基于改进萤火虫优化算法的多阈值彩色图像分割

毛 肖 和丽芳 王庆平

(昆明理工大学城市学院 昆明 650093)

摘 要 为了提高彩色图像的分割效果，提出一种基于改进的萤火虫优化(IGSO)算法的彩色图像多阈值分割方法，该方法以 Kapur 熵为目标函数。针对基本萤火虫优化(GSO)算法进化后期收敛速度慢和求解精度低的问题，采用自适应步长和添加全局信息两种策略，提出了一种改进的萤火虫优化(IGSO)算法。IGSO 算法根据步长和萤火虫的移动方向对萤火虫算法收敛性的影响，在萤火虫移动过程中引入全局信息，采用随着迭代次数和搜索空间维数自适应变化步长的策略，来提高收敛性能。实验结果表明，该方法能够较好地对彩色图像进行分割，其性能优于基本的萤火虫优化(GSO)算法、改进的量子行为粒子群优化算法(CQPSO)和改进的细菌觅食算法(MBF)。

关键词 萤火虫优化算法，彩色图像分割，自适应步长，全局信息

中图法分类号 TP319 **文献标识码** A

Multilevel Color Image Segmentation Based on Improved Glowworm Swarm Optimization Algorithm

MAO Xiao HE Li-fang WANG Qing-ping

(Faculty of City, Kunming University of Science and Technology, Kunming 650093, China)

Abstract In order to improve the segmentation effect of color image, a novel multilevel color image segmentation method was presented based on an improved glowworm swarm optimization (IGSO) algorithm which uses Kapur's entropy. Aiming at the problem that the glowworm swarm optimization algorithm has low convergence speed and accuracy in the later period, an improved glowworm swarm optimization (IGSO) algorithm was presented based on adaptive step and global information. Depending on the effect of step size and the direction of movement on the convergence, IGSO algorithm improves convergence by adding global information and adaptive step with iterations and dimension of the search space during the course of movement. The experimental results show that it is a better method for multilevel color image segmentation compared with GSO algorithm, improved quantum-behaved particle swarm optimization (CQPSO) algorithm and modified bacterial foraging (MBF) algorithm.

Keywords Glowworm swarm optimization algorithm, Color image segmentation, Adaptive step, Global information

1 引言

图像分割就是把图像分成若干个特定的、具有独特性质的区域并将感兴趣的目标提出的技术和过程[1]，在计算机视觉和模式识别中占有非常重要的地位。目前相关的文献已提出上千种图像分割算法，其中阈值分割是一种性能较好并且应用较广泛的图像分割方法。传统的阈值分割算法对于单阈值分割非常有效，但是随着阈值个数的增加，计算量会急剧增加。对于一幅给定的图像，搜索最佳阈值的过程可以被看作是一个约束优化问题，为了解决多阈值分割计算量大的问题，许多智能群体优化算法被用于图像的多阈值分割，如粒子群优化算法[2-4]、遗传算法[5-6]、差分算法[7-9]、人工蜂群算法[10-12]和布谷鸟搜索优化算法[13-15]等。由于彩色图像相对于灰度图像包含更多的信息，因此彩色图像多阈值分割成为近年来多阈值分割算法研究的热点。为了减少多阈值彩色图像分割的计算量，目前有许多相关文献将智能群体优化算法用于多阈值彩色图像的分割问题[16-20]。

萤火虫优化算法属于一种新型的智能群体优化算法，是近年来研究的热点，最早由印度学者 Krishnanand K N 和 Ghose D 于 2005 年的第二届国际人工智能会议上提出，它源于对萤火虫群觅食和吸引同伴行为的模拟[21-22]。该算法与蚁群算法、粒子群优化算法等智能群体优化算法相比，具有许多共同点，但最大的区别在于各个萤火虫个体都拥有自己的动态决策域以及只依赖局部信息进行搜索，最终将整个萤火虫群体分为多个子群体，收敛于多个局部最优解或全局最优解[23-24]。目前，萤火虫优化算法已被成功应用于群体机器人的协同运作[25]、移动信号源的跟踪[26]、无线传感器网络[27]等领域。此外，萤火虫优化算法还被用于灰度图像的多阈值分割[28-29]。

相对于其他智能群体优化算法，萤火虫优化算法在求解全局最优解时，存在后期搜索速度慢、求解精度低的缺点。为了提高算法的收敛性能，本文在萤火虫的移动过程中，首先借鉴其他智能群体优化算法中全局信息的思想，其次引入自适应步长的更新策略，提出了一种改进的萤火虫优化算法(IGSO)，以提高基本萤火虫优化算法的全局优化能力。最后，以 Kapur 熵为目标函数，将改进的萤火虫优化算法用于多阈值彩色图像分割，对多幅彩色图像进行分割，并将分割结果与基本的萤火虫优化算法、改进的量子行为粒子群优化算法

本文受国家自然科学基金项目(51204077)资助。

毛 肖(1980－)，男，硕士，讲师，主要研究方向为数字图像分割，E-mail:49005540@qq.com；和丽芳(1981－)，女，博士，副教授，主要研究方向为数字图像分割，E-mail:843168660@qq.com；王庆平 男，硕士，副教授，主要研究方向为数字图像分割，E-mail:70230261@qq.com。

(CQPSO)[30]和改进的细菌觅食算法(MBF)[31]进行比较。

2　基本萤火虫优化(GSO)算法

在基本萤火虫优化算法中，初始化阶段萤火虫被随机地散布在整个目标函数空间中，每只萤火虫携带有相同的荧光素，荧光素的强度由自己所在位置的目标函数值决定，荧光素的大小决定了萤火虫发出的荧光强度，并且每只萤火虫拥有相同的决策域，决策域受感应域的约束。在决策域中，每一只萤火虫受到荧光素强度比自己大的萤火虫的吸引并向其移动，通过不断迭代，最后绝大多数萤火虫会聚集到最优值。基本萤火虫算法包含 4 个步骤：萤火虫的初始化、荧光素的更新、萤火虫的移动和局部决策域的更新。

(1)萤火虫的初始化阶段：每只萤火虫携带有相同的荧光素和局部决策域。

(2)荧光素的更新阶段：萤火虫荧光素的强度由其所在位置的目标函数值决定，当其位置改变时，荧光素强度也会相应地改变。在荧光素更新阶段，每一只萤火虫都受到其决策域内亮度较高的萤火虫的吸引，并向其移动，因此萤火虫的荧光素需做相应的更新，荧光素的更新依据如下：

$$\ell_i(t+1)=(1-\rho)\ell_i(t)+\gamma J_i(t+1) \tag{1}$$

其中，ρ 为荧光素的延迟因子，γ 为荧光素的更新率。

(3)萤火虫的移动阶段：每只萤火虫在其决策域内选择荧光素强度比自己高的个体组成其邻域集，萤火虫 i 向其邻域集内的萤火虫 j 移动的概率为：

$$p_j(t)=\frac{(\ell_j(t)-\ell_i(t))}{\sum\limits_{k\in N_i(T)}(\ell_k(t)-\ell_i(t))} \tag{2}$$

萤火虫 i 的位置更新公式为：

$$x_i(t+1)=x_i(t)+s\left(\frac{x_j(t)-x_i(t)}{\|x_j(t)-x_i(t)\|}\right) \tag{3}$$

其中 s 为步长。

(4)局部决策域的更新阶段：在萤火虫移动的过程中，由于萤火虫向着荧光素强度比自己高的萤火虫移动，因此在每一次迭代过程中萤火虫的邻居数量发生着变化，此时局部决策域的半径大小需要根据邻居数量的变化做相应的调整，局部决策域更新规则如式(4)所示：

$$r_d^i(t+1)=\min\{r_s,\max\{0,r_d^i(t)+\beta(n_t-|N_i(t)|\}\} \tag{4}$$

其中，β 为常数，n_t 是一个阈值，用于控制邻域的数量。

3　改进的萤火虫优化(GSO)算法

3.1　GSO 算法位置更新方程的收敛性能分析

根据式(3)中萤火虫的位置更新公式可以推出：

$$x_i(t)=x_i(0)+s\sum_{m=1}^{t}\left(\frac{x_j(m)-x_i(m)}{\|x_j(m)-x_i(m)\|}\right) \tag{5}$$

其中，$x_i(0)$为萤火虫 i 的初始位置。从式(5)可以得出，萤火虫 i 在 t 时刻的位置 $x_i(t)$由 $x_i(0)$、步长 s 和矢量 $x_j(m)-x_i(m)$的方向决定。因此在局部寻优的过程中，为了保证萤火虫能够收敛于全局，全最优解、步长 s 和矢量 $x_j(m)-x_i(m)$的方向设置非常重要，其决定了 GSO 算法的收敛性。

3.2　自适应步长

步长 s 是一个关键的参数，其对 GSO 算法的收敛性有着非常重要的影响。在基本的 GSO 算法中，为了保证算法能够精确地收敛于最优解和收敛精度，步长被设置为一个固定值，并且要求步长小于 ε-distance，其中 ε 是一个极小值，这样会导致 GSO 算法的收敛速度非常慢[32]。

在 GSO 算法中，搜索初期，萤火虫与最优解间的距离较远，这时可以将步长设置为一个较大的值，以提高收敛速度；搜索后期，距离越来越近，需要逐步减小步长，保证步长小于 ε-distance，防止步长太大，跳过最优解，以保证收敛精度。因此，为了保证算法的收敛性，GSO 算法在寻优过程中需要动态调整步长的大小，步长是一个与迭代次数有关的函数，并且步长和迭代次数之间不是一种线性的关系[32-33]。此外，在 GSO 算法中步长的大小还与搜索空间的维数有关。对于维数较高的搜索空间，需要较大的步长；对于维数较低的搜索空间，需要较小的步长。因此，本文提出了根据一种依据搜索维数和迭代次数非线性自适应调整步长的策略，如式(6)、式(7)所示。

$$n=(2)^{-c} \tag{6}$$

$$s(t)=(s_{\max}-s_{\min})\left(\frac{t}{t_{\max}}\right)^n+s_{\min} \tag{7}$$

其中，c 是搜索空间的维数，$s_{\max}$ 和 $s_{\min}$ 分别是步长的最大值和最小值，$t_{\max}$ 为最大迭代次数。

3.3　改进的移动方式

萤火虫优化算法是一种新的智能群体优化算法，它通过萤火虫的移动来获得最优解。在基本的 GSO 算法中，萤火虫 i 向萤火虫 j 移动，j 为萤火虫 i 的邻域集内荧光素的值比 i 大的萤火虫。因此，在萤火虫的移动过程中萤火虫 i 只利用邻域集内的局部信息和选择一个邻居并向其移动的策略来获取最优值[32]。

为了提高 GSO 算法的收敛性能，快速和准确地收敛于全局最优解，本文在萤火虫的移动阶段引入了全局信息，即在每一次的迭代过程中，让所有萤火虫都向全局最优值移动。

提出了如下的移动方式：

$$x1_i(t+1)=(gbest[i]-x_i(t+1))+\alpha.*(rand-0.5) \tag{8}$$

其中，α 是一个常数，$rand$ 是[0,1]范围内的随机函数，$gbest$ 为整个萤火虫群中适应度最大的萤火虫。

3.4　算法描述

改进的萤火虫优化算法的基本步骤描述如下。

Step 1　萤火虫的初始化。初始化参数 $n,m,\rho,\gamma,\beta,n_t,s,\ell_0$，并对萤火虫的位置进行初始化，产生 n 只萤火虫。

Step 2　荧光素的更新阶段。根据式(1)更新荧光素的值。

Step 3　萤火虫的移动阶段。萤火虫根据式(2)选择更新概率，根据式(6)和式(7)更新步长，根据式(3)更新萤火虫的位置，再利用式(8)使所有萤火虫向最优萤火虫移动，来更新每个萤火虫个体的位置。

Step 4　决策域的更新阶段。根据式(4)更新萤火虫的局部决策域。

Step 5　判断是否满足终止条件，如果满足则循环结束；否则跳转到 Step 2。

4　实验结果分析

在 GSO 算法中，参数对算法性能有着非常重要的影响，本文的参数设置如表 1 所列。

表 1　参数设置

$t_{\max}$	n	γ	β	n_t	s	$s_{\max}$	$s_{\min}$	ρ	ℓ_0	α
100	50	0.6	0.08	5	0.03	1	0.0001	0.4	5	0.01

本文实验在 MATLAB R2014a 上编写和运行。为了测

试 IGSO 优化算法在多阈值彩色图像分割中的性能，本文选择了 5 幅标准测试图像，图像大小为 512×512，图像和其对应的直方图如图 1 所示。为了保证实验结果的有效性，针对每一幅图和每一种算法独立运行 30 次，IGSO，GSO，CQPSO 和 MBF 算法求出的最佳阈值和目标函数值如表 2 和表 3 所列，PSNR 和 SSIM 值如表 4 所列。当阈值个数为 4，5，6，7 时，基于改进的萤火虫优化算法的彩色图像分割结果如图 2 所示。

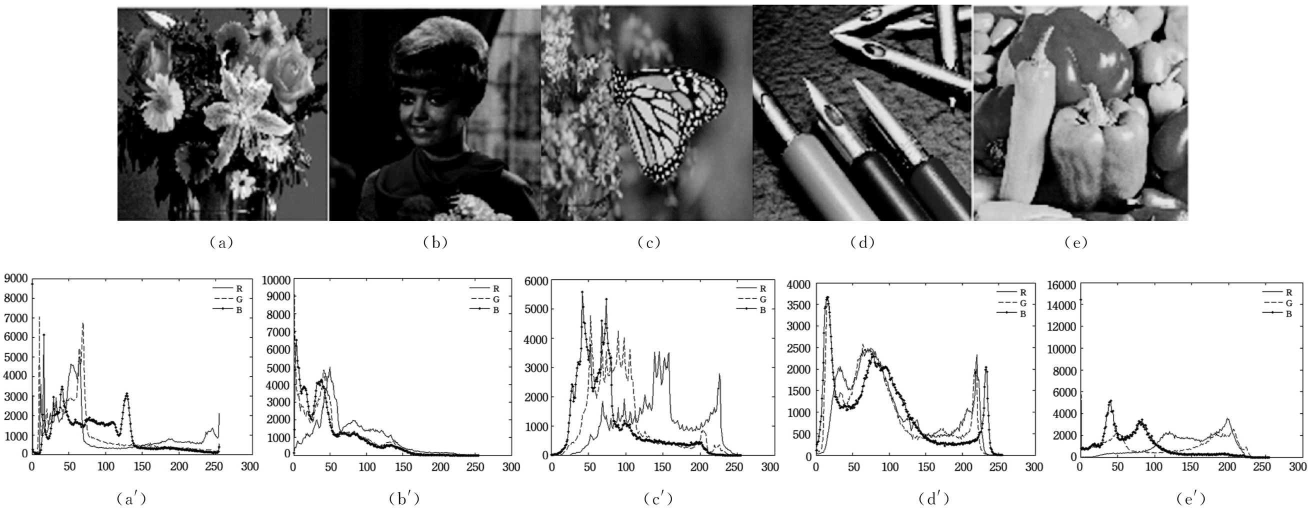

图 1 测试图像和对应的直方图

表 2 基于 Kapur 熵的 IGSO 算法和 GSO 算法的最佳阈值和最佳目标函数值

m	IGSO				GSO			
	R	G	B	f	R	G	B	f
Flower								
4	43,75,138,196	45,87,144,200	53,94,138,198	57.2543	43,75,138,196	45,87,144,200	53,94,138,198	57.2543
5	70,106,145, 181,218	74,108,145, 181,218	57,99,138, 180,218	65.8660	69,106,145, 182,218	74,110,148, 183,218	60,100,139, 181,217	65.8054
6	40,70,106,145, 181,219	43,75,110,146, 182,217	53,94,137,166, 195,224	73.9005	41,72,111,147, 178,215	42,77,113,148, 179,217	57,100,138,165, 195,227	73.8162
7	41,69,102,132, 165,195,227	42,74,103,134, 164,192,223	46,76,106,137, 167,197,226	81.4543	38,69,102,133, 163,192,224	42,75,99,126, 158,191,221	45,73,107,139, 168,199,226	81.3403
Girl								
4	38,90,143,188	52,98,142,181	48,89,129,171	55.8707	38,90,143,188	52,98,142,181	48,89,129,171	55.8707
5	59,101,143, 176,211	52,101,142, 171,203	45,83,115, 147,172	63.7012	62,104,144, 178,213	52,102,143, 172,202	43,86,116, 145,173	63.6447
6	37,72,109,146, 183,223	29,59,99,138, 169,198	45,75,104,135, 164,188	71.6783	36,71,110,147, 184,223	30,64,103,139, 170,203	46,76,106,134, 162,187	71.5969
7	28,60,90,119, 148,179,211	28,55,89,123, 155,185,213	26,53,82,111, 140,168,193	79.0768	28,59,86,115, 144,173,208	26,54,93,124, 157,186,214	28,55,85,113, 141,168,194	78.9621
Monarch								
4	63,100,137,183	48,113,159,208	80,114,147,180	54.4587	63,100,137,183	48,113,159,208	80,114,147,180	54.4587
5	63,108,143, 175,206	39,77,114, 156,208	60,101,138, 165,190	62.3107	63,108,144, 177,207	38,74,114, 154,208	59,101,138, 167,191	62.2485
6	62,96,131,160, 196,230	40,76,113,144, 176,209	50,81,109,134, 160,188	70.2822	58,95,127,160, 194,230	38,74,111,147, 177,210	45,80,105,137, 165,185	70.20400
7	62,95,128,159, 182,206,230	40,77,113,138, 164,188,209	22,52,82,109, 135,161,189	77.4658	64,99,129,158, 181,203,230	41,74,112,135, 158,186,211	22,52,82,110,136, 164,192	77.3562
Pen								
4	58,102,144,188	54,97,135,179	55,104,147,196	56.1530	58,102,144,188	54,97,135,179	55,104,147,196	56.1530
5	55,99,138,169, 200	55,99,136,170, 199	58,100,138, 175,210	64.0398	57,98,140,172,198	56,100,137,171, 199	58,103,141,178, 210	63.9829
6	45,77,109,141, 169,200	38,69,100,130, 162,195	34,69,104,139, 174,210	71.9302	41,72,102,132, 163,196	37,66,99,127, 162,189	36,70,106,142, 177,206	71.8453
7	38,64,90,116, 145,172,200	34,62,90,118, 145,174,201	30,62,91,122, 151,182,213	79.1864	41,65,93,117, 148,173,203	35,63,93,123, 145,174,202	34,65,94,124, 154,185,216	79.0614
Pepper								
4	56,96,137,177	61,103,142,182	54,100,136,179	55.4475	56,96,137,177	61,103,142,182	54,100,136,179	55.4475
5	59,96,130, 161,191	61,101,137, 167,196	53,99,131, 163,195	63.1466	57,96,133, 165,191	61,101,139, 171,199	55,100,133, 166,199	63.0894
6	41,71,100,130, 158,188	39,65,99,130, 162,194	44,73,102,131, 162,195	70.7648	43,75,101,134, 163,193	40,66,98,125, 158,195	43,72,100,127, 156,190	70.7006
7	40,69,98,125, 151,178,204	38,64,94,122, 152,181,211	44,70,96,120, 148,175,200	77.8289	45,74,101,129, 155,180,204	37,64,98,124, 152,179,210	42,67,92,120, 147,172,199	77.7156

表 3　基于 Kapur 熵的 CQPSO 算法和 MBF 算法的最佳阈值和最佳目标函数值

m	CQPSO				MBF			
	R	G	B	f	R	G	B	f
Flower								
4	43,75,138,196	45,87,144,200	53,94,138,198	57.2543	43,75,138,196	45,87,144,200	53,94,138,198	57.2543
5	72,106,145,182,218	74,109,146,183,218	58,99,137,180,217	65.8551	72,106,146,182,218	74,109,146,183,219	58,99,137,181,217	65.8472
6	41,70,107,144,182,218	44,76,111,147,182,218	54,95,139,167,195,224	73.8747	41,70,106,144,180,217	42,76,114,148,182,217	55,95,139,166,196,227	73.8625
7	43,70,103,134,166,195,228	43,76,104,136,165,192,223	47,77,108,138,168,197,226	81.4215	41,70,102,132,166,196,229	41,75,104,136,166,193,222	47,78,107,138,167,197,226	81.4037
Girl								
4	38,90,143,188	52,98,142,181	48,89,129,171	55.8707	38,90,143,188	52,98,142,181	48,89,129,171	55.8707
5	59,102,144,176,212	53,102,142,172,203	47,84,117,149,172	63.6897	60,103,144,176,213	52,102,143,172,202	46,85,116,151,172	63.6813
6	39,73,109,148,183,223	29,60,100,139,171,198	46,76,104,137,164,189	71.6567	36,71,109,147,183,223	30,61,99,140,170,199	46,76,105,135,166,189	71.6471
7	30,61,92,120,149,179,212	29,57,90,125,156,185,214	27,54,84,111,140,169,193	79.0471	27,60,88,119,149,180,211	27,56,90,124,155,185,214	28,54,83,111,140,168,194	79.0285
Monarch								
4	63,100,137,183	48,113,159,208	80,114,147,180	54.4587	63,100,137,183	48,113,159,208	80,114,147,180	54.4587
5	64,109,145,175,206	39,79,115,156,209	61,103,139,165,190	62.3002	63,110,146,175,207	39,78,117,157,208	59,101,138,163,189	62.2923
6	64,97,131,160,198,230	41,77,114,146,176,209	52,82,110,135,161,188	70.2610	63,97,132,161,197,231	39,77,114,145,177,209	48,80,110,134,161,189	70.2502
7	63,97,128,161,182,207,230	42,78,115,139,165,188,209	21,53,84,109,135,161,190	77.4349	61,96,127,159,181,207,230	41,76,112,138,165,188,210	23,50,82,109,136,161,190	77.415666
Pen								
4	58,102,144,188	54,97,135,179	55,104,147,196	56.1530	58,102,144,188	54,97,135,179	55,104,147,196	56.1530
5	56,100,139,169,200	55,99,136,170,199	59,102,138,175,211	64.0281	57,99,140,170,200	57,100,139,175,199	58,103,141,178,210	64.0192
6	47,78,110,142,169,200	39,70,102,130,164,195	33,71,105,139,175,210	71.9181	44,79,109,142,169,201	37,71,100,131,162,196	36,71,103,130,165,211	71.9064
7	39,66,91,117,146,172,200	35,63,91,119,147,174,201	32,62,93,122,153,182,214	79.1570	40,65,90,119,146,173,200	36,64,91,119,145,175,202	29,63,93,123,152,183,213	79.1327
Pepper								
4	56,96,137,177	61,103,142,182	54,100,136,179	55.4475	56,96,137,177	61,103,142,182	54,100,136,179	55.4475
5	60,96,130,163,192	62,101,137,168,197	54,100,131,165,195	63.1350	58,97,132,161,191	61,102,139,167,196	55,100,131,165,195	63.1271
6	42,73,101,131,158,188	39,67,99,132,163,194	45,74,103,132,163,195	70.7648	43,71,101,131,160,188	40,66,99,129,163,194	45,75,103,133,163,195	70.7532
7	40,67,99,127,152,178,204	39,66,95,123,154,181,211	44,71,97,121,148,176,202	77.7957	38,70,99,126,152,178,204	37,65,96,125,152,179,210	43,72,97,122,147,176,201	77.7763

表 4　基于 Kapur 熵的 IGSO、GSO、CQPSO 和 MBF 算法的 PSNR 和 SSIM 值

Image	m	IGSO		GSO		CQPSO		MBF	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Flower	4	20.3021	0.9158	20.3021	0.9158	20.3021	0.9158	20.3021	0.9158
	5	20.3037	0.9213	20.3034	0.9196	20.3036	0.9208	20.3036	0.9205
	6	20.3051	0.9336	20.3047	0.9273	20.3050	0.9325	20.3049	0.9319
	7	20.3065	0.9429	20.3058	0.9346	20.3063	0.9414	20.3062	0.9401
Girl	4	24.0571	0.8320	24.0571	0.8320	24.0571	0.8320	24.0571	0.8320
	5	24.0584	0.8813	24.0582	0.8797	24.0583	0.8809	24.0583	0.8805
	6	24.0599	0.9234	24.0595	0.9185	24.0598	0.9224	24.0597	0.9217
	7	24.0612	0.9344	24.0606	0.9260	24.0610	0.9328	24.0609	0.9217
Monarch	4	19.4731	0.7779	19.4731	0.7779	19.4731	0.7779	19.4731	0.7779
	5	19.4746	0.8901	19.4744	0.8889	19.4745	0.8896	19.4745	0.8893
	6	19.4794	0.9148	19.4789	0.9087	19.4793	0.9137	19.4792	0.9032
	7	19.4805	0.9293	19.4800	0.9214	19.4804	0.9286	19.4803	0.9275
Pen	4	19.8871	0.8986	19.8871	0.8986	19.8871	0.8986	19.8871	0.8986
	5	19.8880	0.9253	19.8877	0.9238	19.8879	0.9248	19.8878	0.9244
	6	19.8887	0.9527	19.8879	0.9468	19.8886	0.9516	19.8885	0.9510
	7	19.8894	0.9617	19.8888	0.9541	19.8892	0.9601	19.8891	0.9589
Pepper	4	18.1749	0.8603	18.1749	0.8603	18.1749	0.8603	18.1749	0.8603
	5	18.1764	0.8837	18.1762	0.8812	18.1760	0.8832	18.1761	0.8828
	6	18.1787	0.9204	18.1781	0.9145	18.1786	0.9192	18.1780	0.9185
	7	18.7195	0.9296	18.7188	0.9209	18.7194	0.9282	18.7193	0.9271

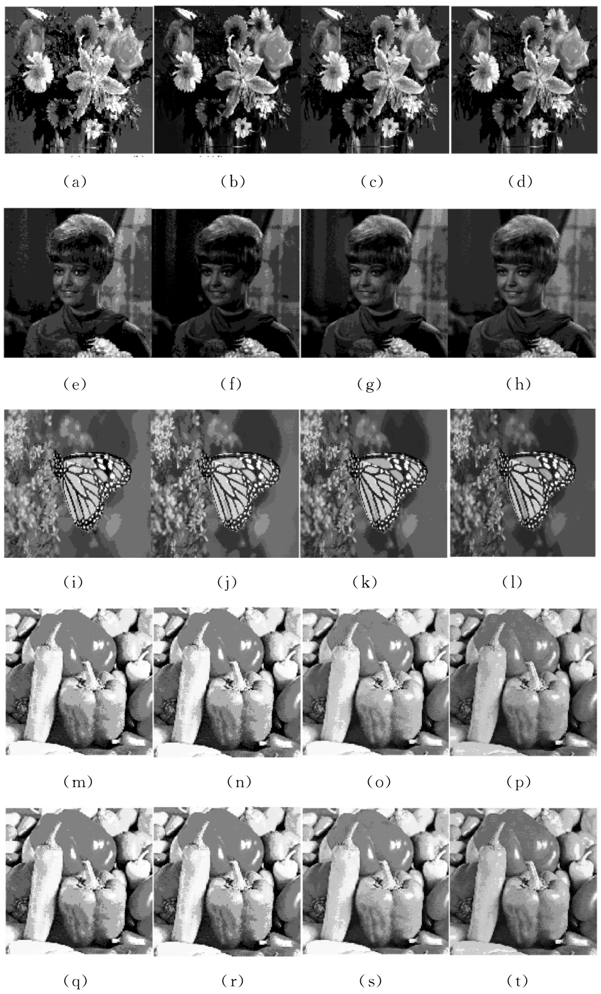

图 2 基于 Kapur 熵的改进萤火虫优化算法的彩色图像多阈值分割结果

由表 2 和表 3 可知，当阈值个数为 4 时，改进的 GSO 获得的最佳目标函数值、PSNR 和 SSIM 值与基本的 GSO，CQPSO 和 MBF 算法一致；当阈值个数大于 4 时，改进的 GSO 得到的最佳目标函数值、PSNR 和 SSIM 值均优于 GSO，CQPSO 和 MBF 算法。图 2 中从左至右分别是阈值个数为 4，5，6，7 的分割结果，可以看出利用本文提出的 GSO 算法对彩色图像进行多阈值分割，具有较好的分割效果。因此，针对多阈值彩色图像分割问题，基于 Kapur 熵的 IGSO 算法具有较好的分割性能。

结束语 GSO 算法是一种新型的智能群体优化算法，其在图像的多阈值分割方面的研究工作开展得还较少。本文通过对萤火虫的位置更新公式进行理论分析，得出步长和萤火虫的移动方向是影响萤火虫优化算法收敛性能的重要参数，步长又与迭代次数和搜索空间的维数有关，提出了根据迭代次数和搜索空间维数自适应更新步长的策略。同时，为了进一步提高算法的收敛性能和全局优化能力，在萤火虫移动阶段的位置更新公式中，引入了全局信息。最后以 Kapur 熵为目标函数，将改进的 GSO 算法用于彩色图像的多阈值分割中，实验结果表明，IGSO 算法相对于 GSO 算法而言，具有较好的分割性能，因此基于 Kapur 熵的 IGSO 算法是进行多阈值彩色图像分割的有效方法。下一步的工作是将 IGSO 算法用于其他复杂的优化问题中。

参 考 文 献

[1] 章毓晋. 图像分割[M]. 北京：科学出版社，2001：1-2.

[2] GAO H，XU W，SUN J，et al. Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm[J]. IEEE Transactions on Instrumentation and Measurement，2010，59(4)：934-946.

[3] LIU Y，MU C，KOU W，et al. Modified particle swarm optimization-based multilevel thresholding for image segmentation[J]. Soft Computing，2015，5(19)：1311-1327.

[4] GHAMISI P，COUCEIRO M S，MARTINS F M. Atli Benediktsson J，Multilevel image segmentation based on fractional-order Darwinian particle swarm optimization[J]. IEEE Transactions on Geoscience and Remote Sensing，2014，52(5)：2382-2394.

[5] HAMMOUCHE K，DIAF M，SIARRY P. A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation[J]. Computer Vision and Image Understanding，2008，109(2)：163-175.

[6] MAULIK U. Medical image segmentation using genetic algorithms[J]. IEEE Transactions on Information Technology in Biomedicine，2009，13(2)：166-173.

[7] CUEVAS E，ZALDIVAR D，PÉREZ-CISNEROS M. A novel multi-threshold segmentation approach based on differential evolution optimization[J]. Expert Systems with Applications，2010，37(7)：5265-5271.

[8] 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.Springer Berlin Heidelberg：Springer Press，2011：51-58.

[9] SARKAR S，DAS S，CHAUDHURI S S. A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution[J]. Pattern Recognition Letters，2015，54 (2015)：27-35.

[10] HORNG M H. Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation[J]. Expert Systems with Applications，2011，38(11)：13785-13791.

[11] MA M，LIANG J，GUO M，et al. SAR image segmentation based on Artificial Bee Colony algorithm[J]. Applied Soft Computing，2011，11(8)：5205-5214.

[12] CUEVAS E，SENCIÓN F，ZALDIVAR D，et al. A multi-threshold segmentation approach based on Artificial Bee Colony optimization[J]. Applied Intelligence，2012，37(3)：321-336.

[13] AGRAWAL S，PANDA R，BHUYAN S，et al. Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm[J]. Swarm and Evolutionary Computation，2013，11：16-30.

[14] BRAJEVIC I，TUBA M. Cuckoo search and firefly algorithm applied to multilevel image thresholding [M]// Cuckoo Search and Firefly Algorithm. Berlin Heidelberg：Springer Press，2014：115-139.

[15] BHANDARI A K，SINGH V K，KUMAR A，et al. Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur'

sentropy[J]. Expert Systems with Applications,2014,41(7):3538-3560.

[16] RAJINIKANTH V,COUCEIRO M S. RGB Histogram Based Color Image Segmentation Using Firefly Algorithm[J]. Procedia Computer Science,2015,46(2015):1449-1457.

[17] SANDIP D,SIDDHARTHA B,UJJWAL M. New quantum inspired meta-heuristic techniques for multi-level colour image thresholding[J]. Applied Soft Computing,2016,46:677-702.

[18] PARE S,KUMAR A,BAJAJ V,et al. A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve[J]. Applied Soft Computing,2016,47:76-102.

[19] BHANDARI A K,KUMAR A,CHAUDARY S,et al. A novel color image multilevel thresholding based segmentation using nature inspired optimization algorithms[J]. Expert Systems with Applications,2016,63(C):112-133.

[20] KUMAR S,PANT M,KUMAR M,et al. Colour image segmentation with histogram and homogeneity histogram difference using evolutionary algorithms[J]. Int. J. Mach. Learn. & Cyber,2015:1-21.

[21] KRISHNANAND K N,GHOSE D. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics[C]// IEEE Swarm Intelligence Symposium. USA,2005:84-91.

[22] KRISHNANAND K N,GHOSE D. Theoretical foundations for multiple rendezvous of glowworm-inspired mobile agents with variable local-decision domains[C]// American Control Conference. 2006:14-16.

[23] KRISHNANAND K N,GHOSE D. A glowworm swarm optimization based multi-robot system for signal source localization [J]. Studies in Computational Intelligence,2009,177(177):49-68.

[24] KRISHNANAND KN,GHOSE D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions[J]. Swarm Intelligence,2009,3(2):87-124.

[25] KRISHNANAND K N,GHOSE D. Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications[J]. Multiagent and Grid Systems,2006,2(3):209-222.

[26] KRISHNANAND K N. Glowworm swarm optimization:a multimodal function optimization paradigm with applications to multiple signal source localization tasks[D]. Indian:Department of Aerospace Engineering,Indian Institute of Science,2007.

[27] LIAO W H,KAO Y C,LI Y S. A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks[J]. Expert Systems with Applications,2011,38(10):12180-12188..

[28] HORNG M H. Multilevel Image Thresholding with Glowworm Swam Optimization Algorithm based on the Minimum Cross Entropy[J]. Advances in Information Sciences and Service Sciences,2013,5(10):1290-1298.

[29] LUO Q,OUYANG Z,CHEN X,et al. A multilevel threshold image segmentation algorithm based on glowworm swarm optimization[J]. J. Comput. Inf. Syst,2014,10(4):1621-1628.

[30] GAO H,XU W,SUN J,et al. Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm[J]. IEEE Transactions on Instrumentation and Measurement,2010,59(4):934-946.

[31] SATHYA P D,KAYALVIZHI R. Modified bacterial foraging algorithm based multilevel thresholding for image segmentation [J]. Engineering Applications of Artificial Intelligence,2011,24(4):595-615.

[32] KRISHNANAND K N,GHOSE D. Glowworm swarm optimisation:A new method for optimising multi-modal functions[J]. The International Journal of Computational Intelligence Studies,2009,1(1):93-119.

[33] 陈贵敏,贾建援,韩琪. 粒子群优化算法的惯性权值递减策略研究[J]. 西安交通大学学报,2006,40(1):53-56.

(上接第 197 页)

[11] SEO S H,OSTROMOUKHOV V. Pointillist video stylization based on particle tracing[J]. Multimedia Tools and Applications,2014,71(1):279-292.

[12] GUO Y,YU J,XU X,et al. Example based painting generation [J]. Journal of Zhejiang University-Science A,2006,7(7):1152-1159.

[13] 陈佳舟. 增强现实和图像风格化中的图像结构分析算法研究[D]. 杭州:浙江大学,2012.

[14] BOUSSEAU A,KAPLAN M,THOLLOT J,et al. Interactive watercolor rendering with temporal coherence and abstraction [C]// Proceedings of the 4th International Symposium on Non-photorealistic animation and Rendering. ACM,2006:141-149.

[15] 赵旭. Kinect 深度图像修复技术研究[D]. 大连:大连理工大学,2013.

[16] 余涛. Kinect 应用开发实战:用最自然的方式与机器对话[M]. 北京:机械工业出版社,2012.

[17] 李应彬,冯杰,张华熊,等. 基于改进双边滤波的 Kinect 深度图像空洞修复算法研究[J]. 工业控制计算机,2013,26(11):105-106.

[18] CAO C,CHEN S,ZHANG W,et al. Automatic motion-guided video stylization and personalization[C]// Proceedings of the 19th ACM International Conference on Multimedia. ACM,2011:1041-1044.

[19] WANG B,WANG W,YANG H,et al. Efficient example-based painting and synthesis of 2d directional texture[J]. IEEE Transactions on Visualization and Computer Graphics,2004,10(3):266-277.

[20] KANG H,LEE S,CHUI C K. Flow-based image abstraction [J]. IEEE Transactions on Visualization and Computer Graphics,2009,15(1):62-76.

[21] EFROS A A,FREEMAN W T. Image quilting for texture synthesis and transfer[C]// Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. ACM,2001:341-346.

[22] LIU C. Beyond pixels:exploring new representations and applications for motion analysis [D]. Massachusetts Institute of Technology,2009.

[23] CRIMINISI A,PEREZ P,TOYAMA K. Region filling and object removal by example-based image inpainting [J]. IEEE Transactions on Image Processing,2004,13(9):1200-1212.