

Otsu 准则下分割阈值的快速计算

何志勇, 孙立宁, 陈立国

(苏州大学机电工程学院, 江苏苏州 215021)

摘 要: 传统 Otsu 法在确定阈值时需要穷举计算图像中每个灰度值为阈值时的类间方差. 文中利用 Otsu 阈值的性质, 提出了一个新算法以快速计算 Otsu 阈值. 新算法搜寻出与两类类内均值的平均值的整数部分相等的阈值, 从中确定一个符合 Otsu 准则的阈值. 传统 Otsu 法在对梯度图像中的小目标分割时分割性能不佳, 文中提出了一个 Otsu 阈值的改进算法, 该算法使用快速计算 Otsu 阈值的新算法递归求解分割阈值. 实验结果表明, 与传统 Otsu 算法相比, 计算 Otsu 阈值的快速算法速度更快, 而阈值的改进算法对梯度图像中的小目标分割效果更好.

关键词: 图像分割; Otsu 准则; 阈值选取; 快速算法

中图分类号: TP391.41 **文献标识码:** A **文章编号:** 0372-2112 (2013) 02-0267-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2013.02.010

Fast Computation of Threshold Based on Otsu Criterion

HE Zhi-yong, SUN Li-ning, CHEN Li-guo

(School of Mechanical and Electrical Engineering, Soochow University, Suzhou, Jiangsu 215021, China)

Abstract: The traditional Otsu algorithm has to exhaustively compute all between-class variances. Based on one characteristic of Otsu threshold, this paperwork proposes a new fast algorithm. The new algorithm finds out every threshold which is equal to the integer part of the average of the mean levels of two classes, and then selects one threshold which is in accord with Otsu criterion. The traditional Otsu algorithm cannot work well when it extracts small object from gradient image, so an improved thresholding algorithm is proposed. Based on the fast Otsu algorithm provided, the improved thresholding algorithm recursively computes threshold. Experimental results show that the fast Otsu algorithm is faster than the traditional Otsu algorithm. Experimental results also show that the improved thresholding algorithm is effective to segment small object of gradient image.

Key words: image segmentation; Otsu criterion; thresholding; fast computation

1 引言

图像分割算法中, 阈值法因为简单、快速而应用十分广泛^[1]. 传统 Otsu 法^[2]是一种最早由 Otsu 于 1979 年提出的经典阈值法, 其很有效且应用广泛^[1,3]. 该方法根据图像的一维灰度直方图, 穷举搜索使像素分为目标和背景两类, 且类间方差最大的一维阈值. 二维 Otsu 法^[4]和三维 Otsu 法^[5]比传统 Otsu 法有更佳的抗噪性能. 虽然一些二维 Otsu 快速算法^[6,7]和三维 Otsu 快速算法^[8]被提出来, 但上述算法的计算复杂度远大于一维 Otsu 法.

传统 Otsu 法对类间方差的穷举计算过程其实也是低效的, 只是多数情况下由于图像的灰度级范围有限而

使总计算量并不大. 目前针对传统 Otsu 法改进其计算效率的算法都不能保证阈值与 Otsu 准则一致, 并不是严格意义的传统 Otsu 法的快速算法. 文献[9]中的快速迭代算法所得阈值可能与 Otsu 阈值并不一致^[10], 从而导致分割结果不同; 文献[11]中算法的阈值与 Otsu 阈值也有偏差.

文中基于文献[12]已经证明了的一维 Otsu 阈值的性质, 提出了一个可以快速搜索出一个符合 Otsu 准则的阈值的算法, 该算法计算快速、准确.

传统 Otsu 法在一些场合下分割性能不佳^[12]. 文中基于所提出的 Otsu 阈值快速算法进一步提出了递归求解分割阈值的算法, 该算法所得阈值与传统 Otsu 阈值相比能更准确地从梯度图像中分割小目标.

2 Otsu 阈值的快速计算

2.1 Otsu 准则

设图像的灰度级分为 $0, 1, \dots, L-1$, 灰度为 i 的像素数目为 n_i , 总像素数为 N , 且 $N = n_0 + n_1 + \dots + n_{L-1}$, $p_i = n_i/N$ 表示灰度为 i 的像素的概率, 于是有 $\sum_{i=0}^{L-1} p_i = 1$, 而图像总均值为 $\mu_T = \sum_{i=0}^{L-1} ip_i$. 设 C_0 和 C_1 分别表示两类像素群, $C_0 = [0, \dots, k]$, $C_1 = [k+1, \dots, L-1]$. C_0 和 C_1 的均值分别记为 $\mu_0(k)$ 和 $\mu_1(k)$. 令 $\omega_0(k) = \sum_{i=0}^k p_i$, $\omega_1(k) = 1 - \omega_0(k)$, $\mu(k) = \sum_{i=0}^k ip_i$, 则:

$$\mu_0(k) = \frac{\sum_{i=0}^k ip_i}{\omega_0(k)} = \mu(k)/\omega_0(k) \quad (1)$$

$$\mu_1(k) = \frac{\sum_{i=k+1}^{L-1} ip_i}{\omega_1(k)} = \frac{\mu_T - \mu(k)}{1 - \omega_0(k)} \quad (2)$$

于是可得类间方差 σ_B^2 :

$$\sigma_B^2(k) = \omega_0(k)[\mu_0(k) - \mu_T]^2 + \omega_1(k)[\mu_1(k) - \mu_T]^2 \quad (3)$$

进一步可得:

$$\sigma_B^2(k) = \frac{[\mu_T \omega_0(k) - \mu(k)]^2}{\omega_0(k)[1 - \omega_0(k)]} \quad (4)$$

最佳阈值 k^* 的选取原则即为:

$$\sigma_B^2(k^*) = \max_{0 \leq k < L-1} \sigma_B^2(k) \quad (5)$$

2.2 Otsu 阈值的快速计算

2.2.1 原理

传统 Otsu 法计算阈值时, 对类间方差的穷举计算为最耗时部分, 因此, 求解 Otsu 阈值时若能减少对二阶矩的计算次数, 则可以提高计算效率.

文献[12]已经证明, 符合 Otsu 准则的阈值有如下性质: 有一个最佳阈值 k^{opt} 与阈值分割后两类的类内均值的平均值的整数部分相等, 即:

$$k^{opt} = \left\lfloor \frac{1}{2}(\mu_0(k^{opt}) + \mu_1(k^{opt})) \right\rfloor \quad (6)$$

若 k^* 的取值唯一, 则 k^{opt} 等于 k^* ; 若 k^* 取值为多个, k^{opt} 只为其中之一, 则不同最佳阈值之间无像素点.

k^{opt} 与 k^* 分割图像的结果相同, 求解 k^{opt} 即可. 令

$$f_1(k) = \left\lfloor \frac{1}{2}(\mu_0(k) + \mu_1(k)) \right\rfloor \quad (7)$$

$$f_2(k) = k \quad (8)$$

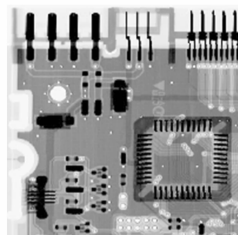
由式(6)可知, k^{opt} 应位于 $f_1(k)$ 曲线与 $f_2(k)$ 曲线的交点处. 例如图 1 中 Lena 图像和电路板图像, k^{opt} 都对应某个 $f_1(k)$ 与 $f_2(k)$ 交点处的灰度值.

若能快速找出所有 $f_1(k)$ 与 $f_2(k)$ 的交点, 并比较

以这些交点处灰度值作为阈值时的类间方差, 则使类间方差最大的阈值就是 Otsu 阈值.



(a) Lena



(c) 电路

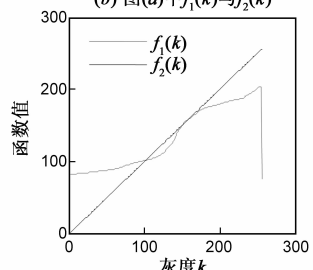
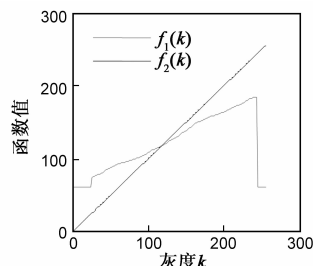


图1 图像的 $f_1(k)$ 与 $f_2(k)$

2.2.2 $f_1(k)$ 的性质及计算

性质 1 在满足 $\omega_0(k+1) < 1$ 的 k 的取值区间上, $f_1(k+1) \geq f_1(k)$.

证明:

$$\begin{aligned} \mu_0(k+1) &= \frac{\sum_{i=0}^k ip_i + (k+1)p_{k+1}}{\sum_{i=0}^k p_i + p_{k+1}} \\ &= \frac{\mu_0(k) \sum_{i=0}^k p_i + (k+1)p_{k+1}}{\sum_{i=0}^k p_i + p_{k+1}} \end{aligned}$$

由于 $\mu_0(k) < k+1$, 于是有:

$$\mu_0(k+1) \geq \frac{\mu_0(k) \sum_{i=0}^k p_i + \mu_0(k) p_{k+1}}{\sum_{i=0}^k p_i + p_{k+1}} = \mu_0(k) \quad (9)$$

式(9)中等号仅在 p_{k+1} 为 0 时成立.

$$\begin{aligned} \mu_1(k) &= \frac{(k+1)p_{k+1} + \sum_{i=k+2}^{L-1} ip_i}{p_{k+1} + \sum_{i=k+2}^{L-1} p_i} \\ &= \frac{(k+1)p_{k+1} + \mu_1(k+1) \sum_{i=k+2}^{L-1} p_i}{p_{k+1} + \sum_{i=k+2}^{L-1} p_i} \end{aligned}$$

由 $\mu_1(k+1)$ 为 $C_1 = [k+2, \dots, L-1]$ 的均值, 有 $k+1 < \mu_1(k+1)$, 于是:

$$\mu_1(k) \leq \frac{\mu_1(k+1)p_{k+1} + \mu_1(k+1) \sum_{i=k+2}^{L-1} p_i}{p_{k+1} + \sum_{i=k+2}^{L-1} p_i} = \mu_1(k+1) \quad (10)$$

式(10)中等号仅在 p_{k+1} 为 0 时成立.

由式(9)、(10)可得:

$$\begin{aligned} f_1(k+1) &= \left\lfloor \frac{1}{2}(\mu_0(k+1) + \mu_1(k+1)) \right\rfloor \\ &\geq \left\lfloor \frac{1}{2}(\mu_0(k) + \mu_1(k)) \right\rfloor \end{aligned}$$

亦即: $f_1(k+1) \geq f_1(k)$. 证毕.

性质 2 在 $k=0$ 处, 有 $f_1(0) \geq 0$; 对满足 $\omega_0(k) < 1$ 且 $\omega_0(k+1) = 1$ 的 k 值, 有 $f_1(k) \leq k$.

证明: 由 $f_1(k)$ 的定义可知, 必有 $f_1(0) \geq 0$. 由 $\mu_0(k)$ 概念可知 $\mu_0(k) \leq k$. 若 $\omega_0(k) < 1$ 且 $\omega_0(k+1) = 1$, 可得: $p_{k+1} > 0$, 灰度大于 $k+1$ 的概率为 0. 由 $\mu_1(k)$ 定义有:

$$\mu_1(k) = \frac{(k+1)p_{k+1} + \sum_{i=k+2}^{L-1} ip_i}{p_{k+1} + \sum_{i=k+2}^{L-1} p_i} = \frac{(k+1)p_{k+1}}{p_{k+1}} = k+1$$

于是:

$$\begin{aligned} f_1(k) &= \left\lfloor \frac{1}{2}(\mu_0(k) + (k+1)) \right\rfloor \leq \left\lfloor \frac{1}{2}(k + k+1) \right\rfloor \\ &= k. \text{ 证毕.} \end{aligned}$$

$f_1(k)$ 的计算可依 $\omega_0(k)$ 的取值按不同公式计算.

若 $0 < \omega_0(k) < 1$, 由式(1)、(2)、(7)可得:

$$f_1(k) = \left\lfloor \frac{[\mu(k) + \mu_T \omega_0(k) - 2\mu(k)\omega_0(k)]}{2\omega_0(k)[1 - \omega_0(k)]} \right\rfloor \quad (11)$$

若 $\omega_0(k) = 0$ 或 $\omega_0(k) = 1$, 由 $\mu_0(k)$ 、 $\mu_1(k)$ 、 μ_T 的概念及式(7)有:

$$f_1(k) = \lfloor \mu_T/2 \rfloor \quad (12)$$

2.2.3 快速计算算法

令 $d = |f_1(k) - f_2(k)|$, 则关于 $f_1(k)$ 曲线与 $f_2(k)$ 曲线的交点有下述命题成立.

命题 1 若 $f_1(k) > f_2(k)$, 且 $\omega_0(k+d) < 1$, 则在 $(k, k+d)$ 的区间上, $f_1(k)$ 曲线与 $f_2(k)$ 曲线没有交点.

证明: 对 $0 < i < d$, 皆有: $f_1(k) = f_2(k) + d = f_2(k+d) > f_2(k+i)$. 由 $\omega_0(k+d) < 1$ 及 $f_1(k)$ 性质 1 得: $f_1(k+d) \geq f_1(k) > f_2(k+i)$. 证毕.

命题 2 若 $f_1(k) < f_2(k)$, 且 $\omega_0(k) < 1$, 则在 $(k-d, k)$ 的区间上, $f_1(k)$ 曲线与 $f_2(k)$ 曲线没有交点.

证明: 对 $0 < i < d$, 皆有: $f_1(k) = f_2(k) - d = f_2(k-i)$

$-d < f_2(k-i)$. 由 $\omega_0(k) < 1$ 及 $f_1(k)$ 性质 1 得: $f_1(k-d) \leq f_1(k) < f_2(k-i)$. 证毕.

图像的最佳分割阈值 k^{opt} 应满足 $\omega_0(k^{opt}) < 1$, 因而在满足 $\omega_0(k) < 1$ 的 k 的取值区间上, 利用上述命题 1、命题 2 及 $f_1(k)$ 的性质, 快速求解 $f_1(k)$ 曲线与 $f_2(k)$ 曲线的交点, 进而实现 k^{opt} 的快速求解. 算法具体如下:

step 1 按定义计算 $\omega_0(k)$ 、 $\mu(k)$ 序列及 μ_T .

step 2 从 $k=0$ 起探测使 $f_1(k) = f_2(k)$ 成立的最小 k 值, 探测过程为: 当 $f_1(k) > f_2(k)$ 时, 令 $d = |f_1(k) - f_2(k)|$, 若 $\omega_0(k+d) < 1$ 则 k 值加 d , 否则 k 值加 1. 直至 $f_1(k) = f_2(k)$ 时, 令 $k_1 = k$, 结束探测.

step 3 从 $k=L-1$ 起探测符合 $\omega_0(k) < 1$, 且使 $f_1(k) = f_2(k)$ 成立的最大 k 值, 探测过程为: 当 $\omega_0(k) = 1$, k 值减 1. 当 $\omega_0(k) < 1$ 且 $f_1(k) < f_2(k)$ 时, 令 $d = |f_1(k) - f_2(k)|$, k 值减 d . 直至 $f_1(k) = f_2(k)$ 时, 令 $k_2 = k$, 结束探测.

step 4 若 $k_1 = k_2$, 则 k_1 值即为 k^{opt} , 结束求解; 否则, 继续 step 5 和 step 6.

step 5 在 (k_1, k_2) 区间上自右向左探测使 $f_1(k) = f_2(k)$ 成立的 k 值, 探测过程为: 当 $f_1(k) > f_2(k)$, k 值减 1. 当 $f_1(k) = f_2(k)$, 记录 k 值, k 值减 1. 当 $f_1(k) < f_2(k)$, 令 $d = |f_1(k) - f_2(k)|$, k 值减 d .

step 6 计算各交点处的 k 值对应的类间方差, 使类间方差最大的 k 值即为 k^{opt} .

算法中, step 3 需从 $k=L-1$ 起计算 $f_1(k)$, 进而需计算 $\omega_0(k)$ 、 $\mu(k)$, 因此对 $i \in [0, \dots, L-1]$ 的所有 p_i 和 ip_i 的累加过程是无法避免的, 故 step 1 与传统算法一样先对 $\omega_0(k)$ 、 $\mu(k)$ 序列及 μ_T 计算.

新算法探测计算 $f_1(k)$ 的次数与探测过程中 d 的取值序列及交点数目有关, 但小于 L . 新算法中, 当 $f_1(k)$ 与 $f_2(k)$ 只有一个交点时, 无需计算类间方差; 当有多个交点时, 类间方差的计算次数也仅为交点个数.

对大量图像中 $f_1(k)$ 与 $f_2(k)$ 的交点情况进行分析, 所分析图像包括 USC-SIPI 库中的部分图像、伯克利的 BSDS300 库中 100 张测试图像等. 发现只有少部分图像中交点数目多于一个, 且都未超过 10 个. 分析中还发现, 若灰度明显集中于几个区域, 随 k 值增大, $f_1(k)$ 在灰度集中区域有快速上升, 在其它区域变化缓慢, 可能使 $f_1(k)$ 曲线对斜率为 1 的 $f_2(k)$ 曲线反复交叉穿越.

阈值法中, 文献[13, 14]采用不同方法计算阈值搜索范围的上限和下限, 也缩减了阈值的遍历范围. 与文献[13, 14]算法相比, 新算法是基于文中证明的命题和性质, 尽量采用迭代探测方法寻找最佳阈值的可能值; 仅当 $f_1(k)$ 与 $f_2(k)$ 有多个交点时才需遍历某局部范围

的阈值,但也在 step 5 中通过迭代探测压缩遍历范围.

2.2.4 时间性能分析

传统 Otsu 法在计算最佳阈值时,需计算 $\omega_0(k)$ 、 μ_T 、 $\mu(k)$ 和 $\sigma_B^2(k)$, 其中 μ_T 等于 $\mu(L-1)$. 将计算一次 $\omega_0(k)$ 、 $\mu(k)$ 、 $\sigma_B^2(k)$ 的耗时分别记为 t_0 、 t_1 和 t_2 . 由于零阶矩 $\omega_0(k)$ 和一阶矩 $\mu(k)$ 可采用递推方式计算, 远快于对二阶矩 $\sigma_B^2(k)$ 的计算, 故有: $t_0 + t_1 \ll t_2$. 传统 Otsu 法耗时为 $L(t_0 + t_1) + Lt_2$, 其中 Lt_2 是主要部分.

本文快速算法计算一次 $f_1(k)$ 的耗时记为 t_3 . 假设计算一次浮点乘法或除法操作耗时 r_1 , 计算一次浮点加法或减法耗时 r_2 . 由 $f_1(k)$ 的定义式(7)及式(1)、(2)可知, t_3 约为 $3r_1 + 3r_2$. 对于 $\sigma_B^2(k)$ 依据式(4)计算, 将分子看作 $[\mu_T \omega_0(k) - \mu(k)]$ 与其自身的一次浮点乘法, t_2 约为 $4r_1 + 2r_2$. 通常 r_1 显著大于 r_2 , 因此 t_3 一般会小于 t_2 .

以 M 表示对 $f_1(k)$ 与 $f_2(k)$ 交点的探测次数, 以 Q 表示交点个数. 本文快速算法在 $Q=1$ 时耗时 $L(t_0 + t_1) + Mt_3$, 在 $Q>1$ 时耗时 $L(t_0 + t_1) + Mt_3 + Qt_2$. 由于 M 大幅小于 L , Q 则更小, 且 t_3 小于 t_2 , 因而本文算法总耗时少于传统 Otsu 法.

3 Otsu 阈值改进及 Otsu 快速算法应用

传统 Otsu 法在一些场合下分割性能不佳^[12,15], 当目标面积在整幅图像中占比小于 10% 时分割性能退化明显^[15]. 梯度图像中往往灰度高的像素数目较少, 考虑梯度图中目标面积占比显著低于 10% 的情况, 假设梯度图中目标灰度高于背景灰度, 且存在阈值 T 可以准确分割小目标和背景, 则应有: $\omega_1(T) \ll 0.5 < \omega_0(T)$.

式(3)中类间方差可以改写为:

$$\sigma_B^2(k) = \omega_0(k) \omega_1(k) [\mu_1(k) - \mu_0(k)]^2 \quad (13)$$

$$\text{令: } g_1(k) = \omega_0(k) \omega_1(k) = (1 - \omega_1(k)) \omega_1(k) \quad (14)$$

$$g_2(k) = [\mu_1(k) - \mu_0(k)]^2 \quad (15)$$

$g_1(k)$ 在 $\omega_1(k) = 0.5$ 时取得最大值. 由 $\omega_1(T) \ll 0.5$ 可知 $g_1(\omega_1(T))$ 远小于其峰值. 当阈值 k 从 T 开始减小并使 $\omega_1(k)$ 迅速增加时, $g_1(k)$ 亦快速增加, 但 $\mu_1(k)$ 和 $\mu_0(k)$ 作为均值往往变化相对缓慢并导致 $g_2(k)$ 变化缓慢, 此时易有 $\sigma_B^2(k)$ 大于 $\sigma_B^2(T)$, 即 Otsu 阈值易将部分背景错分为目标.

对于上述场合, 可以基于 Otsu 快速算法采用递归的方法求解阈值, 具体算法如下:

step 1 用本文的 Otsu 快速算法对图像求阈值的初值 T_i , 其中 $i=1$. 若 $\omega_1(T_i) \geq \lambda_1$, 则执行 step 2 和 step 3; 否则, T_i 即为所求阈值, 结束求解.

step 2 重新计算 $[T_i, L-1]$ 范围的灰度直方图, 其中, 令灰度 T_i 的概率为原图的 $\omega_0(T_i)$, 其余概率不变.

在 $[T_i, L-1]$ 内用本文的 Otsu 快速算法求阈值 T_{i+1} .

step 3 若原图的 $\omega_1(T_i)$ 、 $\omega_1(T_{i+1})$ 满足 $(\omega_1(T_i) - \omega_1(T_{i+1}))/0.5 < \lambda_2$, 则 T_{i+1} 即为所求阈值, 结束求解; 否则, 令 $i = i+1$, 返回 step 2.

由于目标像素比例小于 10% 时 Otsu 法性能明显退化^[15], 一般可令 $\lambda_1 = \lambda_2 = 0.1$; 若目标占比范围更明确, λ_1 、 λ_2 的值可根据情况做调整. 若忽略 $g_2(k)$ 而只考虑 $\omega_1(k)$ 对 $\sigma_B^2(k)$ 的影响, 则 $\omega_1(k) = 0.5$ 时 $g_1(\omega_1(k))$ 最大并对应 Otsu 阈值, 该情况下一次 Otsu 分割所得的背景比例也是 0.5, 故在 step 3 中将不等式的分母设为 0.5.

4 实验结果

实验的硬件条件为 T7250 2.00G CPU, 3G 内存的笔记本电脑, 软件采用 Matlab7.1. 实验中对算法程序生成 P 文件, 并采用程序执行第二次以后的计时值.

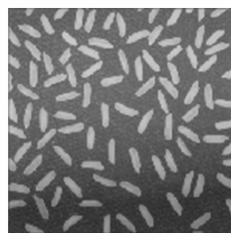
大量实验结果表明, 本文提出的一维 Otsu 快速算法给出的阈值完全符合 Otsu 准则, 运算时间与传统算法相比大幅减少. 现选取图 1 中的 (a)、(c) 和图 2 中的 (a)、(b)、(c)、(d), 给出文献[2]中传统算法与本文快速算法的时间性能比较结果, 如表 1 所示. 所选取的 6 幅图像中, $L=256$.

表 1 中所列两种算法耗时为在已知灰度直方图的情况下计算出阈值所需时间. 表 1 中两种算法的耗时均包括了计算 $\omega_0(k)$ 、 $\mu(k)$ 序列部分的耗时.

图 3 和图 4 分别展示了对布匹上的针孔和冲浪运动员的梯度图像应用传统 Otsu 法和本文的 Otsu 阈值改进算法进行分割时所得结果, 梯度图像是用 Sobel 算子对原始图做梯度处理而得, $L=256$.



(a) 建筑



(b) 大米



(c) 飞机



(d) 青椒

图2 原始图像

表 1 一维 Otsu 算法时间性能

	Lena	电路	建筑	大米	飞机	辣椒
$f_1(k)$ 与 $f_2(k)$ 交叉次数	1	3	1	1	1	1
传统算法耗时(ms)	0.374	0.293	0.312	0.286	0.350	0.371
计算 $\omega_0(k)$ 、 $\mu(k)$ 序列						
部分耗时(ms)	0.040	0.044	0.042	0.042	0.028	0.036
本文算法耗时(ms)	0.071	0.079	0.075	0.070	0.065	0.060

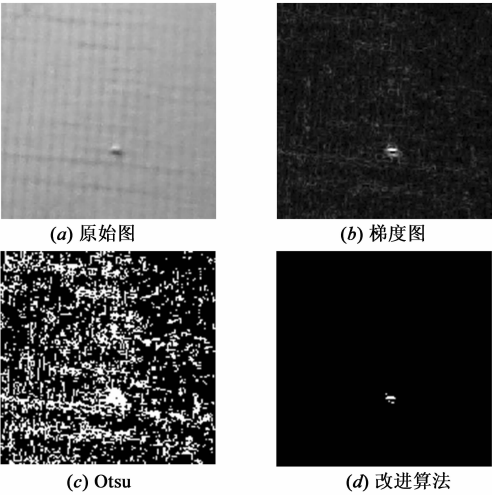


图3 对布匹上的针孔梯度图像的分割

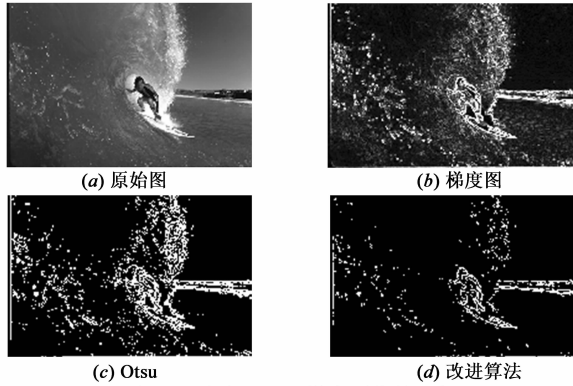


图4 对冲浪运动员梯度图像的分割

传统 Otsu 法分割针孔时误分割较多,在分割冲浪运动员时所得目标形状模糊,而本文的 Otsu 阈值改进算法有更好的分割结果.表 2 中比较了对图 3 和图 4 中梯度图像分割时几种算法的耗时.表 2 中的递归 Otsu 算法采用与本文的 Otsu 阈值改进算法相同的计算流程,但其计算 Otsu 阈值时采用传统 Otsu 法.表 2 中,本文的 Otsu 阈值改进算法耗时远小于递归 Otsu 算法,也少于传统 Otsu 法.

表 2 梯度图像分割中的算法时间

	传统 Otsu	递归 Otsu	Otsu 阈值改进算法
针孔(ms)	0.308	1.352	0.228
冲浪运动员(ms)	0.293	1.045	0.193

5 结束语

本文中利用 Otsu 准则下最佳阈值的性质,提出了一个可以快速搜索出一个 Otsu 阈值的算法,克服了传统算法穷举计算类间方差的缺点.针对传统 Otsu 法在对梯度图像中的小目标分割时性能不佳的情况,文中基于 Otsu 快速算法提出了一种递归计算阈值的 Otsu 阈值改进算法.实验结果表明,本文的 Otsu 快速算法与传统算法一样能求解 Otsu 阈值,但速度更快;与传统 Otsu 法相比,本文的 Otsu 阈值改进算法有良好的分割速度,但对小目标梯度图像有更好的分割效果.本文提出的 Otsu 快速算法有利于 Otsu 方法的实时应用;提出的 Otsu 阈值改进算法对梯度图像中的小目标分割是一种有效算法.

参考文献

[1] Mehmet Sezgin, Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation[J]. Journal of Electronic Imaging, 2004, 13(1): 146 – 165.

[2] Otsu N. A threshold selection method from gray-level histograms[J]. IEEE Transactions on System Man and Cybernetic, 1979, 9(1): 62 – 66.

[3] Sahoo P K, Soltani S, Wong A K C, Chen Y C. A survey of thresholding techniques[J]. Computer Vision, Graphics and Image Processing, 1988, 41: 233 – 260.

[4] 刘建庄, 粟文青. 灰度图像的二维 Otsu 自动阈值分割法[J]. 自动化学报, 1993, 19(1): 101 – 105.

Liu Jian-zhuang, Li Wen-qing. Automatic thresholding of gray-level pictures using two-dimension Otsu method[J]. Acta Automatica Sinica, 1993, 19(1): 101 – 105. (in Chinese)

[5] 景晓军, 李剑峰, 刘郁林. 一种基于三维最大类间方差的图像分割算法[J]. 电子学报, 2003, 31(9): 1281 – 1285.

Jing Xiao-jun, Li Jian-feng, Liu Yu-lin. Image segmentation based on 3-D maximum between-cluster variance [J]. Acta Electronica Sinica, 2003, 31(9): 1281 – 1285. (in Chinese)

[6] 汪海洋, 潘德炉, 夏德深. 二维 Otsu 自适应阈值选取算法的快速实现[J]. 自动化学报, 2007, 33(9): 968 – 971.

Wang Hai-yang, Pan De-lu, Xia De-shen. A fast algorithm for two-dimensional Otsu adaptive threshold algorithm [J]. Acta Automatica Sinica, 2007, 33(9): 968 – 971. (in Chinese)

[7] 吴成茂, 田小平, 谭铁牛. 二维 Otsu 阈值法的快速迭代算法[J]. 模式识别与人工智能, 2008, 21(6): 746 – 757.

Wu Cheng-mao, Tian Xiao-ping, Tan Tie-niu. Fast iterative algorithm for two-dimensional Otsu thresholding method [J]. PR & AI, 2008, 21(6): 746 – 757. (in Chinese)

[8] 申铨京, 龙建武, 陈海鹏, 魏巍. 三维直方图重建和降维的 Otsu 阈值分割算法[J]. 电子学报, 2011, 39(5): 1108 – 1114.

- Shen Xuan-jing, Long Jian-wu, Chen Hai-peng, Wei Wei. Otsu thresholding algorithm based on rebuilding and dimension reduction of the 3-dimensional histogram[J]. Acta Electronica Sinica, 2011, 39(5): 1108 – 1114. (in Chinese)
- [9] Reddi S S, Rudin S F, Keshavan H R. An optimal multiple threshold scheme for image segmentation[J]. IEEE Transactions on System Man and Cybernetic, 1984, 14(4): 661 – 665.
- [10] Han Lee, Rae-Hong Park. Comments on “an optimal multiple threshold scheme for image segmentation”[J]. IEEE Transactions on System Man and Cybernetic, 1990, 20(3): 741 – 742.
- [11] Huang Deng-yuan, Wang Chia-hung. Optimal multi-level thresholding using a two stage Otsu optimization approach [J]. Pattern Recognition Letters, 2009, 30: 275 – 284.
- [12] 许向阳, 宋恩民, 金良海. Otsu 准则的阈值性质分析[J]. 电子学报, 2009, 37(12): 2716 – 2719.
- Xu Xiang-yang, Song En-min, Jin Liang-hai. Characteristic analysis of threshold based on Otsu criterion[J]. Acta Electronica Sinica, 2009, 37(12): 2716 – 2719. (in Chinese)
- [13] Li Zuo-yong, Yang Jian, Liu Guang-hai, et al. Unsupervised range-constrained thresholding [J]. Pattern Recognition Letters, 2011, 32: 392 – 402.
- [14] Hu Qing-mao, Hou Zu-jun, Wieslaw L. Nowinski, et al. Supervised range-constrained thresholding[J]. IEEE Transactions on Image Processing, 2006, 15(1): 228 – 240.
- [15] Lee S U, Chung S Y, Park R H. A comparative performance study of several global thresholding techniques for segmentation[J]. Computer Vision, Graphics and Image Processing, 1990, 52(2): 171 – 190.

作者简介



何志勇 男, 1976 年 9 月出生于内蒙古包头市, 现为苏州大学机电工程学院讲师、博士研究生, 主要从事机器视觉、信号处理等方面的研究.

E-mail: hezhiyong@suda.edu.cn



孙立宁 男, 1964 年 1 月出生于黑龙江鹤岗市, 现为苏州大学机电工程学院教授、博士生导师, 主要从事先进机器人技术、微纳操作技术与装备、基于 MEMS 技术微纳器件与系统等方面的研究.

E-mail: lnsun@hit.edu.cn