

基于改进狼群算法的三维 OTSU 阈值法

曹爽¹, 贺玉珍¹, 安建成²

(1. 运城学院 数学与信息技术学院, 山西 运城 044000;

2. 太原理工大学 信息与技术学院, 山西 晋中 030600)

摘要: 三维 OTSU 阈值法引入了邻域中值及均值信息, 抗噪性明显提高, 但仍存在分割效果不佳的现象, 而且阈值维度的增加导致算法运算量庞大、运行时间过长。为提高算法的计算效率和寻优准确率, 利用改进的狼群优化算法来搜寻最佳分割阈值。采用花授粉算法优化后计算的最佳阈值调节狼群算法的初始阈值, 使狼群在算法初期具有较高的全局探索能力, 提高求解速度。为避免算法后期种群聚集的现象, 将高斯变异引入围攻行为, 设置变异函数, 将满足变异条件的个体狼进行变异, 避免算法陷入局部最优, 提高寻优精度。将改进的算法与传统三维 OTSU 算法以及 CWPA 算法优化的三维 OTSU 法进行图像分割对比, 实验显示, 改进的算法使分割时间显著降低, 并提高了计算阈值的准确度。

关键词: 图像分割; 三维 OTSU; 狼群算法; 花授粉算法; 高斯变异

中图分类号: TP391

文献标识码: A

文章编号: 1673-629X(2020)04-0094-06

doi: 10.3969/j.issn.1673-629X.2020.04.018

A Three-dimension OTSU Threshold Algorithm Based on Improved Wolf Pack Algorithm

CAO Shuang¹, HE Yu-zhen¹, AN Jian-cheng²

(1. School of Mathematics and Information Technology, Yuncheng University, Yuncheng 044000, China;

2. School of Information and Technology, Taiyuan University of Technology, Jinzhong 030600, China)

Abstract: The 3-D OTSU threshold algorithm introduces the information of neighborhood median and mean value, and the anti-noise property is obviously improved, but the segmentation effect is still poor. Moreover, the increase of threshold dimension leads to a large amount of computation and a long running time. In order to improve the efficiency and accuracy, the improved Wolf pack optimization algorithm is used to search for the optimal segmentation threshold. The optimal threshold after the optimization of the flower pollination algorithm is adopted to adjust the initial threshold of the wolf pack algorithm, so that the wolf pack has a higher global exploration ability at the initial stage of the algorithm and improves the solution speed. To avoid population aggregation and local extreme, Gaussian mutation is introduced in beleaguering behavior, and the variable function is set to mutate the individual wolves that meet the mutation conditions, so as to avoid the algorithm to fall into the local optimal and improve the optimization accuracy. The improved algorithm is compared with the traditional three-dimensional OTSU algorithm and the three-dimensional OTSU method optimized by CWPA. The results show that the improved algorithm significantly reduces the segmentation time and improves the accuracy of calculating threshold.

Key words: image segmentation; 3-D OTSU; wolf pack algorithm; flower pollination algorithm; Gaussian mutation

0 引言

一维 OTSU 阈值法是由日本学者大津于 1979 年提出的^[1], 其求解阈值简单、分割速度较快、提取效果良好, 因此得到了广泛应用。但一维 OTSU 法只使用图像的灰度信息, 没有涉及空间邻域信息^[2], 当图像有噪声干扰时无法得到满意的分割结果。为提高抗噪性和分割准确率, 研究者们将 OTSU 算法拓展到二维和

三维^[3]。其中三维 OTSU 阈值法引入了邻域中值及均值信息, 可得到理想的分割效果。

由于中值滤波易滤除椒盐噪声, 均值滤波易滤除高斯噪声^[1], 因此, 三维 OTSU 法既可对椒盐噪声或高斯噪声有较高的抗噪性, 又可对叠加混合噪声(椒盐噪声加高斯噪声)的图像有良好的提取结果。然而 OTSU 法采用穷尽搜索策略, 对于三维 OTSU 法的计

收稿日期: 2019-04-12

修回日期: 2019-08-13

网络出版时间: 2019-12-05

基金项目: 山西省国际科技合作项目(2014081018-2)

作者简介: 曹爽(1992-), 女, 助教, CCF 会员(66895G), 通信作者, 研究方向为人工智能及图像处理。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20191205.1133.042.html>

算需要三重循环,导致计算量过多,求解复杂度呈指数型增长,实时性较差。为此,景晓军等^[4]提出了三维 OTSU 快速递推法,使计算效率提高,但分割准确率一般,在此基础上,范九伦等^[5]对递推公式进行修正,给出新的递推算法,使求解精度提高。叶志伟等^[6]对萤火虫算法 (firefly algorithm, FA) 进行研究,将莱维飞行引入其中,用改进的萤火虫算法来搜寻最优分割阈值,使分割准确率显著上升。仇国庆等^[7]提出了改进的三维 OTSU 法,使用一维 OTSU 法缩小搜索空间以及迭代空间,并采用布谷鸟算法 (cuckoo search, CS) 求解阈值,既降低了计算量又具有较高的抗噪性。

新的群智能算法—狼群算法 (wolf pack algorithm, WPA) 是由吴虎胜等^[8]于 2013 年提出的,其参数设置简单,已被大量应用在多个领域。对狼群算法进行研究,将花授粉算法 (flower pollination algorithm, FPA)^[9]求解的最优阈值加入到狼群算法的初始阈值,再在围攻过程中引入高斯变异,将改进的狼群算法与三维 OTSU 算法结合,可取得较好的分割结果。

1 三维 OTSU 阈值法

三维 OTSU 算法不仅考虑了像素点灰度值,还考虑了像素点邻域均值和中值。设计包含像素灰度及其邻域均值和中值的三维直方图,根据三维直方图求取分割阈值,从而实现自动阈值选取。

对于 $m \times n$ 的图像,用 $h(x, y)$ 表示点 (x, y) 的像素灰度值。对于某一个选定点 $k \times k$ 区域内的平均灰度值用 $f(x, y)$ 表示,计算公式如下:

$$f(x, y) = \frac{1}{k^2} \sum_{m=-k/2}^{k/2} \sum_{n=-k/2}^{k/2} h(x+m, y+n) \quad (1)$$

该选定点 $k \times k$ 区域内的灰度中值是 $g(x, y)$, 计算公式如下:

$$g(x, y) = \text{med} \{ h(x+m, y+n) \} \quad (2)$$

其中, $m = -k/2, \dots, k/2; n = -k/2, \dots, k/2$ 。

根据上面的一系列推导可知,若 L 是图像的灰度等级,则 $f(x, y)$ 和 $g(x, y)$ 的灰度等级也都为 L 。将 $h(x, y)$ 、 $f(x, y)$ 以及 $g(x, y)$ 组成的三元组 (i, j, k) 称作三维直方图,如图 1(a) 所示。对于直方图上某一点代表的向量 (i, j, k) ,其出现的概率是 $p_{i,j,k}$,计算公式为:

$$p_{i,j,k} = \frac{q_{i,j,k}}{m \times n} \quad (3)$$

其中, $q_{i,j,k}$ 代表 (i, j, k) 发生的次数, $0 \leq i, j, k \leq$

$$L-1, \sum_0^{L-1} \sum_0^{L-1} \sum_0^{L-1} p_{i,j,k} = 1。$$

对于一个可行的三维阈值向量 (s, t, q) 可将三维

直方图分成八个部分,如图 1(b)~(d) 所示。其中,将区域 1 和 0 分别划分成图像的目标和背景,区域 2~7 可看成噪声和边缘。由于目标和背景区域的像素点多于边缘和噪声区域的像素点,从而将区域 2~7 的像素点忽略不计。

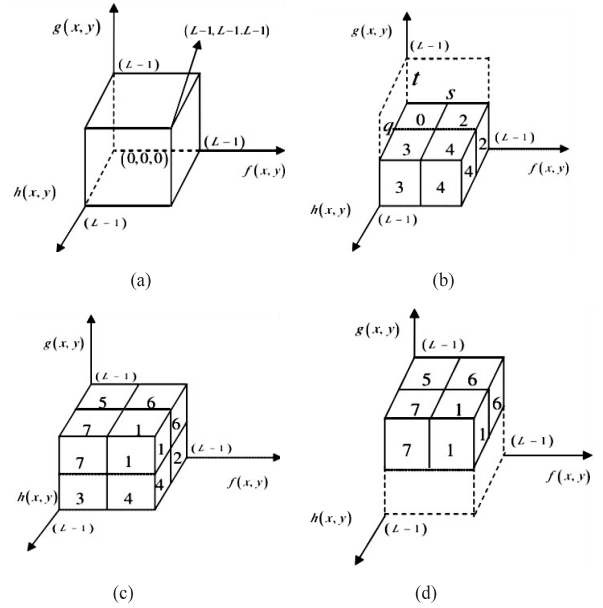


图 1 三维 OTSU 直方图

令 C_1 和 C_0 分别代表目标和背景这两个区域,则目标与背景发生的概率如下:

$$p_1 = \sum_{(i,j,k \in C_1(s,t,q))} p_{i,j,k} \quad (4)$$

$$p_0 = \sum_{(i,j,k \in C_0(s,t,q))} p_{i,j,k} \quad (5)$$

目标与背景均值矢量分别为 μ_1 和 μ_0 :

$$\mu_1 = (\mu_{1i}, \mu_{1j}, \mu_{1k})' = \left(\frac{\sum_{(i,j,k \in C_1(s,t,q))} ip_{i,j,k}}{p_1}, \frac{\sum_{(i,j,k \in C_1(s,t,q))} jp_{i,j,k}}{p_1}, \frac{\sum_{(i,j,k \in C_1(s,t,q))} kp_{i,j,k}}{p_1} \right) \quad (6)$$

$$\mu_0 = (\mu_{0i}, \mu_{0j}, \mu_{0k})' = \left(\frac{\sum_{(i,j,k \in C_0(s,t,q))} ip_{i,j,k}}{p_0}, \frac{\sum_{(i,j,k \in C_0(s,t,q))} jp_{i,j,k}}{p_0}, \frac{\sum_{(i,j,k \in C_0(s,t,q))} kp_{i,j,k}}{p_0} \right) \quad (7)$$

直方图总均值矢量 μ_T 为:

$$\mu_T = (\mu_{Ti}, \mu_{Tj}, \mu_{Tk})^T = \left(\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} ip_{i,j,k}, \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} jp_{i,j,k}, \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} kp_{i,j,k} \right)^T \quad (8)$$

由于区域 2~7 的像素点几乎为 0, 则:

$$p_0 + p_1 = 1 \quad (9)$$

$$p_{0\mu_0} + p_{1\mu_1} = \mu_T \quad (10)$$

那么目标与背景区域之间的离散度矩 S_B 为:

$$S_B = p_0 [(\mu_0 - \mu_t)(\mu_0 - \mu_t)^T] + p_1 [(\mu_1 - \mu_t)(\mu_1 - \mu_t)^T] \quad (11)$$

将离散度矩阵迹的值 $\text{Tr}S_B$ 作为两类区域的类间距离度量函数:

$$\text{Tr}S_B = p_0 [(\mu_{0i} - \mu_{Ti})^2 + (\mu_{0j} - \mu_{Tj})^2 + (\mu_{0k} - \mu_{Tk})^2] + p_1 [(\mu_{1i} - \mu_{Ti})^2 + (\mu_{1j} - \mu_{Tj})^2 + (\mu_{1k} - \mu_{Tk})^2] \quad (12)$$

当 $\text{Tr}(S_B(s^*, t^*, q^*)) = \max\{\text{Tr}(S_B(s, t, q))\}$, $0 \leq s, t, q \leq L-1$ 时, 对应的 (s^*, t^*, q^*) 为图像最佳分割阈值。

2 基于改进狼群算法的三维 OTSU 法

2.1 狼群算法原理

狼群算法是通过个体狼与整个狼群进行信息交互来搜索全局最优解^[10-11]。一个狼群体系包括头狼、猛狼以及探狼这三种成员。头狼负责指挥其他个体狼搜索猎物。探狼随机游走去搜寻猎物, 根据猎物气味浓度判断猎物源的远近, 浓度越大(离散度矩阵迹的值越大)则狼离猎物越近。探狼发现猎物会马上通知头狼, 头狼向猛狼发起召唤, 猛狼收到指令进行围攻追捕猎物。为保证种群的生存和发展, 狼群会不断更新群体, 除去瘦弱的个体狼, 保存种群实力。

2.2 改进的狼群算法

2.2.1 花授粉算法优化狼群算法初值

花授粉算法是由 Yang Xinshe 于 2012 年提出的。由于该算法参数少、结构简单、全局搜索性能良好, 将其引入到狼群算法中, 提高狼群算法的全局寻优能力。

在花授粉算法初始时, 随机产生 M 个种群个体 $p(t) = \{x_i^t\}$, 其中 $x_i^t = [x_{i,1}^t, x_{i,2}^t, \dots, x_{i,j}^t, \dots, x_{i,Q}^t]$, $i=1, 2, \dots, M; Q$ 为优化问题的维度, $j=1, 2, \dots, Q; t$ 是迭代次数。花授粉算法利用转换概率 $p \in [0, 1]$ 来把来自花授粉和交叉授粉之间的转换。

自花授粉为局部授粉, 其过程为:

$$x_i^{(t+1)} = x_i^{(t)} + \varepsilon * (x_j^{(t)} - x_i^{(t)}) \quad (13)$$

其中, i, j 为 $[1, M]$ 区间内的随机整数, ε 为变异因子, 是 $[0, 1]$ 上服从均匀分布的随机数。

交叉授粉为传粉者采用莱维飞行的全局授粉, 其过程为:

$$x_i^{(t+1)} = x_i^{(t)} + \gamma L(x_i^{(t)} - g^*) \quad (14)$$

其中, γ 为步长控制因子, g^* 为当前群体最佳值, 参数 L 是步长, 服从莱维分布。

转换概率 p 的计算公式为:

$$p = 0.8 + 0.2 \times \text{rand} \quad (15)$$

其中, rand 为 $[-1, 1]$ 之间的随机数。

花授粉算法主要步骤如下:

(1) 初始化参数。设置群体数量 M 和种群个体 $p(t)$, 优化问题维度 Q , 迭代次数 t , 变异因子 ε , 步长控制因子 γ 。

(2) 计算个体适应度值, 确定全局最佳适应度值和对应的最佳位置 g^* 。

(3) 根据式(15)计算转换概率 p 。

(4) 在 $[0, 1]$ 区间生成一个随机数 rand_1 , 若 $p \leq \text{rand}_1$, 则根据式(13)进行局部授粉。

(5) 若 $p > \text{rand}_1$, 则根据式(14)进行全局授粉。

(6) 判断算法是否满足循环结束的条件, 若满足则输出结果; 否则, 返回步骤(2)。

狼群算法初始时位置是随机的, 若位置分布得均匀, 则容易寻找全局最佳值; 若位置分布得不均匀, 则容易陷入局部极值, 降低寻优准确率。为了提高寻优精度, 加快全局搜索速度, 将花授粉算法求解的较优值作为狼群算法的初始值。

具体实现过程如下: 令个体狼的总数为 N , 待求解的阈值变量数为 D , 那么狼群被抽象在 $N \times D$ 的欧氏空间中。设待寻优的初始阈值向量为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, 则 $Y_i = f(X_i)$ (Y 是目标函数, 即离散度矩阵迹的值) 为每条人工狼感知的猎物气味浓度。将花授粉算法求得的最佳解对应的位置加入到狼群算法的初始位置。花授粉算法中, 若狼群算法的群体数量 N 大于 M , 可将花授粉算法求解的最佳位置全部随机加入狼群算法的初始位置; 否则, 只将花授粉算法求解的个别最佳位置随机加入狼群算法的初始位置。

待初始位置确定后, 选出头狼(具有最优目标函数的人工狼)记作 Y_{lead} 。头狼不参加游走行为、召唤行为和围攻行为, 它直接进入下一轮迭代, 直到其被更凶猛的人工狼取代。

2.2.2 游走过程

先选取 L_Num 只人工狼作为探狼, L_Num 取 $[n/(\alpha + 1), n/\alpha]$ 中的整数(α 是探狼比因子)。探狼群体从各自的当前位置随机向 m 个方向游走 step_a (游走步长), 同时记录游走后的猎物气味浓度(目标函数值)并回到初始位置。向第 $h(h=1, 2, \dots, m)$ 个方向游走后, 探狼 i 在第 d 维求解空间中的位置公式如下:

$$x_{id}^h = x_{id} + \sin(2\pi \times h/m) * \text{step}_a^h \quad (16)$$

$Y_{ih}(h=1, 2, \dots, m)$ 为探狼 i 感知的猎物气味浓度。探狼个体在 h 个方向中选择气味浓度最强的方向游走, 对探狼的位置 X_i 和感知的气味浓度 $Y_i = \max\{Y_{ih}\}$ 进行更新。探狼群体重复上述游走行为, 当某只个体狼 i 感知的气味浓度 $Y_i > Y_{\text{lead}}$ 时, 将这只人工狼作为头狼发起召唤, 否则重复游走行为直到 T

(游走次数)达到上限。

2.2.3 召唤过程

将余下的($N-1-L_Num$)只人工狼作为猛狼。猛狼听到头狼的召唤后迅速追捕 $step_b$ (奔袭步长)逼近猎物源,那么对于第 $k+1$ 次迭代第 i 只猛狼在第 d 维求解空间中的位置公式如下:

$$x_{id}^{k+1} = x_{id}^k + step_b^d * (g_d^k - x_{id}^k) / |g_d^k - x_{id}^k| \quad (17)$$

其中, g_d^k 为第 k 代头狼在第 d 维求解空间的位置。追捕过程中,如果猛狼嗅到的气味浓度 $Y_i > Y_{lead}$, 那么 $Y_{lead} = Y_i$, 将该猛狼作为头狼进行召唤,反之继续追捕直至其与头狼 Q 之间的距离 d_{iq} 小于 d_{nr} 时进入围攻行为。 d_{nr} 的计算公式如下:

$$d_{nr} = \frac{1}{D \times w} * \sum_{d=1}^D |x_{maxd} - x_{mind}| \quad (18)$$

其中, x_{maxd} 与 x_{mind} 分别是第 d 维求解空间的上下限, w 是对距离判定的因子。

2.2.4 改进的围攻过程

令 G_d^k 为第 k 次迭代,猎物在第 d 维求解空间里的位置,那么围攻过程的公式如下:

$$x_{id}^{k+1} = x_{id}^k + \lambda * step_c^d * |G_d^k - x_{id}^k| \quad (19)$$

其中, $step_c$ 为围攻步长, λ 为随机步长因子 ($[-1, 1]$ 中均匀分布的随机数)。如果探狼和猛狼嗅到的猎物气味浓度超过它们原位置嗅到的气味浓度,则对人工狼位置进行更新,反之位置不变。 $step_a$ 、 $step_b$ 以及 $step_c$ 之间关系如下:

$$step_a^d = step_b^d / 2 = 2 * step_c^d \quad (20)$$

$$step_a^d = |x_{maxd} - x_{mind}| / s \quad (21)$$

原始围攻过程中,狼群搜索速度逐渐趋于 0,使得算法出现“假死”状态,降低算法搜索效率。为此,将高斯变异引入围攻行为,使得算法后期能够从局部最佳值附近深度搜索全局最佳值,加快寻优速度,并减少狼群陷入局部最佳的次数。

根据高斯分布概率密度函数构造高斯变异函数:

$$GMF_i(t) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x_* - x_i^t)^2}{2\sigma^2}} \quad (22)$$

其中, σ 为高斯分布的标准差, x_* 为全局最佳值。通过 $GMF_i(t)$ 的扰动,可使狼群在围攻行为中跳出局部极值。根据高斯变异移动的位置公式如下:

$$x_i^{(t)} = GMF_i^{(t)} * x_i^{(t)} \quad (23)$$

设置变异函数是为了让所有个体狼都有机会发生变异移动。变异函数的公式如下:

$$switch(i) = \varphi + \mu * \frac{e^{\frac{n(i-1)}{N}} - 1}{e^n - 1} + rand - 1 \quad (24)$$

其中, N 为种群数量, n 为函数维度。当 $0 < switch(i) < 1$ 时,对第 i 个个体狼进行变异,否则

不变。

仿真实验表明,当 $\mu \in [0.1, 0.5]$, $\varphi \in [0.4, 0.6]$ 时,会有 $1/2$ 的个体狼随机发生变异,此时算法的搜索能力最佳。

2.2.5 群体更新

整个狼群根据“强多弱少”的原则分配猎物。将算法中目标函数值最差的 L 匹人工狼除去,并随机生成 L 匹狼, L 取 $[n/(2 \times \xi), n/\xi]$ (ξ 为狼群更新因子)内的随机数。

2.3 改进狼群算法优化的三维 OTSU 法

2.3.1 改进算法原理

利用改进的狼群算法优化三维 OTSU 算法的本质即人工狼个体表示一个可行的三维阈值向量^[12]。在像素点灰度以及该像素点邻域灰度均值和中值构成的三维向量空间中^[13-14],狼群通过游走、召唤、围攻、种群更新这几种行为的不断迭代以及狼群间的信息交互来求取一组最优解 (s^*, t^*, q^*),使得式(12)的值最大即以目标函数最大化为基础来寻找最优阈值。

2.3.2 改进算法步骤

(1)利用花授粉算法求取最佳解和对应的最佳位置。

(2)设置狼群算法参数。确定人工狼位置 X_i 以及人工狼的数目 N ,迭代次数上限 k_{max} 和游走次数上限 T_{max} ,探狼比因子 α ,步长因子 s ,距离判定因子 w 以及种群更新因子 ξ 。

(3)种群初始化。将花授粉算法得到的最佳位置加入狼群算法的初始位置。

(4)人工狼的位置为初始化的二维阈值向量,通过式(12)计算人工狼嗅到的猎物气味浓度。根据气味浓度大小选出头狼、探狼以及猛狼。

(5)若某只探狼嗅到的气味浓度 $Y_i > Y_{lead}$,将这只探狼作为头狼或达到游走次数上限 T_{max} ,转步骤(6)。

(6)猛狼听到头狼召唤追捕猎物。如果其嗅到的气味浓度 Y_i 大于头狼 Y_{lead} ,则 $Y_{lead} = Y_i$,该猛狼作为头狼召唤狼群。如果 $Y_i \leq Y_{lead}$,那么猛狼继续追捕直至 $d_{is} \leq d_{near}$,转步骤(7)。

(7)根据公式(24)计算变异函数,若 $0 < switch(i) < 1$,则根据式(23)进行高斯变异,否则根据式(19)进行位置更新。

(8)计算人工狼个体对应的目标函数值 Y_i ,同时对狼群群体进行更新。

(9)判断算法是否达到寻优精度或最大迭代次数,如果达到则将头狼位置作为图像最优分割阈值,同时二值化分割图像,否则返回步骤(3)。

3 实验仿真与分析

实验运行环境为 Windows 10 系统下的 MATLAB R2016a。首先,选取 Benchmarks 函数集中的 Sphere 函数和 Griewank 函数对改进的狼群算法进行寻优性能测试。测试函数如表 1 所示。

表 1 测试函数

函数名	函数表达式	全局极值
Sphere	$f_1(x) = \sum_{i=1}^d x_i^2$	$f_{1min}(0,0,\cdots,0) = 0$
Griewank	$f_2(x) = \frac{1}{4\,000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos(\frac{x_i}{\sqrt{i}}) + 1$	$f_{2min}(0,0,\cdots,0) = 0$

由于算法的随机性会对计算结果产生较大的影响,因此,对于标准狼群算法(WPA)和改进狼群算法分别独立运行 20 次。参数设置:狼群群体个数 N 为 30,向量空间维数 D 为 10;游走次数上限 $T_{max} = 10$;探狼比因子 $\alpha = 4$,更新因子 $\zeta = 6$,步长因子 $s = 1\,000$,距离判定因子 $w = 600$ 。表 2 是两种算法在不同测试函数下的最优值和平均值。图 2、3 分别是两种算法在 Sphere 函数和 Griewank 函数上寻优的收敛性能对比。

表 2 函数测试结果

函数	测试值	WPA	文中算法
f_1	最优值	1.703e-07	1.319 2e-10
	平均值	1.301 2e-06	8.807 4e-09
f_2	最优值	1.209e-08	1.923 1e-11
	平均值	1.327 6e-07	1.820 2e-10

通过对比表 2 中两个测试函数的最优值及平均值可知,利用改进的狼群算法求解的全局最优值更加接近两个测试函数的实际最小值,改进算法的求解精度要比标准狼群算法好,寻优能力也更强。

从图 2、3 中可发现,改进狼群算法计算全局最优解时所需的迭代次数比标准狼群算法要小,收敛性能更好,能够快速求解最佳值。其中适应度值的指数为求取两个函数实际最小值的指数。因此,可以使用改进狼群算法对三维 OTSU 法进行优化。

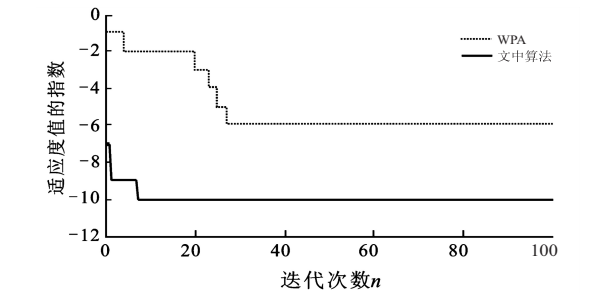


图 2 Sphere 函数的收敛曲线

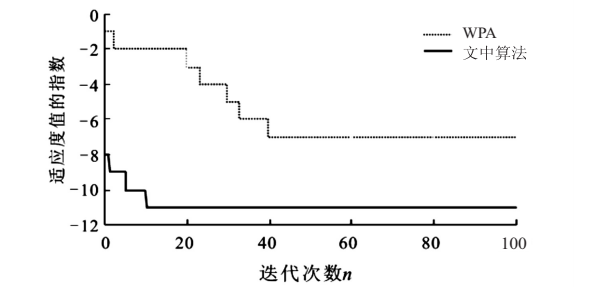


图 3 Griewank 函数的收敛曲线

为验证提出算法的分割效果,将文中算法与传统三维 OTSU 算法以及文献[10]提出的 CWPA 算法优化三维 OTSU 法进行图像分割对比。实验选择三幅图片 Bubbles、Baboon 和 Couple,均为 512×512 像素。狼群群体规模 N 设置为 50,空间维数 D 为 3,最大迭代次数 $K = 100$,CWPA 算法的信仰空间接受更新的概率门限为 0.7。分割效果如图 4 所示,运行结果如表 3 ~ 表 5 所示。

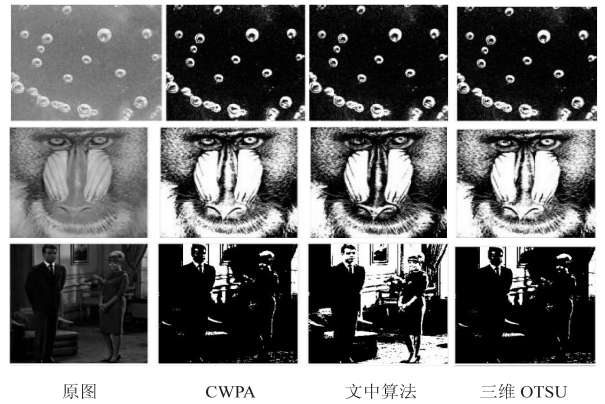


图 4 三种算法的分割结果

表 3 分割阈值对比

图像	CWPA	文中算法	三维 OTSU
Bubbles	(130,238,259)	(128,224,236)	(126,225,238)
Baboon	(113,127,130)	(110,122,125)	(113,123,128)
Couple	(124,142,198)	(120,139,194)	(128,150,201)

从表中可观察到,对于 Bubbles 图片,文中算法的分割阈值与三维 OTSU 算法相差不大。而对于其他两张图片,文中算法的分割阈值均小于另外两种分割算法。由于算法将高斯变异引入围攻行为,不易陷入局部最优,能精准找到最佳阈值。

表 4 分割用时对比

图像	CWPA	文中算法	三维 OTSU
Bubbles	33.96	22.37	160.97
Baboon	40.89	39.61	130.89
Couple	59.42	33.27	120.73

从表中可看出,在时间消耗方面,三维 OTSU 算法用时最长,文中算法和 CWPA 算法用时较短。这是因

为文中算法采用花授粉法优化狼群算法后,加快了全局搜索速度。

表5 区域一致性对比

图像	CWPA	文中算法	三维 OTSU
Bubbles	0.891	0.928	0.929
Baboon	0.912	0.983	0.916
Couple	0.864	0.937	0.861

区域一致性反映了算法对图像的分割效果。在同一类别区域内,像素相似度越大,表明一致性越好,分割效果越佳。区域一致性可定义为:

$$V=1-\frac{1}{C}\sum\left\{\sum_{(x,y\in R_i)}[f(x,y)-\frac{1}{I_i}\sum_{(x,y\in R_i)}f(x,y)]^2\right\}$$

(25)

其中, R_i 是分割完的第 i 个区域, I_i 是区域 i 的面积, C 是图像的像素点总个数, $f(x,y)$ 是图像 (x,y) 处的灰度值。区域内部的均匀性越好则 V 越大(V 的范围在 $[0,1]$ 内)。从表5观察到,对于 Bubbles 图片,三维 OTSU 算法的区域一致性较好,但分割时间过长。而对于另外两张图片,文中算法的区域一致性均优于 CWPA 算法和三维 OTSU 法。因此,文中算法不仅减少了搜索阈值的运行时间,而且提高了分割图像的精度。

4 结束语

三维 OTSU 法引入了邻域中值及均值信息,较好地解决了三维 OTSU 法对噪声敏感的问题,但该算法计算量庞大,计算复杂度较高。为此,提出应用改进的狼群算法来搜索最优三维阈值向量。实验结果表明,该算法使计算量大大减少,比 CWPA 算法的求解能力更强,求解精度显著提高。

参考文献:

[1] OTSU N. A threshold selection method from gray-level his-

togram[J]. IEEE Transactions on Systems, Man and Cybernetic, 1979, 9(1): 62-66.

[2] 彭启伟, 罗旺, 冯敏, 等. 改进二维 Otsu 法和果蝇算法结合的图像分割方法[J]. 计算机应用, 2017, 37(S2): 193-197.

[3] GUO W, WANG X, XIA X. Two-dimensional Otsu's thresholding segmentation method based on grid box filter[J]. International Journal for Light and Electron Optics, 2014, 125(18): 5234-5240.

[4] 景晓军, 李剑峰, 刘郁林. 一种基于三维最大类间方差的图像分割算法[J]. 电子学报, 2003, 31(9): 1281-1285.

[5] 范九伦, 赵凤, 张雪峰. 三维 OTSU 阈值分割方法的递推算法[J]. 电子学报, 2007, 35(7): 1398-1402.

[6] 叶志伟, 徐炜, 赵伟, 等. 一种基于改进萤火虫算法的三维 Otsu 阈值法[J]. 中国体视学与图像分析, 2016, 21(4): 374-380.

[7] 仇国庆, 熊耕耘, 赵文铭. 一种改进的三维 Otsu 图像分割算法[J]. 计算机科学, 2018, 45(8): 247-252.

[8] 吴虎胜, 张凤鸣, 吴庐山. 一种新的群体智能算法—狼群算法[J]. 系统工程与电子技术, 2013, 35(11): 2430-2438.

[9] YANG X S. Flower pollination algorithm for global optimization[C]//Proceeding of the 11th international conference unconventional computation and natural computation. Berlin: Springer-Verlag, 2012: 240-249.

[10] 钱荣鑫. 一种基于文化机制的狼群算法[J]. 信息技术, 2015, 35(12): 98-102.

[11] 王水花, 张煜东, 吉根林. 群智能算法的理论及应用综述[J]. 南京师范大学学报: 工程技术版, 2014, 14(4): 31-38.

[12] 刘建庄, 栗文青. 灰度图像的二维 Otsu 自动阈值分割方法[J]. 自动化学报, 1993, 19(1): 101-105.

[13] XUE J, TITTERINGTON D M. Median-based image thresholding[J]. Image and Vision Computing, 2011, 29(9): 631-637.

[14] SHAHID M, TAJ I A. Robust retinal vessel segmentation using vessel's location map and frangi enhancement filter[J]. IET Image Processing, 2018, 12(4): 494-501.