

基于离散灰狼算法的多级阈值图像分割

王 钰¹, 许 斌², 李林国², 齐 晋²

(1. 南京邮电大学 自动化学院, 江苏 南京 210003;

2. 南京邮电大学, 江苏 南京 210003)

摘要: 阈值分割方法的关键在于阈值选取。阈值决定了图像分割结果的好与坏, 随着阈值数量的增加, 图像分割的计算过程越来越复杂。为了选取适当的阈值进行图像分割, 文中提出了离散灰狼算法(Discrete Grey Wolf Optimizer, DGWO), 即经过离散化处理的灰狼算法, 并用该算法求解以 Kapur 分割函数为目标函数的全局优化问题。DGWO 算法具有很好的全局收敛性与计算鲁棒性, 能够避免陷入局部最优, 尤其适合高维、多峰的复杂函数问题的求解, 并且可以很好地融合到图像分割过程当中。大量的理论分析和仿真实验的结果表明, 与遗传算法(GA)、粒子群算法(PSO)的图像分割结果相比, 在选取多张分割图像、多个分割阈值的情况下, 该算法具有更好的分割效果, 更高的分割效率, 优化得到的阈值范围更加稳定, 分割质量更高。

关键词: 图像分割; 优化算法; 离散灰狼算法; Kapur 熵

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2016)07-0030-06

doi: 10.3969/j.issn.1673-629X.2016.07.007

A Multi-threshold Image Segmentation Algorithm Based on Discrete Grey Wolf Optimization

WANG Tai¹, XU Bin², LI Lin-guo², QI Jin²

(1. College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2. Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: The key of threshold segmentation is to select the thresholds which can determine the result of segmentation. With the increasing amounts of thresholds, the computation complexity gets higher. In this paper, a Discrete Grey Wolf Optimization (DGWO) is proposed to select the appropriate thresholds for image segmentation and apply it to the global optimization problem of objective function of Kapur segmentation function. The DGWO can be well blended into image segmentation. It specially suits for solving complex function with high-dimension and multi-peak for its excellent performance in global convergence, robustness and ability to avoid trapping into local optimization. Extensive theoretical analysis and the results of simulation have shown that DGWO has better effectiveness, efficiency, stability of the range of thresholds and quality in multi-images and multi-thresholds segmentation compared with GA and PSO.

Key words: image segmentation; optimization algorithm; Discrete Grey Wolf algorithm (DGWO); Kapur entropy

1 概述

图像是人类接收、处理和传递信息的一种重要途径。由于图像信息有着直观、易懂、形象以及信息量庞大的优点, 它也是人们生活与生产过程中最不可或缺和最高效的信息获取与交流的方式之一。图像分割是目前为止帮助人们获取有用信息最有效的方法。图像分割是指将要处理的目标图像划分成若干个特定的具有独特性质的区域并且提取出有用的或者感兴趣的目標的技术与过程。

图像分割技术的研究已经涉及众多领域, 它的深入研究和最终解决, 能够极大地推动众多学科的发展与成熟。近年来, 国内外都对其进行了大量的研究, 新的分割技术和方法层出不穷。但是, 依然没有具有普遍适用性的分割理论。图像分割算法有很多种, 经典算法有基于阈值的分割算法、基于边缘的分割算法、基于区域的分割算法以及与其他特定理论结合的图像分割算法。而阈值分割法是图像分割中最常用的方法, 通过选取适当的分割阈值, 把图像中的目标与非目标

收稿日期: 2015-10-23

修回日期: 2016-01-27

网络出版时间: 2016-06-22

基金项目: 国家自然科学基金资助项目(61401225); 南京邮电大学引进人才科研启动基金(NY213047, NY214102, NY213050, NY214098)

作者简介: 王 钰(1986-), 男, 硕士研究生, 研究方向为智能系统应用; 许 斌, 博士, 讲师, 研究方向为智能计算、云制造服务组合。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160622.0844.022.html>

分割开,为后续工作提供依据。

Seyedali 等^[1]模仿自然界中灰狼的领导等级制度和狩猎行为在 2014 年提出了一种新型算法——离散灰狼算法(DGWO)。根据灰狼群的等级制度将狼群分成 $\alpha, \beta, \delta, \omega$ 四个等级。其中, α 是领导狼(首领),等级依次降低, ω 等级最低。此外,狩猎行为有三个主要步骤:搜寻、包围和攻击。与其他经典的智能算法如 PSO、ABC、GA 等相比, DGWO 算法有较好的优化效果和收敛性,拥有强大的探索和开发能力,能够有效避免陷入局部最优,而且在解决未知空间和具有挑战的问题中也有良好效果,对于约束问题和非约束问题同样具有高度绩效,尤其适用于解决实际问题。Nipotepat 等^[2]提出了一种改进的灰狼算法(IGWO),应用了参数选择策略使搜索能力加强,采用杂交策略增加了代理的多样性。

文中运用 DGWO 来确定最优阈值,进而达到成功分割图像的目的。该算法应用到图像分割上能够迅速选取合适的阈值使得分割效果更好,提高了分割效率和分割精度,从而有利于后续工作的高效进行。实验结果表明了 DGWO 的优越性能。

2 相关工作

图像分割发展至今,已经有很多学者研究出了多种多样的实现方法,而由于图像的种类繁多,图像数据通常还有着许多不确定性,所以迄今为止也还不存在某一种通用的图像分割算法可以准确地依照人们的理解来分割任一幅图像。一些算法能直接分割一般图像,而有一些却只能分割一些特定的图像。到目前为止,已经有很多经典优化算法被运用到图像分割中,主要用于解决阈值的选择问题,并且也获得了较为理想的效果。遗传算法(GA)^[3]就被多次应用到分割工作中。李康顺等^[4]提出一种改进遗传算法(IGA)的图像分割方法。该方法能够根据个体适应度大小和种群分散程度自适应调整遗传控制参数,保持了种群的多样性,也提高了收敛速度。受到群体行为启发的进化算法如粒子群算法(PSO)^[5]和人工蜂群算法(ABC)^[6],也是广泛采用到图像分割问题中的经典优化算法。这两种算法是将 Kapur 熵函数作为适应度函数寻找最优的阈值进行分割的。张小峰^[7]将模糊聚类算法(FCA)应用到医学图像分割中。该算法是一种典型的“软分割”方法,它允许像素以相异的隶属度同时隶属于相异的组织与器官,能够有效解决医学图像中的部分容积效应现象问题,能够从原始图像中尽可能多地保留信息,因此可以获得较好的分割效果。Diego 等^[8]将 EMO 算法和 HSA 算法^[9]用于多级阈值转换优化分割问题中。当然还有许多优化算法也被研究者们

应用到这类问题当中,并且也取得了较为满意的效果,如 DE、CS、FA、HBM、BF 等^[10-13]。

文中采用 DGWO 进行图像分割,解决阈值选择的问题,把 Kapur 熵函数作为适应度函数进行优化,使得分割效果更好,效率更高,阈值的范围较稳定。

3 问题描述

3.1 熵标准法图像分割

熵标准法是 Kapur 提出的非参数阈值寻优方法^[14]。它是一种依据灰度直方图概率分布的基于熵阈值转换法的有效分割技术。当分离类的最佳阈值被正确分配时,熵是最大的。图像的熵测量类之间的紧凑性和分离性。该方法的目的就是寻找最优阈值产生最大熵。

从灰度图像或一个 RGB(红色、绿色、蓝色)图像各分量中取亮度水平 L ,亮度值的概率分布计算如下:

$$P_{h_i^a} = \frac{h_i^a}{NP}, \sum_{i=1}^{NP} P_{h_i^a} = 1, a = \begin{cases} 1, 2, 3 & \text{if RGB Image} \\ 1 & \text{if Gray scale Image} \end{cases} \quad (1)$$

式中, i 是一个特定的亮度级,取值范围为 $0 \leq i \leq L - 1$; a 是取决于图像是否为亮度图像或 RGB 图像的图像组件; NP 是图像的像素总数; h_i^a (直方图)是亮度级 i 在 a 中的像素个数;直方图是标准化的概率分布 $P_{h_i^a}$ 。

对于多个类(多级)的定义如下:

$$\begin{aligned} C_1 &= \frac{P_{h_1^a}}{\omega_0^a(\text{th})}, \dots, \frac{P_{h_n^a}}{\omega_0^a(\text{th})} \\ C_2 &= \frac{P_{h_{n+1}^a}}{\omega_1^a(\text{th})}, \dots, \frac{P_{h_m^a}}{\omega_1^a(\text{th})} \\ &\vdots \\ C_k &= \frac{P_{h_{m+1}^a}}{\omega_{k-1}^a(\text{th})}, \dots, \frac{P_{h_n^a}}{\omega_{k-1}^a(\text{th})} \end{aligned} \quad (2)$$

其中, $\omega_0(\text{th}), \omega_1(\text{th}), \dots, \omega_{k-1}(\text{th})$ 分别是 C_1, C_2, \dots, C_k 的概率分布,由式(3)计算:

$$\begin{aligned} \omega_0^a(\text{th}) &= \sum_{i=1}^{th_1} P_{h_i^a} \\ \omega_1^a(\text{th}) &= \sum_{i=th_1+1}^{th_2} P_{h_i^a} \\ &\vdots \\ \omega_{k-1}^a(\text{th}) &= \sum_{i=th_{k-1}+1}^L P_{h_i^a} \end{aligned} \quad (3)$$

基于 Kapur 熵的多个阈值方法,有必要用阈值的相似数将图像分成 k 类。因此,目标函数为:

$$f(\mathbf{TH}) = \sum_{i=1}^k H_i^a, a = \begin{cases} 1, 2, 3 & \text{if RGB Image} \\ 1 & \text{if Gray scale Image} \end{cases} \quad (4)$$

式中, $\mathbf{TH} = [\text{th}_1, \text{th}_2, \dots, \text{th}_{k-1}]$ 是包含多个阈值的矢量。

每个熵分别与其对应的阈值进行计算,公式如下:

$$\begin{aligned} H_1^a &= \sum_{i=1}^{\text{th}_1} \frac{P_{h_i^*}}{\omega_0^a(\text{th})} \ln\left(\frac{P_{h_i^*}}{\omega_0^a(\text{th})}\right) \\ H_2^a &= \sum_{i=\text{th}_1+1}^{\text{th}_2} \frac{P_{h_i^*}}{\omega_1^a(\text{th})} \ln\left(\frac{P_{h_i^*}}{\omega_1^a(\text{th})}\right) \\ &\vdots \\ H_k^a &= \sum_{i=\text{th}_{k-1}+1}^L \frac{P_{h_i^*}}{\omega_{k-1}^a(\text{th})} \ln\left(\frac{P_{h_i^*}}{\omega_{k-1}^a(\text{th})}\right) \end{aligned} \quad (5)$$

其中, k 个类所产生的概率值 $\{\omega_0^a, \omega_1^a, \dots, \omega_{k-1}^a\}$ 是由式(3)得到。

最后,需要使用式(6)将像素分隔在各自的类中。阈值转换是一个把图像的像素依据灰度级 (L) 分成集或类的过程。这种分类必须选择一个阈值(th)或者遵循下面多级阈值转换公式准则:

$$\begin{aligned} C_1 &\leftarrow p \text{ if } 0 \leq p < \text{th}_1 \\ C_2 &\leftarrow p \text{ if } \text{th}_1 \leq p < \text{th}_2 \\ &\vdots \\ C_i &\leftarrow p \text{ if } \text{th}_i \leq p < \text{th}_{i+1} \\ &\vdots \\ C_n &\leftarrow p \text{ if } \text{th}_n \leq p < L - 1 \end{aligned} \quad (6)$$

其中, p 为灰度图像 I_g 中 $m \times n$ 个像素点中的一个,它代表着在 L 灰度级中的等级, $L = \{0, 1, \dots, L - 1\}$; C_i 为像素点 p 所属的类; $\{\text{th}_1, \text{th}_2, \dots, \text{th}_k\}$ 为不同的阈值。

多级或多阈值转换的问题都是选择正确识别类的阈值 th 。Kapur 熵法是众所周知的用来确定这些阈值的方法。该方法提出的目标函数必须是最大化的,以便找到最佳阈值。文中使用式(7)来获得 k 维最优阈值和最大化 Kapur's 目标函数。

$$\text{th} = \text{argmax}\left(\sum_{i=0}^{k-1} \omega_i^a(\text{th})\right) \quad (7)$$

3.2 DGWO 数学模型

3.2.1 社会等级

灰狼算法依据社会等级,把认为最适当的解作为 α ,第二、第三适当解作为 β 、 δ ,其余的候选解假设为 ω 。DGWO 算法的狩猎(优化)是在 α 、 β 、 δ 的指导下进行的,即 ω 跟随它们。

3.2.2 包围

捕猎过程中的包围是依据下列公式进行的:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (8)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (9)$$

其中, t 为迭代次数; \vec{A} 、 \vec{C} 为系数向量; \vec{X}_p 为猎物的位置; \vec{X} 为灰狼的位置。

\vec{A} 、 \vec{C} 由以下公式计算:

$$\vec{A} = 2e \cdot \vec{r}_1 - e \quad (10)$$

$$\vec{C} = 2 \vec{r}_2 \quad (11)$$

式中, e 在迭代过程中从 2 线性递减至 0; \vec{r}_1 、 \vec{r}_2 是 $[0, 1]$ 内的随机向量。

3.2.3 捕猎

捕猎行为通常是在 α 的领导下, β 、 δ 不定时参加。在抽象空间中很难确定最优解的位置。为了精确模拟灰狼的捕猎行为,把猎物的潜在位置信息知识交给 α 、 β 、 δ 。因此,保存目前为止获得前三个最优解,并迫使其他的搜索代理(含 ω) 根据最好搜索代理的位置来更新自己的位置。公式如下:

$$\begin{aligned} \vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}(t)|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}(t)| \\ \vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}(t)| \end{aligned} \quad (12)$$

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha(t) - \vec{A}_1 \cdot \vec{D}_\alpha, \vec{X}_2 = \vec{X}_\beta(t) - \vec{A}_2 \cdot \vec{D}_\beta, \vec{X}_3 = \\ &\vec{X}_\delta(t) - \vec{A}_3 \cdot \vec{D}_\delta \end{aligned} \quad (13)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (14)$$

3.2.4 攻击

在猎物停止移动时,灰狼群就会进行攻击以完成捕猎。在接近猎物的过程中通过减小 e 的值,使 \vec{A} 的波动幅度下降;即在迭代过程中 e 从 2 递减至 0 时, \vec{A} 取 $[-e, e]$ 内的随机值。

3.2.5 算法过程

初始化灰狼群的位置 $X_i(i = 1, 2, \dots, m)$

初始化 e 、 A 和 C

计算每个搜索代理的适应度值

\vec{X}_α = 最好搜索代理的位置

\vec{X}_β = 第二好搜索代理的位置

\vec{X}_δ = 第三好搜索代理的位置

While($t <$ 最大迭代次数)

for 所有的搜索代理

依据式(14)更新当前搜索代理的位置

end for

```
更新  $e$ 、 $A$  和  $C$ 
计算所有搜索代理的适应度值
更新  $X_\alpha, X_\beta, X_\delta$ 
 $t = t + 1$ 
end while
return  $X_\alpha$ 
```

4 基于 DGWO 的图像分割

DGWO 算法应用于图像分割主要是对 Kapur 熵函数进行优化,以期取得最优阈值来分割图像。因此,把 Kapur 熵函数作为算法的适应度函数。

Step1: 读取图像 J 进行判断,如果是 RGB 图像就分离成 J_R, J_G, J_B 三个通道;如果是灰度图像就存储到 J_{gr} 。 $a = 1, 2, 3$ 是 RGB 图像, $a = 1$ 是灰度图像。

Step2: 获得图像的直方图。灰度图像的为 h^{gr} , RGB 图像的为 h^R, h^G, h^B 。

Step3: 根据式(3)计算出各灰度值的概率分布和直方图。

Step4: 初始化 DGWO 的种群数 NP 和参数 e 、 A 、 C 和最大迭代次数 Max_iter。

Step5: 初始化灰狼种群位置 $X_i (i = 1, 2, \dots, NP)$ 。

Step6: 把 Kapur 熵函数作为适应度函数,计算出每个搜索代理的适应度值。

Step7: 根据适应度值的大小,把最好搜索代理的位置赋给 α 狼,把第二好搜索代理的位置赋给 β 狼,把第三好搜索代理的位置赋给 δ 狼。

Step8: 分别依据式(10)~(13)计算出 $\vec{A}_1, \vec{A}_2, \vec{A}_3$ 、 $\vec{C}_1, \vec{C}_2, \vec{C}_3$ 、 $\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$ 和 X_1, X_2, X_3 ,再根据式(14)更新当前搜索代理的位置。

Step9: 循环指针 q 增加 1;如果 $q \geq \text{Max_iter}$ 或者满足算法停止条件,完成迭代过程,跳至第十步;否则跳至第六步。

Step10: 把拥有最优目标函数值的狼的位置作为最优分割阈值。

Step11: 输出最优分割阈值、分割前后的图像。

5 实验

5.1 参数设置

文中将会用 8 张标准图像对 DGWO 算法进行测试。这些图像都是采用灰色图像以及同样的格式(JPGE)。

为了对算法进行分析,把 DGWO 与 GA 和 PSO 进行对比。为了避免结果的随机性,采用适当的统计度量来比较这些算法的效能。因此,所有算法对每张图执行 35 次,依据文献[8]测试阈值 $th = 2, 3, 4, 5$ 。每个实验的停止准则都为 150 次迭代, $NP = 50$ 。为了验证稳定性,在每次测试结束时根据式(15)计算标准差(STD)。

$$STD = \sqrt{\sum_{i=1}^{\text{Max_iter}} \frac{(\theta_i - \varepsilon)^2}{\text{Max_iter}}} \quad (15)$$

另外,峰值信噪比(PSNR)依据每个像素的均方差(MSE)来比较图像(图像分割)与基准图像(原图像)的相似性。

$$PSNR = 20 \log_{10} \left(\frac{255}{\text{MSE}} \right) \quad (16)$$

$$MSN = \sqrt{\frac{\sum_{i=1}^{ro} \sum_{j=1}^{co} (I_o^a(i,j) - I_{th}^a(i,j))^2}{ro \times co}} \quad (17)$$

其中, I_o^a 是原始图像; I_{th}^a 是分割图像; a 取决于图像(灰度或 RGB); ro, co 分别是图像的总行数和总列数。

5.2 结果

把 Kapur 熵函数作为目标函数来分析 DGWO 的性能。表 1 给出了 DGWO 对整组基准图像的实验结果。列出 PSNR、STD、阈值个数 k 、最优阈值 Thresholds 和目标函数均值 MEAN。

5.3 比较

比较 DGWO 与 GA 和 PSO 的分割结果,如表 2 所示。所有方法都要对每张图像运行 35 次;每张图像都分别计算 PSNR、STD 和目标函数的平均值,而且目标函数都为 Kapur 熵函数。

表 1 DGWO 对整组基准图像的实验结果

Image	k	Thresholds	PSNR	STD	MEAN
Camera man	2	7,111	14.279	0.769 9	14.634
	3	43,95,146	19.696	1.162 5	21.107
	4	24,62,108,150	20.809	1.456 4	24.927
	5	41,85,121,155,195	22.404	1.675 0	30.436
Lena	2	93,164	14.680	1.106 5	17.809
	3	74,123,170	17.416	0.429 4	22.074
	4	45,78,122,175	19.762	1.459 0	25.318
	5	47,92,127,164,197	21.299	1.546 8	29.252

续表 1

Image	k	Thresholds	PSNR	STD	MEAN
Baboon	2	79,142	16.024	0.098 3	17.679
	3	46,102,153	18.632	0.352 3	22.129
	4	42,86,129,170	20.480	0.855 3	26.194
	5	29,62,97,131,166	22.060	1.242 8	30.067
Butterfly	2	89,150	14.762	0.309 2	17.425
	3	73,114,158	17.873	0.787 9	21.585
	4	52,84,119,157	21.021	1.352 3	25.267
	5	19,48,82,120,163	21.485	1.334 7	29.492
Maize	2	87,166	13.950	0.939 2	18.604
	3	63,109,179	16.201	1.259 6	22.941
	4	48,95,145,187	18.713	1.260 2	26.936
	5	44,85,122,165,205	20.410	1.401 1	31.023
Sea star	2	60,141	14.809	1.248 6	18.321
	3	64,113,172	17.431	1.297 2	23.245
	4	40,85,126,186	19.421	1.204 3	27.136
	5	40,71,105,147,194	20.887	1.517 5	31.167
Smiling girl	2	92,144	17.989	0.508 3	17.136
	3	39,87,142	18.742	0.322 3	21.253
	4	21,54,100,146	19.823	1.305 0	25.050
	5	24,73,119,153,202	21.214	1.512 4	29.870
Surfer	2	53,150	16.154	0.768 7	18.283
	3	51,99,163	18.895	0.831 2	23.243
	4	44,84,123,194	20.234	1.256 3	27.275
	5	44,74,104,155,202	21.699	1.146 1	31.384

表 2 DGWO、GA 和 PSO 的图像分割结果对比

image	k	DGWO			GA			PSO		
		PSNR	STD	MEAN	PSNR	STD	MEAN	PSNR	STD	MEAN
Camera man	2	14.279	3.386e-14	14.634	11.941	0.127 0	15.341	12.259	0.100 1	16.071
	3	19.696	5.346e-14	21.107	14.827	0.213 6	20.600	15.211	0.110 7	21.125
	4	20.809	2.851e-14	24.927	17.166	0.285 7	24.267	18.000	0.200 5	25.050
	5	22.404	3.564e-15	30.436	19.795	0.352 8	28.326	20.963	0.273 4	28.365
Lena	2	14.680	7.129e-15	17.809	12.334	0.004 9	16.122	12.345	0.003 3	16.916
	3	17.416	3.208e-14	22.074	14.995	0.110 0	20.920	15.133	0.039 0	20.468
	4	19.762	0	25.318	17.089	0.259 4	23.569	17.838	0.181 0	24.449
	5	21.299	8.198e-14	29.252	19.549	0.304 3	27.213	20.442	0.218 1	27.526
Baboon	2	16.024	2.138e-14	17.679	12.184	0.056 7	16.425	12.213	0.007 7	16.811
	3	18.632	3.564e-15	22.129	14.745	0.158 0	21.069	15.008	0.081 6	21.088
	4	20.480	2.495e-14	26.194	16.935	0.176 5	25.489	17.574	0.085 3	24.375
	5	22.060	3.208e-14	30.067	19.662	0.277 5	29.601	20.224	0.189 9	30.994
Butterfly	2	14.762	4.634e-14	17.425	10.470	0.087 2	15.481	10.474	0.002 5	14.098
	3	17.873	5.703e-14	21.585	11.628	0.202 1	20.042	12.313	0.188 0	19.340
	4	21.021	7.485e-14	25.267	13.314	0.259 6	23.980	14.231	0.247 3	25.190
	5	21.485	7.485e-14	29.492	15.756	0.397 7	27.411	16.337	0.282 1	27.004
Maize	2	13.950	1.069e-14	18.604	13.506	0.072 5	18.521	13.466	0.001 2	18.631
	3	16.201	6.772e-14	22.941	15.150	0.158 2	23.153	15.018	0.053 0	23.259
	4	18.713	3.208e-14	26.936	15.909	0.269 7	26.798	15.834	0.142 4	27.470
	5	20.410	2.851e-14	31.023	16.921	0.897 1	30.852	16.319	0.498 0	31.255
Sea star	2	14.809	7.129e-14	18.321	14.282	0.081 6	18.753	14.346	0.000 2	18.593
	3	17.431	3.921e-14	23.245	8.2638	0.198 7	23.260	16.949	0.172 3	23.289
	4	19.421	7.129e-14	27.136	15.035	0.269 1	26.533	18.389	0.248 1	27.407
	5	20.887	1.782e-14	31.167	19.005	0.974 0	30.798	19.849	0.615 9	31.288

续表 2

image	k	DGWO			GA			PSO		
		PSNR	STD	MEAN	PSNR	STD	MEAN	PSNR	STD	MEAN
Smiling girl	2	17.989	7.129e-15	17.136	13.092	0.017 8	17.295	13.352	0.036 8	17.321
	3	18.742	4.634e-14	21.253	17.764	0.217 9	21.580	18.201	0.055 6	21.887
	4	19.823	6.059e-14	25.050	17.923	0.302 4	25.432	18.063	0.281 7	25.815
	5	21.214	1.069e-14	29.870	19.026	0.712 8	27.940	9.200	0.588 7	29.700
Surfer	2	16.154	5.346e-14	18.283	11.521	0.021 9	18.237	11.698	0.114 4	18.194
	3	18.895	2.495e-14	23.243	17.181	0.171 5	22.865	18.413	0.233 2	22.214
	4	20.234	9.980e-14	27.275	18.868	0.209 3	26.447	19.125	0.421 4	26.676
	5	21.699	7.129e-14	31.384	19.521	0.318 2	30.363	19.491	0.478 9	30.587

6 结束语

文中提出一种基于 DGWO 的多级阈值图像分割方法。将 DGWO 算法应用到多阈值图像分割领域,从而实现图像的多阈值分割。实验结果表明,基于 DGWO 的图像分割方法不仅能够获得更好的图像分割质量,而且在运行时间和收敛性方面相较 GA 和 PSO 方法具有明显优势。

参考文献:

- [1] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69:46-61.
- [2] Muangkote N, Sunat K, Chiewchanwattana S. An improved grey wolf optimizer for training q-Gaussian radial basis functional-link nets[C]//*Proc of computer science and engineering conference*. [s. l.]:[s. n.], 2014:209-214.
- [3] Pal S K, Bhandari D, Kundu M K. Genetic algorithms for optimal image enhancement[J]. *Pattern Recognition Letters*, 1994, 15(3):261-271.
- [4] 李康顺, 李茂民, 张文生. 一种基于改进遗传算法的图像分割方法[J]. *计算机应用研究*, 2009, 26(11):4364-4367.
- [5] 孙越泓. 基于粒子群优化算法的图像分割研究[D]. 南京: 南京理工大学, 2010.
- [6] Karaboga D, Gorkemli B. A quick artificial bee colony -qABC - algorithm for optimization problems[C]//*Proc of international symposium on innovations in intelligent systems and applications*. [s. l.]:[s. n.], 2012:1-5.
- [7] 张小峰. 基于模糊聚类算法的医学图像分割技术研究[D]. 济南: 山东大学, 2014.
- [8] Oliva D, Cuevas E, Pajares G, et al. A multilevel thresholding algorithm using electromagnetism optimization[J]. *Neurocomputing*, 2014, 139:357-381.
- [9] Oliva D, Cuevas E, Pajares G, et al. Multilevel thresholding segmentation based on harmony search optimization[J]. *Journal of Applied Mathematics*, 2013, 2013:575414.
- [10] Sathya P, Kayalvizhi R. Optimal multilevel thresholding using bacterial foraging algorithm[J]. *Expert Systems with Applications*, 2011, 38(12):15549-15564.
- [11] Sarkar S, Patra G R, Das S. A differential evolution based approach for multilevel image segmentation using minimum cross entropy thresholding[M]//*Swarm, evolutionary, and memetic computing*. Berlin: Springer, 2011:51-58.
- [12] Horng M H, Liou R J. Multilevel minimum cross entropy threshold selection based on the firefly algorithm[J]. *Expert Systems with Applications*, 2011, 38(12):14805-14811.
- [13] Agrawal S, Panda R, Bhuyan S, et al. Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm[J]. *Swarm and Evolutionary Computation*, 2013, 11(1):16-30.
- [14] Kapur J N, Sahoo P K, Wong A K. A new method for gray-level picture thresholding using the entropy of the histogram[J]. *Computer Vision, Graphics, and Image Processing*, 1985, 29(3):273-285.