(x - x_i)^2 + (y - y_i)^2 - r^2|, \quad (9)$$

其中: f 是一个与 x, y 和 r 有关的函数。本文使用 IPSO 算法来优化函数 f ,并可获得最优的 (x, y, r) 参数组合,且用相应的圆 o 来拟合虹膜内边缘。

4 使用 IPSO 算法进行虹膜外定位

4.1 虹膜外定位前的去噪处理

使用 Canny 算子获得的边缘图像如图 5 所示。

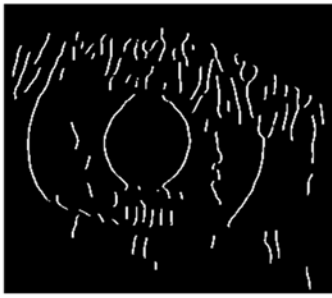


图 5 边缘图像
Fig. 5 Edge image

对虹膜外边缘来说, 睫毛、虹膜内边缘、虹膜纹理都是噪声。这些噪声不仅会影响定位的精度, 还会影响定位的速度。为了减少噪声对虹膜外定位的影响, 需要采取一定的措施来克服这些困难。具体如下: 根据虹膜内圆 o 的 3 个参数 (x_{inner} 、 y_{inner} 和 r) 可以确定 3 个噪声区域, 它们分别表示为:

区域 1:

$$\{(x, y) | 1 \leq x \leq \text{rows}, y_{\text{inner}} - \Delta y_1 \leq y \leq y_{\text{inner}} + \Delta y_1\}, \quad (10)$$

其中: 变化量 Δy_1 的值可以根据 r 值进行自适应调整。当 r 值在 $[r_{\text{min}} + \frac{2}{3}(r_{\text{max}} - r_{\text{min}}), r_{\text{max}}]$ 范围内时, Δy_1 的值设置为 15。否则, Δy_1 的值设置为 25。在该区域内的一些噪声可以被有效地消除掉, 这些噪声包括睫毛、虹膜纹理和虹膜内边缘。

区域 2:

$$\{(x, y) | [x - (x_{\text{inner}} - 2r)]^2 + (y - y_{\text{inner}})^2 > (2r + \Delta r_1)^2\}, \quad (11)$$

其中: 变化量 Δr_1 的值可以根据 r 值进行自适应调整。当 r 值在 $[r_{\text{min}} + \frac{2}{3}(r_{\text{max}} - r_{\text{min}}), r_{\text{max}}]$ 范围内时, Δr_1 的值设置为 80。否则, Δr_1 的值设置为 100。在该区域内的一些噪声可以被有效地消除掉, 这些噪声主要为眼睛下方的睫毛。

区域 3:

$$\{(x, y) | [x - (x_{\text{inner}} + 2r)]^2 + (y - y_{\text{inner}})^2 > (2r + \Delta r_2)^2\}, \quad (12)$$

其中: 变化量 Δr_2 的值可以根据 r 值进行自适应调整。当 r 值在 $[r_{\text{min}} + \frac{2}{3}(r_{\text{max}} - r_{\text{min}}), r_{\text{max}}]$ 范围内时, Δr_2 的值设置为 60。否则, Δr_2 的值设置为

80。在该区域内的噪声主要为眼睛上方的睫毛。

在以上 3 个区域消除噪声后, 即可得到去噪图像, 如图 6 所示:

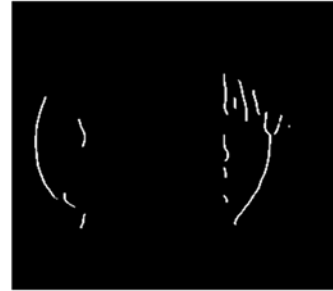


图 6 去噪后的虹膜图像
Fig. 6 Denoised iris edge image

从图 6 可以看出, 大部分噪声已经被消除了。这样虹膜外定位的精度和速度就会得到提高。

4.2 检测虹膜外边缘

在图 6 中, 两条最长的曲线段留在了去噪后的虹膜边缘图像中, 且它们分别是虹膜外边缘的左半部分和右半部分。为了提取出这两部分, 可采用如图 7 中的模板:

P_8	P_7	P_6
P_1	P_0	P_5
P_2	P_3	P_4

图 7 提取虹膜外边缘的模板

Fig. 7 Template of extracting iris outer edge

在图 6 中, 采用 3 个步骤来提取虹膜外边缘的左半部分和右半部分, 步骤如下:

(1) 用直线 $y = y_{\text{inner}}$ 将去噪图像划分为两部分, 在后续的步骤中, 分别在这 2 个被划分后的图像中来检测虹膜外边缘的左半部分和右半部分。

(2) 在被划分后的左侧图像中检测虹膜外边缘的左半部分。如果当前边缘点是 P_0 , 则需要检测出下一个与 P_0 相连的边缘点 P_0' , 且选择的顺

序分别为 P_2 、 P_3 、 P_4 和 P_5 。换句话说,如果与 P_0 相连的边缘点 P_0' 已经被找到,则比 P_0' 优先级低的那些像素点将不被考虑。搜索到 P_0' 后,将 P_0' 作为 P_0 ,并执行新的搜索。如果在 P_2 、 P_3 、 P_4 和 P_5 中没有像素点被找到,则意味着一条曲线段的检测已经结束,并继续检测其它的曲线段。而且,一旦一条曲线段被检测出,则在它上面的像素点的总数以及每个像素点的坐标就被记录下来。与此同时,这条曲线段将从图 6 中排除掉,这样就不会影响对于其它曲线段的检测。当检测完所有的曲线段后,最长的那条即为虹膜外边缘的左半部分。

(3)在被划分后的右侧图像中检测虹膜外边缘的右半部分,这与步骤(2)相似。唯一不同的是选择 P_0' 的顺序。此时 P_0' 的选择顺序分别为 P_4 、 P_3 、 P_2 和 P_1 。

基于以上 3 个步骤,即可获得虹膜外边缘的左半部分和右半部分,如图 8 所示。

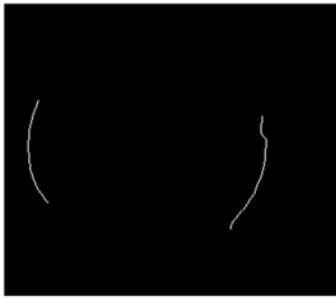


图 8 虹膜外边缘的左、右部分

Fig. 8 Left and right parts of iris outer edge

4.3 基于 IPSO 算法的虹膜外定位

虹膜外边缘近似为一个圆,因此在图 8 中,分别从虹膜的左侧和右侧各选取 6 个边缘点来拟合一个圆 O 。设这 12 个边缘点的坐标分别为 (a_1, b_1) 、 (a_2, b_2) ... (a_{12}, b_{12}) 。设圆 O 的圆心坐标为 (X, Y) ,半径为 R ,使用 IPSO 来优化如下公式:

$$f = \min_{X, Y, R} \sum_{i=1}^{12} |(X - a_i)^2 + (Y - b_i)^2 - R^2|. \quad (13)$$

f 是一个关于 X 、 Y 和 R 的函数。本文使用 IPSO 来优化函数 f ,且将 f 作为 IPSO 的目标函数。通过运行 IPSO,可以获得最优 (X, Y, R) 的

参数组合,且用一个最合适的圆 O 来拟合虹膜外边缘。

通过使用 IPSO 算法来解决虹膜定位问题可以得到准确的虹膜定位图。如图 9 所示:

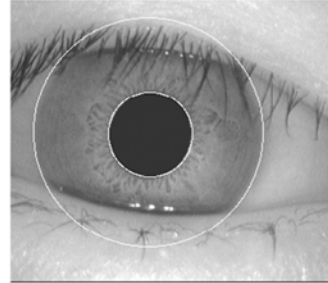


图 9 使用 IPSO 算法获得的准确的虹膜定位图

Fig. 9 Accurate iris location image obtained by IPSO

5 实验结果和分析

本实验从中国科学院自动化研究所^[18]的数据库中选择了 108 个不同个体的虹膜图像来验证 ILA-IPSO 的性能。本文使用 Matlab7.0 软件实现了 3 种虹膜定位方法。用于仿真实验的电脑配置为 Intel(R) Core(TM) i5-2410M CPU @ 2.30 GHz。ILA-IPSO 的相关参数设置如下:种群大小 $M=60$;最大迭代次数 $K=150$;学习因子 $c_1 = c_2 = 2$;停滞周期 $SP=5$;标记 $sig(i)=1$; $count(i)=0$ ($i=1, 2, \dots, M$);最小、最大变异率分别为 $p_{min}=0.01$ 和 $p_{max}=0.05$;高斯分布的标准差 $\delta=2$;变量的数目 $N=3$,虹膜内圆半径的最小、最大值分别为 $r_{min}=35$ 和 $r_{max}=65$;外圆半径的最小、最大值分别为 $R_{min}=85$ 和 $R_{max}=125$ 。另外,本文还选择 Hough 变换^[6,12]和基于小范围搜索的虹膜定位算法^[10]来与 ILA-IPSO 进行比较。3 种方法获得的虹膜定位结果如表 1 所示。

在表 1 中, t_{min} 、 t_{max} 分别表示执行虹膜定位所需要的最少、最多时间; t_{mean} 表示平均定位时间; t_{std} 表示关于时间的标准差;SR 表示成功率。另外,本文通过定义 PTS(Percentage of Time Saving)来测量 ILA-IPSO 相对于其它 2 种方法的时间节省程度,其表示为:

$$PTS = \frac{t_{other} - t_{ILA-IPSO}}{t_{other}}, \quad (14)$$

其中: $t_{ILA-IPSO}$ 表示 IPSO 算法所用的定位时间, t_{other} 表示其它 2 种方法所用的定位时间。

表 1 3 种方法的性能比较

Tab. 1 Comparison among three methods

虹膜定位	定位方法	t_{\min}/s	t_{\max}/s	t_{mean}/s	t_{std}/s	PTS/(%)	SR/(%)
内定位	HT	1.312 6	5.136 7	2.750 3	0.814 2	95.29	93.518 5
	ILA-SSS	0.334 3	1.475 7	0.726 5	0.232 1	82.18	92.592 6
	ILA-IPSO	0.117 3	0.162 8	0.129 5	0.032 4	-	97.222 2
外定位	HT	1.949 6	8.615 4	5.362 2	1.255 0	96.51	94.444 4
	ILA-SSS	0.908 0	4.129 1	2.444 6	0.716 9	92.34	93.518 5
	ILA-IPSO	0.161 2	0.237 9	0.187 2	0.022 6	-	96.296 3
整个定位	HT	3.708 4	13.116 1	8.112 4	1.890 0	96.10	93.518 5
	ILA-SSS	1.282 9	4.998 5	3.171 0	0.851 2	90.01	92.592 6
	ILA-IPSO	0.276 7	0.356 2	0.316 7	0.050 1	-	96.296 3

根据表 1 可知,ILA-IPSO 算法的性能要好于其它 2 种方法。无论是虹膜内定位还是外定位,ILA-IPSO 算法的 t_{\min} 、 t_{\max} 和 t_{std} 都要比其它 2 种方法的小,它的 t_{\max} 甚至会比其它方法的 t_{\min} 还小,这表明 ILA-IPSO 算法具有速度快的特点。ILA-IPSO 算法的 t_{std} 也是最小的,这说明 ILA-IPSO 算法具有较强的稳定性。在内定位方面,ILA-IPSO 算法对 HT 和 ILA-SSS 的时间节省百分比(PTS)分别为 95.29% 和 82.18%。在外定位方面,ILA-IPSO 算法对 HT 和 ILA-SSS 的时间节省百分比(PTS)分别为 96.51% 和 92.34%。另外,ILA-IPSO 算法的成功率(SR)也要比其它 2 种方法的高,其在内定位、外定位和整个虹膜定位方面的成功率分别是 97.222 2%、96.296 3% 和 96.296 3%。虽然与其它 2 种方法相比,ILA-IPSO 算法在成功率方面的优势并不是特别明显,但即使是成功率微小的提高,对于虹膜定位效率

来讲也是很有意义的。值得注意的是,整体定位与外定位的成功率不一定相同,本文得到的结果一样只是巧合而已。另外,若要整体成功定位则内定位和外定位必须同时成功,因此,出现整体定位的成功率低于内定位的成功率是正常的。

6 结 论

为了兼顾虹膜定位的速度和精度,本文设计了一种改进的粒子群优化算法(IPSO)。由于该算法使用了 2 种速度更新公式和一种变异操作,使其具有较强的收敛性和稳定性。基于 IPSO 算法较好的优化性能,优化了 2 个圆的参数,并用这 2 个圆来拟合虹膜的内边缘和外边缘。通过实验比较了 IPSO、HT 和 ILA-SSS 算法的性能。仿真结果表明,ILA-IPSO 算法在定位精度和定位速度方面都要好于其它两种方法。

参考文献:

- [1] BOWYER K W, HOLLINGSWORTH K, FLYNN P J. Image understanding for iris biometrics: A survey [J]. *Computer Vision and Image Understanding*, 2008, 110(2): 281-307.
- [2] 陈传虎, 邹德旋, 刘海宽. 应用统计距离实现虹膜定位 [J]. *光学精密工程*, 2012, 20(11): 2516-2522.
- [3] CHEN CH H, ZOU D X, LIU H K. Iris location algorithm by counting distances [J]. *Opt. Precision Eng.*, 2012, 20(11): 2516-2522. (in Chinese)
- [4] XU G Z, ZHANG Z F, MA Y D. An image segmentation based method for iris feature extraction [J]. *The Journal of China Universities of Posts and Telecommunications*, 2008, 15(1): 96-101, 117.
- [5] DONG W B, SUN ZH N, TAN T N. Iris matching based on personalized weight map [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, 33(9): 1744-1757.
- [6] DAUGMAN J G. How iris recognition works [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2004, 14(1): 21-30.
- [7] WILDES R. Iris recognition: an emerging biometric technology [C]. *Proceedings of the IEEE*, 1997, 85(9): 1348-1363.
- [8] HUANG J, YOU X G, TANG Y Y, et al.. A no-

- vel iris segmentation using radial-suppression edge detection [J]. *Signal Processing*, 2009, 89(12): 2630-2643.
- [8] BASIT A, JAVED M Y. Localization of iris in gray scale images using intensity gradient [J]. *Optics and Lasers in Engineering*, 2007, 45(12): 1107-1114.
- [9] PUNDLIK S, WOODARD D, BIRCHFIELD S. Iris segmentation in non-ideal images using graph cuts [J]. *Image and Vision Computing*, 2010, 28(12): 1671-1681.
- [10] 吴建华, 邹德旋, 李静辉. 基于小范围搜索的虹膜定位方法 [J]. *仪器仪表学报*, 2008, 29(8): 1704-1708.
WU J H, ZOU D X, LI J H. Iris location algorithm based on small-scale searching [J]. *Chinese Journal of Scientific Instrument*, 2008, 29(8): 1704-1708. (in Chinese)
- [11] KENNEDY J, EBERHART R C. Particle swarm optimization [C]. *Proc. IEEE International Conference on Neural Networks*, 1995, 1942-1948.
- [12] 葛宝臻, 马云峰, 魏耀林. 求解粒子群粒度分布的改进 Projection 算法 [J]. *光学精密工程*, 2012, 20(1): 197-203.
GE B ZH, MA Y F, WEI Y L. Improved projection algorithm for measuring distribution of particle sizes [J]. *Opt. Precision Eng.*, 2012, 20(1): 197-203. (in Chinese)
- [13] NAVALERTPORN T, AFZULPURKAR N V. Optimization of tile manufacturing process using particle swarm optimization [J]. *Swarm and Evolutionary Computation*, 2011, 1(2): 97-109.
- [14] KHARE A, RANGNEKAR S. A review of particle swarm optimization and its applications in Solar Photovoltaic system [J]. *Applied Soft Computing*, 2013, 13(5): 2997-3006.
- [15] PEREZ R E, BEHDINAN K. Particle swarm approach for structural design optimization [J]. *Computers & Structures*, 2007, 85(19-20): 1579-1588.
- [16] 吴建华, 邹德旋, 李静辉. 一种快速精确的虹膜定位方法 [J]. *仪器仪表学报*, 2007, 28(8): 1469-1473.
WU J H, ZOU D X, LI J H. Fast and accurate iris location algorithm [J]. *Chinese Journal of Scientific Instrument*, 2007, 28(8): 1469-1473. (in Chinese)
- [17] CANNY J A. Computational approach to edge detection [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, 8(6): 679-698.
- [18] CASIA Iris Image Database, Version 1.0. <http://www.sinobiometrics.com/casiairis.htm>.

作者简介:



邹德旋(1982—),男,辽宁大连人,博士,讲师,2005年于辽宁大学获得学士学位,2008年、2011年于东北大学分别获得硕士、博士学位,主要从事数字图像处理、最优化的算法研究。E-mail: zoudexuan@163.com



陈传虎(1974—)男,陕西商洛人,硕士,讲师,1998年于中国矿业大学获得学士学位,2004年于苏州大学获得硕士学位,主要从事智能控制和信号分析与信息处理方面的教学和研究。E-mail: cch@jsnu.edu.cn