

文章编号: 1001-0920(2007)02-0202-04

基于粒子群和二维 Otsu 方法的快速图像分割

唐英干, 刘 冬, 关新平

(燕山大学 电气工程学院, 河北 秦皇岛 066004)

摘 要: 二维 Otsu 方法同时考虑了图像的灰度信息和像素间的空间邻域信息, 是一种有效的图像分割方法. 针对二维 Otsu 方法计算量大的特点, 采用粒子群算法来搜索最优二维阈值向量. 每个粒子代表一个可行的二维阈值向量, 通过粒子群之间的协作来获得最优阈值. 结果表明, 所提出的方法不仅能得到理想的分割结果, 而且计算量大大减少, 达到了快速分割的目的, 便于二维 Otsu 方法的实时应用.

关键词: 图像分割; 二维 Otsu 方法; 粒子群

中图分类号: TP391.41

文献标识码: A

Fast image segmentation based on particle swarm optimization and two-dimension Otsu method

TANG Ying-gan, LIU Dong, GUAN Xin-ping

(Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China. Correspondent: TANG Ying-gan, E-mail: ygtang@ysu.edu.cn)

Abstract: 2-D Otsu method, which considers the gray information and spatial neighbor information between pixels in the image simultaneously, is an efficient image segmentation method. However, the computational burden of finding optimal threshold vector is very large for 2-D Otsu method. A optimization method, i. e., particle swarm optimization (PSO), is used to find the best 2-D threshold vector, in which each particle represents a possible 2-D threshold vector and the best 2-D threshold is obtained through the cooperation among particles. Experimental results show that the proposed method can not only obtain ideal segmentation results but also decrease the computation cost reasonably, and it is suitable for real time application.

Key words: Image segmentation; 2-D Otsu; Particle swarm

1 引 言

图像分割是指按照一定的相似性准则将图像分成几个有意义、互不重叠的区域, 在同一个区域内, 图像的特征(灰度、纹理、色彩)是相似的, 而在不同的区域内, 图像的特征则明显不同. 图像分割是图像处理中的基本问题, 也是图像理解的基础, 分割质量直接影响着后续处理结果.

阈值法是一种常用的分割方法, 它假定图像的直方图呈双峰分布, 根据某一准则在两峰之间(谷底)选择一个恰当的阈值, 即可将图像的背景和目标分离. 著名的阈值分割法有 Otsu 法^[1]、最大熵方法^[2]等. Otsu 阈值法是一种基于可分离性的方法, 它选择一个阈值使得类间方差最大. 由于 Otsu 方

法是基于二维直方图的分割方法, 它只考虑了图像的灰度信息而没有考虑像素的空间位置信息, 当图像直方图没有明显的双峰或者图像信噪比较低时, 其分割效果并不理想. 借助 Abutaleb 等^[3,4]的二维直方图的思想, 文献[5]提出了二维 Otsu 分割方法. 该方法通过邻域平均得到平滑的图像, 然后统计原始图像和平滑图像的二维直方图, 根据二维直方图选择最优二维阈值向量使得二维类间方差最大. 由于同时考虑了灰度信息和空间邻域信息, 二维 Otsu 方法的分割结果常常优于二维 Otsu 方法.

基于二维直方图的分割方法虽然分割性能较好, 但计算量非常巨大, 其主要原因是从二维直方图扩展到二维直方图后, 最佳阈值的搜索空间变大了.

收稿日期: 2005-11-02; 修回日期: 2005-12-15.

基金项目: 国家杰出青年基金项目(60525303); 国家自然科学基金项目(60404022); 河北省教育厅基金项目(2002209); 燕山大学博士基金项目(B243).

作者简介: 唐英干(1975—), 男, 江西新建人, 讲师, 博士生, 从事智能算法、图像处理等研究; 关新平(1963—), 男, 黑龙江齐齐哈尔人, 教授, 博士生导师, 从事鲁棒控制、网络控制等研究.

如果原始图像的灰度级为 L , 那么一维直方图最佳阈值的搜索空间为 $0 \sim L$, 而二维直方图的最佳阈值搜索空间为 $0 \sim L \times L$. 另外, 一维直方图情形下的一些特征量将变成二维的, 其计算复杂性也增加了. 为了降低计算量, Chen 等^[6]提出了一种快速分割方法, 该方法将原始图像的灰度进一步量化, 以减少图像的灰度等级, 然后再用二维阈值分割图像; 而 Gong 等^[7]从减少重复计算的观点出发, 提出一种递推快速算法. 这两种方法都取得了较好的分割性能和较少的计算代价.

粒子群算法是 Kennedy 等^[8]于 1995 年提出的一种群智能优化算法, 它模拟了鸟集群飞行觅食的行为, 通过鸟之间的集体协作使群体达到所期望的目的. 粒子群算法已成功地应用于优化问题, 解决了大量的工程实际问题, 如无功优化^[10]、多边形近似^[11]、组合优化^[12]. 实践表明, 粒子群算法是一种高效的优化算法. 针对二维 Otsu 方法计算量大的特点, 本文采用粒子群算法寻找最优的二维阈值向量, 实验结果表明, 粒子群算法不仅能寻找到全局最优阈值向量, 而且只需要花费很小的计算代价.

2 二维 Otsu 分割方法

设 $f(x, y)$ ($1 \leq x \leq M, 1 \leq y \leq N$) 是一幅大小为 $M \times N$ 的图像, 其灰度级为 L . 在每个像素点处, 计算 $n \times n$ 邻域的平均灰度值, 得到一幅平滑的图像 $g(x, y)$, 其灰度级也为 L . 设 r_{ij} 为图像中灰度级为 i 邻域平均灰度为 j 的像素点的个数, 那么

$$p_{ij} = \frac{r_{ij}}{M \times N},$$
$$i = 1, 2, \dots, L, j = 1, 2, \dots, L \quad (1)$$

为图像的二维直方图, 它是一个 $L \times L$ 的矩阵, 如图 1 所示.

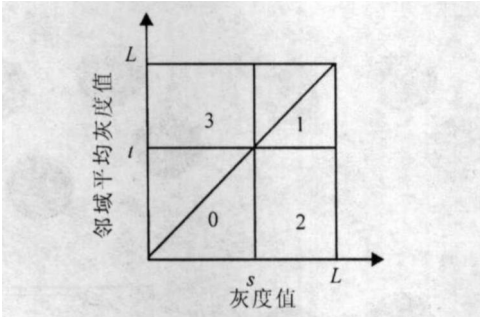


图 1 二维直方图

假设阈值向量 (s, t) 将二维直方图分成 4 个区域. 对于背景或目标内部的像素而言, 其灰度值与邻域平均灰度值是相似的, 而对于位于目标和背景边界处的像素, 其灰度值与邻域平均灰度值有很大的不同, 所以区域 0 和 1 代表目标或背景类, 区域 2 和 3 表示边界点或噪声点. 由于边界点和噪声点占少数,

可假设二维直方图中远离对角线的分量近似为 0, 即 $p_{ij} \approx 0 (i = s + 1, \dots, L; j = 1, \dots, t \text{ 和 } i = 1, \dots, s; j = t + 1, \dots, L)$.

设图像中存在背景 C_0 和目标 C_1 两类, 那么两类的概率分别为

$$\omega_0 = \Pr(C_0) = \sum_{i=1}^s \sum_{j=1}^t p_{ij} = \omega_0(s, t), \quad (2)$$

$$\omega_1 = \Pr(C_1) = \sum_{i=s+1}^L \sum_{j=t+1}^L p_{ij} = \omega_1(s, t). \quad (3)$$

两类对应的均值向量分别为

$$\mu_0 = (\mu_{0i}, \mu_{0j})^T =$$
$$\left[\sum_{i=1}^s i \Pr(i | C_0), \sum_{j=1}^t j \Pr(j | C_0) \right]^T =$$
$$\left[\sum_{i=1}^s \sum_{j=1}^t i p_{ij} / \omega_0, \sum_{i=1}^s \sum_{j=1}^t j p_{ij} / \omega_0 \right]^T, \quad (4)$$

$$\mu_1 = (\mu_{1i}, \mu_{1j})^T =$$
$$\left[\sum_{i=s+1}^L \sum_{j=t+1}^L i p_{ij} / \omega_1, \sum_{i=s+1}^L \sum_{j=t+1}^L j p_{ij} / \omega_1 \right]^T. \quad (5)$$

二维直方图总的均值向量为

$$\mu_T = (\mu_Ti, \mu_Tj)^T =$$
$$\left[\sum_{i=1}^L \sum_{j=1}^L i p_{ij}, \sum_{i=1}^L \sum_{j=1}^L j p_{ij} \right]^T. \quad (6)$$

由于假设远离对角线方向的分量近似为零, 有

$$\omega_0 + \omega_1 \approx 1, \mu_T \approx \omega_0 \mu_0 + \omega_1 \mu_1. \quad (7)$$

定义类间方差矩阵为

$$\sigma_B(s, t) = \sum_{k=0}^1 \Pr(C_k) [(\mu_k - \mu_T)(\mu_k - \mu_T)^T], \quad (8)$$

以矩阵 $\sigma_B(s, t)$ 的迹作为类间离散度的测度, 有

$$\text{tr}[\sigma_B(s, t)] =$$
$$([\mu_{Ti} \omega_0(s, t) - \mu_i(s, t)]^2 + [\mu_{Tj} \omega_0(s, t) - \mu_j(s, t)]^2) / (\omega_0(s, t) [1 - \omega_0(s, t)]). \quad (9)$$

其中

$$\mu_i(s, t) = \sum_{i=1}^s \sum_{j=1}^t i p_{ij}, \quad (10)$$

$$\mu_j(s, t) = \sum_{i=1}^s \sum_{j=1}^t j p_{ij}. \quad (11)$$

选择最佳的阈值向量 (s^*, t^*) , 使得

$$(s^*, t^*) = \arg \max_{1 \leq s, t \leq L} \text{tr}[\sigma_B(s, t)]. \quad (12)$$

在 $L \times L$ 空间内直接搜索使得式 (12) 最大的阈值向量是很耗费时间的, 该算法难以实时应用. 本文采用粒子群优化算法优化式 (12), 通过粒子群之间的相互协作来获得最佳阈值向量.

3 粒子群算法

受鸟集群飞行觅食行为的启发, Kennedy 等于 1995 年提出了一种新的进化算法, 即粒子群算法

(PSO). 与遗传算法类似, PSO 也是一种基于群体和适应度的全局优化方法. 在 PSO 算法中, 群体中的每个个体称为“粒子”, 在 D 维解空间中, 每个粒子 i 有一个位置 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 和“飞行”速度 $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. 粒子的位置表示问题的解, 其对应的目标函数值作为评价该粒子优劣程度的适应度; 速度表示粒子在解空间中的当前位置移动到下一个位置的速度大小.

PSO 算法首先初始化一群具有随机位置和速度的粒子, 然后通过迭代的方式在解空间中寻找最优解. 在迭代过程中, 每个粒子动态跟踪两个极值来更新其位置和速度. 假设在第 t 次迭代时刻, 粒子 i 目前所找到的最优解为 $pBest_i$, 称为个体极值; 整个粒子群目前所找到的最优解为 $gBest$, 称为全局极值. 在 $t+1$ 次迭代时刻, 粒子 i 根据上述两个极值, 由式(13)更新自己的速度, 有

$$v_i(t+1) = wv_i(t) + c_1 r_1 (pBest_i - x_i(t)) + c_2 r_2 (gBest - x_i(t)). \quad (13)$$

然后以速度 $v_i(t+1)$ 从当前位置移动到下一个位置, 即

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (14)$$

其中: t 表示迭代次数; w 是惯性因子; c_1, c_2 是学习因子, 通常 $c_1 = c_2 = 2$; r_1, r_2 是均匀分布在 $(0, 1)$ 区间上的随机数. 粒子在解空间中不断地跟踪个体极值和全局极值进行搜索, 直到达到规定的迭代次数或满足规定的误差标准为止.

为了防止粒子远离搜索空间, 粒子的速度限制在 $[-v_{max}, v_{max}]$ 范围内. Shi^[9] 研究表明, 惯性因子对算法的性能有很大的影响, 他提出了一种采用线性递减的方法来动态调整惯性因子 w , 有

$$w = w_{max} - \text{iter} \times \frac{w_{max} - w_{min}}{\text{iter}_{max}}. \quad (15)$$

其中: w_{max} 和 w_{min} 分别表示最大惯性因子和最小惯性因子, iter 表示当前迭代次数, iter_{max} 表示总的迭代次数. 最大惯性因子和最小惯性因子一般在 $(0, 1)$ 区间按照启发式方法选择, 本文采用文献[9]中建议的值, 即 $w_{max} = 0.95, w_{min} = 0.4$. 与遗传算法相比, 粒子群算法保留了基于种群的全局搜索策略, 但它所采用的位置-速度模型操作简单, 避免了复杂的遗传操作, 使得它具有简单、运行速度快的特点.

为了减少二维 Otsu 分割方法的计算量, 本文采用粒子群算法来搜索最佳的阈值向量, 其基本过程如下:

Step1: 随机生成 m 个二维的粒子, 粒子的位置从 $(0, 255)$ 之间随机产生, 速度在 $[-v_{max}, v_{max}]$ 之间

随机产生, 设置最大的迭代次数 N , 最大惯性因子 w_{max} 和最小惯性因子 w_{min} , 令 $t = 0$.

Step2: 根据式(9) 计算每个粒子的适应度, 更新每个粒子的个体极值 $pBest_i (i = 1, 2, \dots, m)$ 和整个粒子群的全局极值 $gBest$.

Step3: 根据式(15) 计算新的惯性因子, 按照式(13) 和(14) 更新每个粒子的位置和速度.

Step4: 令 $t = t + 1$, 返回 Step2 直至 $t = N$.

Step5: 输出粒子群的最佳位置, 即最优阈值向量 (s^*, t^*) .

设分割后的图像为 $f_{s^*, t^*}(x, y)$, 定义为

$$f_{s^*, t^*}(x, y) = \begin{cases} 0, & \text{if } f(x, y) < s^* \text{ and } g(x, y) < t^*; \\ 1, & \text{if } f(x, y) \geq s^* \text{ or } g(x, y) \geq t^*. \end{cases}$$

4 实验结果与分析

为了验证本文方法的有效性, 采用本文方法对两幅真实的图像进行分割实验. 所有实验在 Pentium IV 2.66 G 上进行, 算法采用 Matlab 6.5 语言实现. 第 1 幅图像是叠加了零均值方差为 1 的高斯噪声的硬币图像, 大小为 302×242 . 第 2 幅图像是红外图像, 大小为 255×185 . 对每一幅图像分别采用穷举法、文献[7] 中的快速递推方法和 PSO 算法来寻找最佳的二维阈值向量, 并将二维 Otsu 分割结果与一维 Otsu 分割结果进行比较.

PSO 算法的主要参数为: 每个粒子的维数是 2, 粒子群的个数为 20, 最大迭代次数为 25, 惯性因子采用线性递减方式, $w_{max} = 0.95, w_{min} = 0.4$, 粒子每一维的最大速度 $v_{max} = 4$. 由于粒子群采用随机初始化的方式, 为了客观地比较算法的性能, 对每一幅图像运行 10 次 PSO 算法. 分割结果如图 2 和图 3 所示.

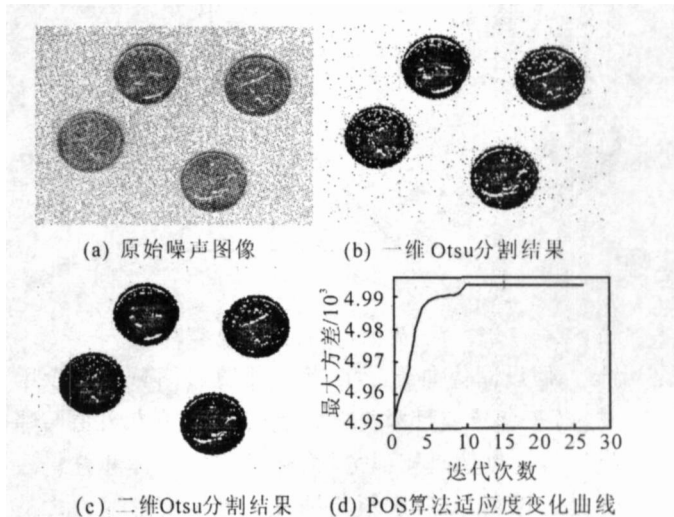


图2 硬币图像分割结果

由图 2 和图 3 可见, 二维 Otsu 方法的分割效果

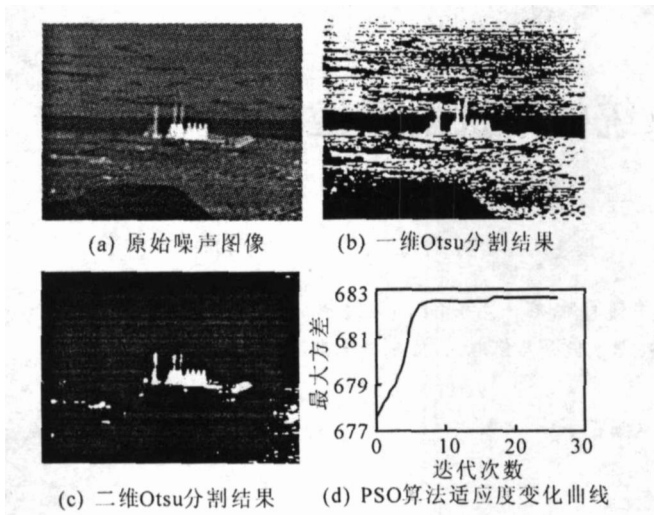


图 3 红外图像分割结果

要比一维 Otsu 方法的结果好得多,后者有很多的错分点,而前者的错分点则很少,这主要是因为二维 Otsu 方法同时考虑了图像的灰度信息和像素间的空间位置关系.对于二维 Otsu 方法,采用穷举法、文献[7]中的快速递推方法和 PSO 算法所得到的最优阈值向量、计算时间(s) 和最大目标函数(式(12))值见表 1.

表 1 穷举法、快速递推法、PSO 算法性能比较

方法	硬币图像			红外图像		
	最佳阈值向量	计算时间/s	最优目标值	最佳阈值向量	计算时间/s	最优目标值
穷举法	(185, 173)	115.86	4 993.4	(126, 123)	72.34	682.68
快速递推法	(185, 173)	1.64	4 993.4	(126, 123)	2.0	682.68
PSO 算法	(185, 173)	1.25	4 993.4	(126, 123)	1.04	682.68

实验中运行 10 次 PSO 算法,10 次 PSO 算法的平均适应度值的变化曲线如图 2(d) 和图 3(d) 所示.每次 PSO 算法只要经过 10 几次迭代就能搜索到全局最优值和全局最优的阈值向量,其计算时间要远远少于穷举法的计算时间,与文献[7]的快速递推算法的计算时间相比也要少.实验结果表明,PSO 算法是一种高效的优化方法,用它来计算二维 Otsu 方法的最佳阈值向量不仅能搜索到全局最佳阈值向量而且计算代价很小,为二维 Otsu 方法的实时应用提供一个新的途径.

5 结 语

二维 Otsu 方法是一种有效的图像分割方法,它不仅考虑了图像的灰度信息而且考虑了像素间的空间邻域信息,在图像的信噪比较低时,该方法能够得到比一维 Otsu 方法更好的分割结果.由于该方法需要根据二维类间方差最大的准则计算二维最优

阈值向量,与一维 Otsu 方法相比,其计算量大大增加.为此,本文提出采用 PSO 算法来计算最优二维阈值向量.结果表明,PSO 算法不仅能有效地搜索到全局最优二维阈值向量而且计算量大大减少,为二维 Otsu 方法的实时应用提供了一个新的途径.

参考文献(References)

[1] Otsu N. A threshold selection method from gray-level histogram [J]. IEEE Trans on Systems, Man and Cybernetic,1979, 9(1): 62-66.

[2] Kapur J N,Sahoo P K,Wong A K C. 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.

[3] Abutaleb A S. Automatic thresholding of gray-level pictures using two-dimensional entropy [J]. Computer Vision, Graphics and Image Processing, 1989, 47(1): 22-32.

[4] Brink A D. Thresholding of digital images using two-dimensional entropies [J]. Pattern Recognition, 1992, 25(8): 803-808.

[5] 刘健庄,栗文青. 灰度图象的二维 Otsu 自动阈值分割方法[J]. 自动化学报, 1993, 19(1): 101-105. (Liu J Z, Li W Q. The automatic threshold of gray-level pictures via two-dimensional otsu method [J]. Acta Automatica Sinica, 1993, 19(1): 101-105.)

[6] Chen W T, Wen C H, Yang C W. A fast two-dimensional entropy threshold algorithm [J]. Pattern Recognition, 1994, 27(7): 885-893.

[7] Gong J, Li L Y, Chen W N. Fast recursive algorithm for two-dimensional thresholding [J]. Pattern Recognition,1998,31(3): 295-300.

[8] Kennedy J, Eberthart R C. Particle Swarm Optimization [C]. Proc IEEE Int Conf on Neural Networks. Piscataway, 1995: 1942-1948.

[9] Shi Y H, Eberthart R C. A modified particle swarm optimizer [C]. Proc IEEE Int Conf on Evolutionary Computation. Anchorage, 1998: 69-73.

[10] Jiang C W, Bompard W. A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimization [J]. Mathematics and Computers in Simulation, 2005, 68(1): 57-65.

[11] Yin P Y. A discrete particle swarm algorithm for optimal polgonal approximation of digital curves [J]. J of Visual Communication and Image Representation, 2004, 15(2): 241-260.

[12] Salman A, Ahmad I, Madani S A. Particle swarm optimization for task assignment roblem [J]. Microprocessors and Microsystems, 2002, 26(8): 363-371.