

# 一种新的图象分割自适应算法的研究

魏志成 周激流 吕航 陶理 刘智明

(四川大学电子信息学院, 成都 610064)

**摘要** 首先对自适应遗传算法的变异算子进行了改进,对单点变异算子与双点变异算子的结合能有效地改善局部收敛进行了验证,然后提出了一种新的用自适应遗传算法分割图象的方法,并与传统的 Otsu 方法、灰度差直方图法和基于熵的方法作了比较。实验表明,该文的算法可保留图象的大部分信息,对一些复杂图象的处理能得到很好的处理结果,同时本文算法在时间上还有很大的优势。

**关键词** 图象分割 自适应遗传算法 交叉算子 变异算子

中图分类号: TP394.1 文献标识码: A 文章编号: 1006-8961(2000)03-0216-05

## A Study on Image Segmentation by A New Adaptive Algorithm

WEI Zhi-cheng, ZHOU Ji-liu, Lü Hang, TAO Li, LIU Zhi-ming

(Electronic and Information College, Sichuan University, Chengdu 610064)

**Abstract** Firstly, an improvement was made on mutation of adaptive genetic algorithm (AGA). The combination of one-point mutation and two-point mutation could prevent AGA from converging to the local optimum. Secondly, a new method for image segmentation using the improved AGA was put forward. The method of Otsu, the method of histogram based on gray difference and the entropy-based methods were compared with this algorithm. The improved AGA was proved to give attention to the main information by experiments. Good results can be achieved from some images. And much less time was used by the algorithm.

**Keywords** Image segmentation, Adaptive genetic algorithm(AGA), Crossover, Mutation

## 0 引言

图象分割是计算机视觉中要解决的第一个重要且困难的任务,尽管它一直受到科研人员的重视,但是它的发展很慢,被认为是计算机视觉的一个瓶颈。迄今为止,还没有一种图象分割方法适用于所有的图象,也没有一类图象所有的方法都适用于它<sup>[1]</sup>。

图象分割的方法有成百上千种。从广义上分,有经典方法和模糊数学的方法,而经典方法又可分为基于灰度直方图的阈值分割和基于边缘检测的迭代法,以及基于语义、句法的方法。其中,基于灰度直方图的阈值分割法又可分为单阈值和多阈值的方法<sup>[1,2]</sup>。本文所讨论的都是基于灰度直方图的方法。

例如,灰度差直方图法<sup>[3]</sup>,它假设图象主体和背景的边界灰度变化剧烈,因此分割后的细节较清晰。1979年由 N. Otsu 提出的动态门限方法<sup>[4]</sup>,由于它利用图象的灰度直方图,以目标和背景之间的方差最大,动态地确定图象分割门限值,因此是传统方法中性能较好的一种。但是, Lee S U 等人的研究表明,当目标的面积大于整幅图象的 30% 时,包括 Otsu 方法在内的传统算法的分割性能接近最优值,但当目标的相对面积减小时,这些方法的性能则迅速下降<sup>[5]</sup>。Kapur 等人提出的熵的分割方法<sup>[6]</sup>,从信息论的角度选择阈值,在一定程度上克服了上述算法的缺点,该方法利用了信号系统中熵的概念和灰度直方图的全局性和客观性,并假设了两个概率分布,其中一个为目标区域的,一个是背景区域的,要求分割

的时候,使这两部分熵的和最大,但是很多图象或者由于背景复杂,或者由于在同一个背景上重叠了若干个目标,因而分割时容易丧失部分信息.为了使分割后的图象保留更多的原图信息,本文在文献[6]介绍的方法基础上提出了一种用自适应遗传算法分割图象的方法.这种方法不仅能很好的分割一些复杂图象,而且速度非常快.

## 1 自适应遗传算法及其改进

在标准遗传算法(SGA)<sup>[7-9]</sup>中,需要使用固定的交叉概率 $P_c$ 和变异概率 $P_m$ .虽然对它们取值的研究也取得了一定的成果<sup>[10]</sup>,但是SGA仍然很容易收敛到局部最优,于是科学工作者就研究出了自适应遗传算法(Adaptive Genetic Algorithm, AGA)<sup>[11]</sup>.AGA是根据解群中所有个体的最大适应度和平均适应度,以及交叉和变异个体的适应度来确定这次操作中的 $P_c$ 和 $P_m$ 的.其 $P_c$ 和 $P_m$ 的设计如下:

$$\begin{cases} P_c = (f_{\max} - f') / (f_{\max} - \bar{f}), & f' > \bar{f} \\ P_c = 1.0, & f' \leq \bar{f} \end{cases} \quad (1)$$

$$\begin{cases} P_m = 0.5(f_{\max} - f) / (f_{\max} - \bar{f}), & f > \bar{f} \\ P_m = 0.5, & f \leq \bar{f} \end{cases} \quad (2)$$

其中, $P_c$ 是交叉概率; $P_m$ 是变异概率. $f_{\max}$ 是当前解群中个体的最大适应度, $\bar{f}$ 是解群的平均适应度, $f'$ 是要交叉的两个个体中较大的适应度, $f$ 是要变异的个体的适应度.

但是本文在利用这种AGA寻找二值化图象分割阈值时,发现结果往往会收敛到局部最优,分析其原因如下:一方面当解群中充斥了有限几个超级个体<sup>[9]</sup>,选择阈值时很难区分优劣,因此本文引入了窗口法<sup>[10,12]</sup>以增强指导性;另一方面,由于超级个体之间再交叉也不易产生适应度更高的新个体,而且一些实验的结果还表明,这几个超级个体中的任何一个变异后得到的新个体的适应度都很低.为此,本文提出了单点变异算子和双点变异算子结合的AGA,从而增加了遗传算法的随机性.这里需要注意的是,两点变异并不是单纯地增加了变异概率,因为有些问题即使是 $P_m = 1$ 也无法解决的,必须由两点变异来解决.

为了比较单点变异算子与双点变异算子结合的AGA和单点变异算子AGA的性能,本文用Otsu算法的类间方差<sup>[4]</sup>作为适应度函数,对大量图象作

了实验,由于篇幅有限,下面给出两幅图象用单点变异算子与双点变异算子结合的AGA和单点变异算子AGA的处理结果(表1和表2).

表1 性能比较之一

穷举法	编码	适应度	加窗(次)		无窗(次)	
			S&D	S	S&D	S
171	10101011	2071	74	42	75	36
167	10100111	2018	14	42	5	37
166	10100110	1971	1	无	无	1
246	11110101	1921	10	11	9	14
165	10100101	1918	无	无	无	无
159	10011111	1912	无	4	6	5

表2 性能比较之二

穷举法	编码	适应度	加窗(次)		无窗(次)	
			S&D	S	S&D	S
180	10110100	2074	34	17	32	17
198	11000110	2060	43	45	39	38
199	11000111	2059	无	无	无	无
200	11001000	2058	19	21	14	14
190	10111110	1980	4	11	1	7
189	10111101	1980	无	6	3	8

从表1、表2可见,在穷举法一列最上边的值是使用Otsu算法得到的全局最优灰度值;表中加窗(S&D)表示选择阈值时使用加窗法做适应度变换,且使用单点变异算子与双点变异算子相结合的AGA,无窗S表示选择阈值时不做适应度变换且使用单点变异算子的AGA.表中的次数是指运行程序100次,收敛于对应灰度的次数.另外,在表1的比较中,加窗S&D还收敛于175一次;无窗S&D还收敛于158、160、147各一次,收敛于162两次;加窗S还收敛于127一次,无窗S还收敛于160、168、172各一次,以及收敛于146、162各两次.在表2的比较中,无窗S&D还收敛于184一次,收敛于181、191、201各两次,收敛于197四次;无窗S还收敛于181两次,收敛于192、179、191各两次,收敛于197五次.

这些数据也验证了采用加窗法做适应度变换,有利于防止AGA收敛到局部最优<sup>[10]</sup>.由我们处理过的大量数据可以得到如下结论:

(1) 单点变异算子和双点变异算子结合的

AGA 的性能总是优于或等于单点变异 AGA<sup>[11]</sup> 的性能, 它比单点变异 AGA 能更有效地防止收敛于局部最优;

(2) 用加窗法做适应度变换有利于防止遗传算法收敛到局部最优。

## 2 门限自适应选择算法

本文给出的新的图象分割门限自适应选择算法的具体步骤如下:

(1) 统计图象的灰度直方图;

(2) 设计适应度函数 适应度函数是判别个体优劣的标准, 它的形式直接影响分割门限的选择。本文算法的适应度函数设计如下:

$$Fitness = H_1 + H_2 + H_3 \quad (3)$$

$$H_1 = \ln P_{s_1} + H_{s_1} / P_{s_1}$$

$$H_2 = \ln P_{s_2} + H_{s_2} / P_{s_2}$$

$$H_3 = \ln P_{s_3} + H_{s_3} / P_{s_3}$$

$$H_{s_1} = - \sum_{i=0}^{s_1} P_i \ln P_i$$

$$H_{s_2} = - \sum_{i=s_1+1}^{s_2} P_i \ln P_i$$

$$H_{s_3} = - \sum_{i=s_2+1}^{255} P_i \ln P_i$$

$$P_{s_1} = \sum_{i=0}^{s_1} P_i$$

$$P_{s_2} = \sum_{i=s_1+1}^{s_2} P_i$$

$$P_{s_3} = \sum_{i=s_2+1}^{255} P_i$$

$$P_i = N_i / N$$

其中  $N_i$  表示图象中灰度值是  $i$  的像素数,  $N$  为图象的像素总数,  $s_1$  和  $s_2$  是图象分割的双阈值, 它们要使式(3)的值最大;

(3) 产生初始群体 在 0 和 255 灰度值之间随机产生初始解群 *Gene1*, 按照式(3)计算每个个体的适应度。本文的解群规模为 20, 个体是 16 位二进制编码, 每个个体的物理意义是两个阈值  $s_1$  和  $s_2$ 。

(4) 选择运算 用窗口法<sup>[10, 12]</sup>的适应度变换思想对 *Gene1* 中的个体适应度进行变换, 再用两两竞争法<sup>[8]</sup>从中选取 20 个个体(有重串)进入配对池

(*Matching Pool*) 中, 记作 *MatPool*。其中窗口法就是让解群中每个串的适应度都减去  $q$  后作为其适应度( $q$  为当前解群中最小的适应度), 再用转轮法形成匹配集。用这种方法, 串的入选匹配集不仅与适应度有关(不成比例), 还与串的分散程度有关, 但比直接用转轮法效果要好。两两竞争法就是每次随机从解群中选两个串, 将适应度大的串加入匹配集中。这种方法比较充分地体现出了遗传算法的思想, 即既有随机性, 又有指导性, 是一个比较好的方法;

(5) 交叉运算 随机从 *MatPool* 中选两个个体, 随机产生一个均匀分布的伪随机数  $r$ 。如果  $r < P_c$ , 就交叉(本文采用一致交叉), 反之就直接进入下一代 *Gene2* 中。(1) 式是交叉概率的表达式, 交叉后的个体放入下一代 *Gene2* 中, 按照式(3)计算 *Gene2* 中的每个个体的适应度;

(6) 变异操作 接下来对 *Gene2* 中个体实现逐一变异。变异概率设计如式(2), 当产生的均匀分布的伪随机数  $r$  小于  $P_m$ , 则发生变异。变异算子采用前面阐述的单点变异算子和双点变异算子相结合的方法, 并按照式(3)重新计算 *Gene2* 中的每个个体的适应度;

(7) 把 *Gene1* 和 *Gene2* 中的所有个体用窗口法做适应度变换, 再用两两竞争法选择 20 个个体放入 *Gene1* 中, 执行下一轮循环, 直到收敛。由于到目前为止, 遗传算法数学上还没有比较严格的收敛判据, 因此本文采用启发式收敛, 当循环的次数达到 40 次时结束循环。

## 3 实验结果

图 2~ 图 5 给出 3 幅图象(图 1)的处理结果。

表 3 给出了在 PC 机上分别用熵的方法(穷举法)和自适应遗传算法对图象进行双阈值分割的时间比较。

表 3 自适应遗传算法和穷举法对图 1(a)(b)(c) 图象进行双阈值分割的时间比较

方法	(a)	(b)	(c)	平均用时 (s)
穷举法	6.2	7.3	7.1	6.9
自适应 GA	0.37	0.46	0.45	0.43

注: 表 3 数据均是在 Pentium 166, 32MB RAM 的兼容机上测得



图 1 待分割原图象

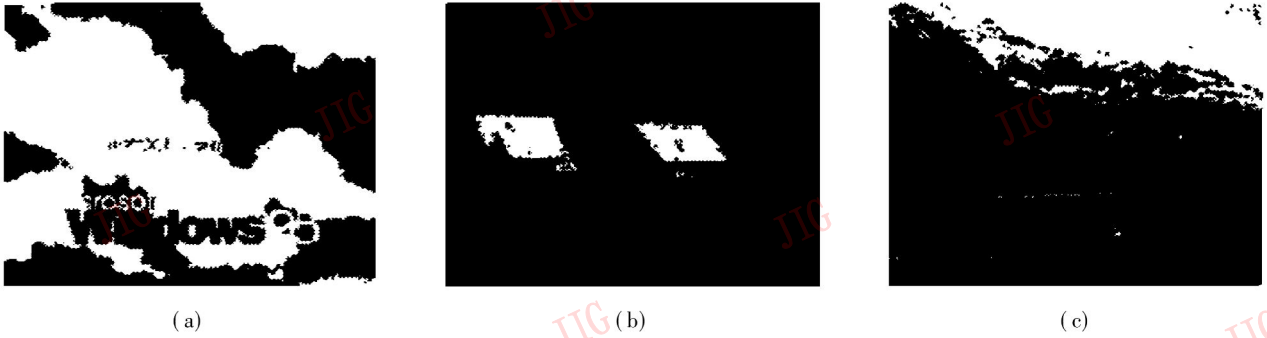


图 2 用 Otsu 法分割结果图象

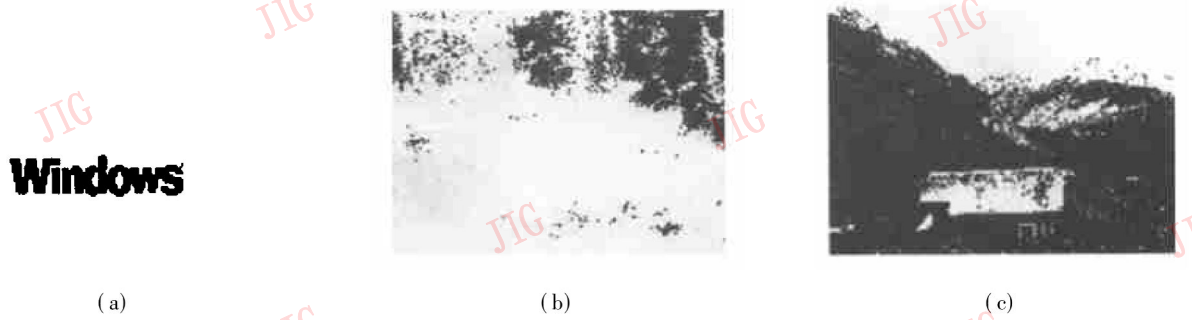


图 3 用灰度差直方图法分割结果图象

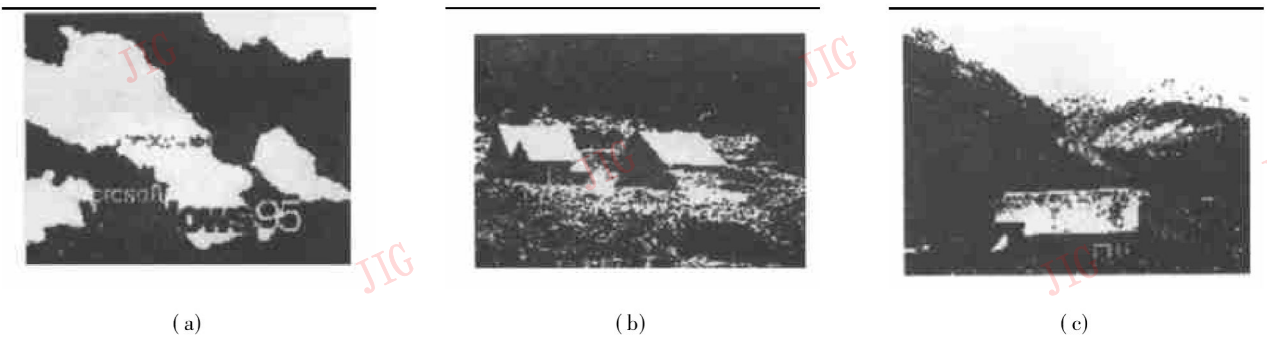


图 4 用熵的单阈值法分割结果图象

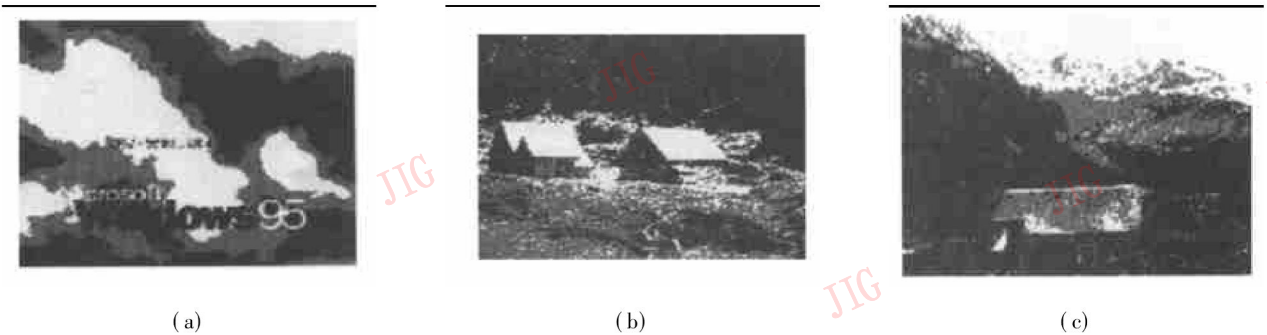


图 5 用自适应算法分割结果图象

试验结果表明, 本文的算法在分割复杂背景的图象或者有重叠目标的图象时, 较有优势, 能够得到比较满意的分割结果, 并且本文算法比文献[6]的穷举法有明显的速度优势. 另外, 从算法本身来看, 对自适应遗传算法来说, 当需要改变阈值个数或者添加其他参数(比如空间参数)的时候, 只需要改变个体的基因长度和适应度函数, 无需改变算法的流程, 因而大大增加了该算法的可扩展性.

## 4 结束语

本文利用单点变异和双点变异算子相结合的自适应遗传算法对一些图象进行双阈值分割, 收到了很好的效果. 但是, 我们仍然会遇到一些问题, 即不同的图象如果有相同的灰度直方图, 本文的算法是否对它们都同样适用? 对一幅图象, 两组不同的阈值都得到了最大的熵值, 应该选择那一组? 作者将就这些问题作进一步的研究, 同时也希望本文能够作为大家进一步研究的一个参考.

## 参考文献

- 1 Nikhil R Pal, Sankar K Pal. A review on image segmentation techniques. *Pattern Recognition*, 1993, 26(9): 1277~ 1294.
- 2 Sahoo P K, Soltani S, Wong A K C, Chen Y C. A survey of thresholding techniques. In: *Computer Graphics Image Processing*, 1988, 41: 233~ 260.
- 3 Watanade S *et al.* An automated apparatus for cancer prescreening: CYBEST. *CGIP*, 1974, 3: 350~ 358.
- 4 Otsu N. A threshold selection method form gray-level histograms. *IEEE Transactions on SMC*, 1979, 9(1): 62~ 66.
- 5 Haralick R M, Shapiro L G. Survey: image segmentation techniques. In: *Computer Graphics Image Processing*, 1985, 29: 100~ 132.
- 6 Kapur J N, Sahoo P K, Wong A K C. A new method for gray-

level picture thresholding using the entropy of the histogram. In: *Computer Graphics Image Processing*, 1985, 29: 273~ 285.

- 7 Goldberg D E. *Genetic Algorithms in Search, Optimization and Machine Learning*. MA: Addison-Wesley, 1989.
- 8 张晓绩, 戴冠中, 徐乃平. 一种新的优化搜索算法——遗传算法. *控制理论与应用*, 1995, 12(3).
- 9 韩祯祥, 文福拴. 模拟进化优化方法及其应用——遗传算法. *计算机科学*, 1995, 22(2): .
- 10 Grefenstette J J. Optimization of control parameters for genetic algorithms. *IEEE SMC*, 1986, 16(1).
- 11 Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 1994, 24(4).
- 12 Kreinovich V *et al.* Genetic algorithms: What fitness scaling is optimal? *Cybernetics and Systems*, 1993, 24(1).



魏志成 四川大学电子信息学院 97 级研究生。主要研究兴趣为遗传算法、图象处理和模式识别。



周激流 1963 年生, 教授, 博士。感兴趣的研究方向为: 图象图形处理, 模式识别与人工智能等。

吕航 四川大学电子信息学院 97 级研究生。

陶理, 刘智明 四川大学电子信息学院 98 级研究生。