中国图象图形学报
Journal of Image and Graphics

基于快速递推和搜索策略的优化 2 维熵分割算法

谢剑斌[1)]　刘　通[1)]　王金岩[2)]　何亦征[2)]

[1)]（国防科技大学电子科学与工程学院，长沙　410073）

[2)]（中国航空无线电电子研究所，上海　200233）

摘　要　首先阐述了优化 2 维熵函数的概念；然后提出了一种无失真的快速递推算法和有失真的优化搜索策略，这样进一步提高了优化 2 维熵阈值算法的运算速度，同时能得到与传统算法相近的分割效果；最后从理论和实验两个方面对本文算法进行了分析和验证。

关键词　2 维熵　快速递推　搜索策略

中图法分类号：TP391.4　　**文献标识码**：A　　**文章编号**：1006-8961(2008)04-0662-04

An Optimal Two-dimensional Entropic Segmentation Algorithm Based on Fast Recursion and Search Strategy

XIE Jian-bin[1)], LIU Tong[1)], WANG Jin-yan[2)], HE Yi-zheng[2)]

[1)] (*College of Electronic Science and Engineering, National University of Defense Technology, Changsha* 410073)

[2)] (*China Aeronautical Radio Electronics Research Institute, Shanghai* 200233)

Abstract　This paper presents the concept of optimal two-dimensional entropy function at first. Then we presents an undistorted fast recursion algorithm and a distorted optimal search strategy. In this way, the proposed algorithm can decrease computational time when it gets the similar image segmentation as traditional algorithm does. Finally we analyze and validate the algorithm theoretically and experimentally.

Keywords　two-dimensional entropies, fast recursion, search strategy

1　引　　言

图像分割是图像分析、识别和理解的基础，是机器视觉与图像处理领域的一个重要内容。图像分割方法可以归纳为三大类：基于阈值的分割方法，基于边缘的分割方法和基于区域的分割方法。其中，基于阈值的分割方法是最常用的方法。

最大熵法对不同信噪比和不同大小的目标均能产生较好的图像分割效果，是效率较高的阈值分割方法之一。常用的最大熵法有 1 维熵和 2 维熵两种，前者主要依据图像的 1 维灰度直方图选取阈值，其优点是处理速度快，缺点是对于 1 维灰度直方图，没有明显峰和谷的图像分割效果不理想，而且受噪声、光照等影响大；后者依据图像单像素点的灰度值分布及其邻域的平均灰度值分布所构成的 2 维灰度直方图选取阈值，其优点是算法适应性强，分割效果更好，缺点是处理速度太慢，如 Brink 所述，此算法是一个四重循环，计算复杂度为 $O(l^4)$[1]。为了满足实时处理的需求，Chen 通过由粗到细两步运算将计算复杂度降至 $O(l^{8/3})$[2]，龚坚又将其降至 $O(l^2)$[3]，杨姝优化了 2 维熵函数，将运算速度提高 15% 以上[4]。本文基于杨姝提出的优化 2 维熵函数，提出了一种无失真的快速递推算法和有失真的优化搜索策略，在保证分割效果的前提下进一步提高了优化 2 维熵阈值算法的运算速度。

基金项目："十五"国防重点预研项目(102010201)

收稿日期：2006-07-08；**改回日期**：2006-12-05

第一作者简介：谢剑斌(1971～　)，男。副教授，博士后、室主任、硕士生导师。研究方向为数字视频技术、信息可视化技术和图形图像处理。E-mail：jbxie@126.com

2　实现算法

定义 k_{ij} 表示同时使点灰度为 i、邻域灰度均值为 j 的像素点对数。假设阈值在 (s,t) 处，且目标灰度值较低，背景灰度值较高，则点灰度-邻域灰度均值矩阵 k_{ij} 被划分为 A、B、C、D 4 个区域，如图 1 所示。沿对角线分布的 A 区和 B 区分别代表目标和背景，远离对角线的 C 区和 D 区分别代表边界和噪声。

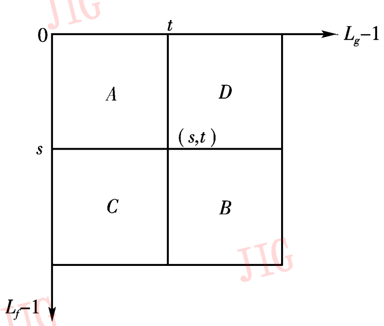


图 1　点灰度-邻域灰度均值平面

Fig. 1　Flat of pixel gray-neighborhood average gray

图 1 中，L_f 和 L_g 分别代表点灰度和邻域灰度均值的灰度级。

设 A 区和 B 区的像素和分别记为 P_A、P_B，它们的概率分布分别记为 p_{ij}^A、p_{ij}^B，则

$$P_A = \sum_{i=0}^{s-1}\sum_{j=0}^{t-1} k_{ij} \tag{1}$$

$$P_B = \sum_{i=s}^{L_f-1}\sum_{j=t}^{L_g-1} k_{ij} \tag{2}$$

$$p_{ij}^A = k_{ij}/P_A$$
$$i = 0,\cdots,s-1;j = 0,\cdots,t-1 \tag{3}$$

$$p_{ij}^B = k_{ij}/P_B$$
$$i = s,\cdots,L_f-1;j = t,\cdots,L_g-1 \tag{4}$$

杨姝根据熵函数在等概率场下取到最大值的性质，提出了一种优化的 2 维熵函数[4]：

$$H(s,t) = \sum_{i=0}^{s-1}\sum_{j=0}^{t-1}\left|p_{ij}^A - \frac{1}{st}\right| + \sum_{i=s}^{L_f-1}\sum_{j=t}^{L_g-1}\left|p_{ij}^B - \frac{1}{(L_f-1)(L_g-1)}\right| \tag{5}$$

最佳阈值 (s^*,t^*) 满足：

$$(s^*,t^*) = \mathrm{Arg}\{\min_{0\leqslant s\leqslant L_f-1,0\leqslant t\leqslant L_g-1} H(s,t)\} \tag{6}$$

2.1　快速递推算法

寻求最佳阈值 (s^*,t^*) 需要遍历 (s,t) 的定义域，求出使 $H(s,t)$ 为最小时的 (s,t)。在这一过程中，P_A、P_B 的求解耗时很多。事实上，P_A、P_B 是 (s,t) 的函数，且

$$P_A(s,t+1) = \sum_{i=0}^{s-1}\sum_{j=0}^{t} k_{ij} = \sum_{i=0}^{s-1}\sum_{j=0}^{t-1} k_{ij} + \sum_{i=0}^{s-1} k_{it} = P_A(s,t) + P_A(s) \tag{7}$$

$$P_B(s,t+1) = \sum_{i=s}^{L_f-1}\sum_{j=t+1}^{L_g-1} k_{ij} = \sum_{i=s}^{L_f-1}\sum_{j=t}^{L_g-1} k_{ij} - \sum_{i=s}^{L_f-1} k_{it} = P_B(s,t) - P_B(s) \tag{8}$$

可见，P_A、P_B 可以由式(7)、(8)递推计算。

算法设计时，先求出起始点 (s_0,t_0) 的 P_A、P_B 值 $P_A(s_0,t_0)$、$P_B(s_0,t_0)$，然后由式(7)、式(8)计算本行其他点的 P_A、P_B 值，同时记录各列初始增量 $P_A(s)$、$P_B(s)$。

计算下一行的 P_A、P_B 值时，先依据式(9)、式(10)递推计算该行初始 P_A、P_B 值，同时更新行初始值。

$$P_A(s+1,t) = \sum_{i=0}^{s}\sum_{j=0}^{t-1} k_{ij} = \sum_{i=0}^{s-1}\sum_{j=0}^{t-1} k_{ij} + \sum_{j=0}^{t-1} k_{sj} = P_A(s,t) + \sum_{j=0}^{t-1} k_{sj} \tag{9}$$

$$P_B(s+1,t) = \sum_{i=s+1}^{L_f-1}\sum_{j=t}^{L_g-1} k_{ij} = \sum_{i=s}^{L_f-1}\sum_{j=t}^{L_g-1} k_{ij} - \sum_{j=t}^{L_g-1} k_{sj} = P_B(s,t) - \sum_{j=t}^{L_g-1} k_{sj} \tag{10}$$

本行其他 P_A、P_B 值由式(7)、式(8)递推计算，其中各列增量由式(11)、式(12)递推计算，同时更新各列增量。

$$P_A(s+1) = \sum_{i=0}^{s} k_{it} = \sum_{i=0}^{s-1} k_{it} + k_{st} = P_A(s) + k_{st} \tag{11}$$

$$P_B(s+1) = \sum_{i=s+1}^{L_f-1} k_{it} = \sum_{i=s}^{L_f-1} k_{it} - k_{st} = P_B(s) - k_{st} \tag{12}$$

在对角线上，

$$P_A(s+1,t+1) = \sum_{i=0}^{s}\sum_{j=0}^{t} k_{ij} = \sum_{i=0}^{s-1}\sum_{j=0}^{t-1} k_{ij} + \sum_{j=0}^{t-1} k_{sj} + \sum_{i=0}^{s} k_{it} = P_A(s,t) + P_A'(t) + P_A'(s) \tag{13}$$

$$P_A'(t+1) = \sum_{j=0}^{t} k_{sj} = \sum_{i=0}^{t-1} k_{it} + k_{st} = P_A'(t) + k_{st} \tag{14}$$

$$P_A'(s+1) = \sum_{i=0}^{s+1} k_{it} = \sum_{i=0}^{s} k_{it} + k_{(s+1)t}$$

$$= P_A'(s) + k_{(s+1)t} \tag{15}$$

$$P_B(s+1,t+1) = \sum_{i=s+1}^{L_f-1}\sum_{j=t+1}^{L_g-1} k_{ij}$$
$$= \sum_{i=s}^{L_f-1}\sum_{j=t}^{L_g-1} k_{ij} - \sum_{j=t}^{L_g-1} k_{sj} - \sum_{i=s+1}^{L_f-1} k_{it}$$
$$= P_B(s,t) - P_B'(t) - P_B'(s) \tag{16}$$

$$P_B'(t+1) = \sum_{j=t+1}^{L_g-1} k_{sj} = \sum_{j=t}^{L_g-1} k_{sj} - k_{st}$$
$$= P_B'(t) - k_{st} \tag{17}$$

$$P_B'(s+1) = \sum_{i=s+2}^{L_f-1} k_{it} = \sum_{i=s+1}^{L_f-1} k_{it} - k_{(s+1)t}$$
$$= P_B'(s) - k_{(s+1)t} \tag{18}$$

由快速递推算法求解 P_A、P_B 可以进一步提高运算速度，而且不影响分割效果。

2.2　优化搜索策略

2 维熵阈值分割算法在搜索最佳阈值时需要利用穷举法遍历 (s,t) 的定义域。事实上，如图 1 所示，图像中目标区域和背景区域都集中在对角线附近，远离对角线的是边缘和噪声。因此，最佳阈值通常分布在图 2 所示对角线上某点周围一个小区域中，为此，本文提出一种优化的搜索策略，减小了搜索区域，进一步提高了运算速度，而且分割效果好。

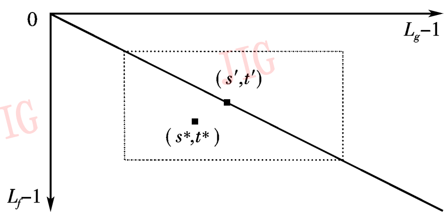


图 2　优化搜索策略

Fig. 2　Optimal search strategy

优化搜索策略在搜索最佳阈值时分两步进行：第 1 步是在点灰度-邻域灰度均值平面的对角线上进行粗略搜索，寻找最佳阈值所在的区域；第 2 步是在第 1 次搜索确定的区域内进行精确搜索，寻找最佳阈值。这一步的算法改进相对于传统算法是有失真的。

本文算法的基本步骤如下：

(1) 读入图像 $f(x,y)$；

(2) 求图像 $f(x,y)$ 的 $k\times k$ 邻域灰度均值图像 $g(x,y)$；

(3) 求同时使 $f(x,y)=i$，$g(x,y)=j$ 的点灰度-邻域灰度均值点对 $k_{ij}(i=0,\cdots,L_f-1;\ j=0,\cdots,L_g-1)$；

(4) 粗略搜索，在对角线上搜索使 $H(s,t)$ 取最小值的阈值 (s',t')；

(5) 精确搜索，在 (s',t') 周围 $m\times n$ 邻域内搜索使 $H(s,t)$ 取最小值的阈值 (s^*,t^*)，此即为最佳阈值；

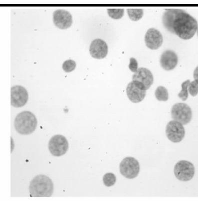
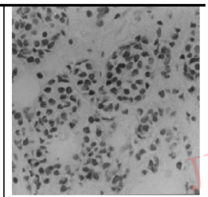
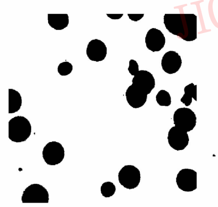
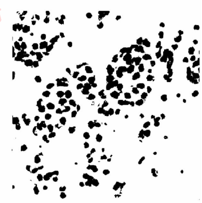
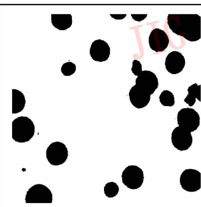
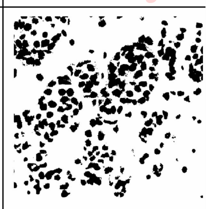
(6) 按最佳阈值分割图像，结束程序。

3　实验结果分析

表 1 给出了本文算法与杨姝算法的图像分割和耗时对比。其中，实验图像尺度为 512×512，点灰度-邻域灰度均值尺度从左到右分别为 17×17、13×13，搜索邻域尺度从左到右分别为 64×64、32×32。

表 1　图像分割与耗时

Tab. 1　Image segmentation and computational time

原始算法			
杨姝算法	分割后图像		
	阈值	(219,227)	(118,126)
	耗时	7.102s	6.860s
本文算法	分割后图像		
	阈值	(219,227)	(123,135)
	耗时	1.573s	0.585s

从计算量上分析，杨姝算法中计算 $H(s,t)$ 的次数是 $L_f\times L_g$ 次，而采用优化搜索策略后，计算次数要小于 $L_f+L_g+m\times n$ 次；另外，杨姝算法中求 P_A、P_B 的计算复杂度为 $O(l^2)$，而采用快速递推算法后 P_A、P_B 的计算复杂度降为 $O(l)$。本文算法的运算速度受点灰度-邻域灰度均值尺度和搜索邻域尺度的影响很大，一般尺度越小，运算速度越快，相对于杨姝算法，实验中本文算法运算速度提高 78% 以上，而在存储空间上，本文算法仅增加约 40 个字节

的存储空间。

从图像分割效果看,两种算法差异不大,而且搜索邻域尺度越大,二者的分割效果越接近。本文算法对于背景和目标内部变化平缓的图像分割效果好,而对于背景和目标内部变化尖锐的图像分割效果一般,所以分割前分析图像特性是必要的,这是以后将深入研究的课题。

参考文献(References)

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

2 Chen W T, Wen C H, Yang C H. A fast two-dimensional entropic thresholding algorithm [J]. Pattern Recognition, 1994, **27** (7): 885 ~ 893.

3 Gong Jian, Li li-yuan, Chen Wei-nan. A fast two dimensional entropic thresholding method [J]. Journal of Southeast University, 1996, **25**(4): 33 ~ 38. [龚坚, 李立源, 陈维南. 二维熵阈值分割的快速算法[J]. 东南大学学报, 1996, **25**(4): 33 ~ 38.]

4 Yang Shu, Gao Li-qun, Bian Li-ying. Improvement of 2-D maximum entropy threshold algorithm based on optimal entropy function[J]. Journal of System Simulation, 2005, **17**(6): 1350 ~ 1352. [杨姝, 高立群, 边丽英. 基于优化熵函数二维最大熵阈值算法改进[J]. 系统仿真学报, 2005, **17**(6): 1350 ~ 1352.]

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