

2 维最大类间平均离差阈值选取快速递推算法

吴一全 潘 喆

(南京航空航天大学信息科学与技术学院, 南京 210016)

摘 要 阈值分割是广泛使用的最为有效的图像分割方法之一。阈值选取是阈值分割的关键。Otsu 提出的基于 L_2 范数的最大类间方差法是备受关注的一种方法, 而基于 L_1 范数的最大类间平均离差法则更为简捷, 效果很好。2 维最大类间平均离差阈值选取方法分割精确、抗噪性能好, 其效果优于 2 维最大类间方差法, 但存在计算量大、难以实用等缺点。提出了 2 维最大类间平均离差阈值选取的两种不同的快速递推算法, 都可将计算复杂性由 $O(L^4)$ 减少为 $O(L^2)$ 。给出了 2 维最大类间平均离差两种快速递推算法的分割结果及运行时间, 并与原始算法及原有的快速算法进行了分析和比较。实验结果表明, 这两种递推算法都可以大幅度地提高运算速度, 运行时间可减少到原始算法的 0.1%, 使 2 维最大类间平均离差阈值分割方法更为实用, 目前已被应用于红外目标、车牌、指纹等自动识别系统中。

关键词 图像分割 阈值选取 2 维最大类间平均离差法 递推算法

中图法分类号: TP391.4 **文献标识码:** A **文章编号:** 1006-8961(2009)03-0471-06

Fast Recursive Two-dimensional Maximum between-cluster Average Deviation Thresholding Algorithms

WU Yi-quan, PAN Zhe

(School of Information and Science Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)

Abstract Thresholding is one of the widely used and efficient techniques for image segmentation in digital image processing. Threshold selection is crucial to thresholding. The maximum between-cluster variance algorithm based on L_2 -Norm, which was proposed by Otsu, is one of the most famous methods. And the maximum between-cluster average deviation thresholding algorithm based on L_1 -Norm is simpler and has good performance. The two-dimensional maximum between-cluster average deviation thresholding algorithm, which has high accuracy of segmentation and good resistance to noise, has better performance than the maximum between-cluster variance algorithm, but the two-dimensional algorithm requires a large amount of computation and is impractical in applications. In this paper, two fast recursive two-dimensional maximum between-cluster average deviation thresholding algorithms are proposed, whose computational complexities are only $O(L^2)$, while the computational complexity of the original algorithm is $O(L^4)$. Using those two recursive algorithms, the results and processing time of the two-dimensional maximum between-cluster average deviation thresholding algorithm are given, which are compared with the original algorithm. Experimental results show that both of those two recursive algorithms can greatly reduce the processing time, which is only 0.1% of that of the original algorithm. Currently the proposed algorithms have been used in automatic infrared target, vehicle license plate and fingerprint recognition system.

Keywords image segmentation, threshold selection, two-dimensional maximum between-cluster average deviation, recursive algorithm

基金项目:国家自然科学基金项目(60872065)

收稿日期:2007-05-31; **改回日期:**2007-10-23

第一作者简介:吴一全(1963 ~),男。副教授,硕士生导师。1998 年于南京航空航天大学获信息与通信工程专业博士学位。主要研究方向为图像处理与识别、视频处理与通信、网络信息安全及智能与现代信号处理。已发表学术论文 70 余篇。E-mail: gumption_s@yahoo.com.cn

1 引言

图像分割是计算机视觉和图像处理研究领域中的重要内容。阈值分割是广泛使用的最为有效的图像分割方法之一,其关键是如何选取阈值。从 20 世纪 60 年代以来,国内外学者提出了大量的阈值选取方法^[1-2],其中 Otsu 提出的 1 维最大类间方差法^[3]引起人们普遍关注,应用广泛。该方法通过图像的 1 维灰度直方图来选取阈值。1 维直方图虽然处理速度快,但不能反映图像的局部空间信息,当图像受到噪声干扰、光照不均匀等因素影响时,难以获得满意的分割效果,甚至产生分割错误。为此,1993 年刘健庄将 Otsu 法从 1 维推广到 2 维^[4],其效果较 1 维方法有明显改善,但同时由于将 1 维信息搜索拓展为 2 维信息搜索,导致运算量按指数增加,运算速度慢,难以用于实时处理。为了降低计算量,1998 年 Gong Jian 等人^[5]和 2001 年景晓军等人^[6]从减少重复计算的观点出发,提出 2 维 Otsu 法的一种快速递推算法^[5-6]。2005 年郝颖明等人^[7]在改变 2 维直方图区域划分的基础上,提出通过将 2 维阈值转换成 1 维阈值的快速算法^[7-8]。这两种快速算法都能不同程度地提高运行速度。

Otsu 的 1 维最大类间方差法实质上是在基于 L_2 范数的判决分析或最小二乘法原理的基础上推导出来的,1990 年,笔者提出的一种 1 维最大类间平均离差法^[9]则是基于 L_1 范数,比 1 维最大类间方差法计算更为简捷。同样,为进一步提高分割精度和抗噪性能,又提出了 2 维最大类间平均离差法^[10],分割效果很好,但同 2 维最大类间方差法存在类似的问题,即运算量大,运算时间长,实时性差,计算复杂性为 $O(L^4)$ (L 为图像灰度级数),运算时间通常需近 90 秒,因此,作者也同时给出了一种快速算法,虽大大提高了运行速度,但根据该快速算法编程时涉及较多的条件判断语句,势必占用不少机时。为进一步减少计算量,提高阈值分割的实时性,本文提出了 2 维最大类间平均离差法的两种快速递推算法。这两种快速算法都可分别将 2 维最大类间平均离差法的计算复杂性从 $O(L^4)$ 大大减少为 $O(L^2)$,耗时在 0.07 到 0.16 秒左右。

2 2 维最大类间平均离差阈值选取方法

设图像的尺寸为 $M \times N$,图像灰度级取 $0, 1, \dots,$

$L-1$ 。定义坐标 (m, n) 的像素点的邻域平均灰度级

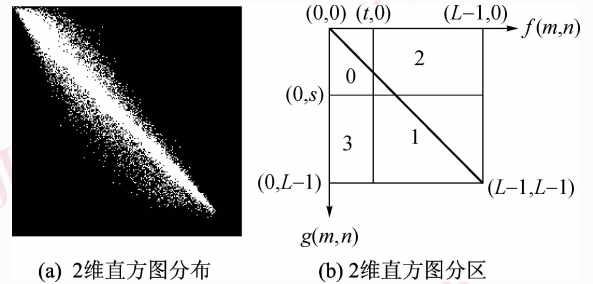
$$g(m, n) = \frac{1}{n_1 \times n_1} \sum_{i=-[n_1/2]}^{[n_1/2]} \sum_{j=-[n_1/2]}^{[n_1/2]} f(m+i, n+j) \quad (1)$$

式中, n_1 一般取奇数,如 3, 5 等, $[n_1/2]$ 表示对 $n_1/2$ 取整。

如果用 $r(i, j)$ 表示相应于灰度级,邻域平均灰度级出现的频次 ($0 \leq r(i, j) \leq M \times N$)。可定义 $p(i, j)$ 是相应于灰度级,邻域平均灰度级的联合概率密度:

$$p(i, j) = \frac{r(i, j)}{M \times N} \quad i, j = 0, 1, \dots, L-1 \quad (2)$$

这样, $\{p(i, j)\}$ 即为图像的 2 维灰度直方图,如图 1 所示。利用阈值向量 (t, s) ,可将图 1(a) 所示的 2 维直方图分割成如图 1(b) 所示的 4 个区域。假设图像的暗(亮)像素视为目标(背景),则区域 0 和目标对应,区域 1 和背景对应,而区域 2 和区域 3 表示边缘和噪声。



(a) 2维直方图分布

(b) 2维直方图分区

图 1 2 维直方图分布及分区

Fig. 1 Two-dimensional histogram distribution and quadrants

2007 年,笔者提出的 2 维最大类间平均离差法,定义目标和背景在灰度级与邻域平均灰度级上的平均绝对距离(即类间平均离差)作为可分离性度量,经推导整理后,得到:

$$d_B(t, s) = 2 [|\mu_f(t, s) - \mu_{f\omega_0}(t, s)| + |\mu_g(t, s) - \mu_{g\omega_0}(t, s)|] \quad (3)$$

其中,

$$\omega_0(t, s) = \sum_{i=0}^t \sum_{j=0}^s p(i, j) \quad (4)$$

$$\mu = (\mu_f, \mu_g)^T = \left(\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} ip(i, j), \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} jp(i, j) \right)^T \quad (5)$$

$$\begin{aligned} \mu(t, s) &= (\mu_f(t, s), \mu_g(t, s))^T \\ &= \left(\sum_{i=0}^t \sum_{j=0}^s ip(i, j), \sum_{i=0}^t \sum_{j=0}^s jp(i, j) \right)^T \end{aligned} \quad (6)$$

上式中, $\omega_0(t, s)$ 表示目标概率; $\mu = (\mu_f, \mu_g)^T$ 表示 2

维直方图总均值向量,其分量 μ_f, μ_g 分别为2维直方图灰度级总均值和邻域平均灰度级总均值, T 为转置; $\mu(t, s)$ 表示阈值向量为 (t, s) 时的2维直方图均值向量,其分量 $\mu_f(t, s)$ 和 $\mu_g(t, s)$ 分别为阈值向量为 (t, s) 时的灰度级均值和邻域平均灰度级均值。

选择可分离性度量 $d_B(t, s)$ 最大值相应的阈值向量 (t_0, s_0) 作为2维最大类间平均离差法的最佳阈值向量,即

$$(t_0, s_0) = \arg \max_{0 \leq t, s < L-1} \{d_B(t, s)\} \quad (7)$$

从上述阈值选取方法式(3)~式(7)可以看出,计算 $d_B(t, s)$ 需要计算 $\omega_0(t, s)$ 、 $\mu_f(t, s)$ 和 $\mu_g(t, s)$ 以及 μ_f 和 μ_g 。对于同一幅图像, μ_f 和 μ_g 是固定的,对于每一个阈值向量 (t, s) ,如果每次计算判别函数 $d_B(t, s)$ 都重新从 $i=0, j=0$ 开始计算 $\omega_0(t, s)$ 、 $\mu_f(t, s)$ 和 $\mu_g(t, s)$,势必造成大量的重复计算,计算复杂性都为 $O(L^2)$,而共有 L^2 个 (t, s) 向量,从而使总的计算复杂性达到 $O(L^4)$ 。 t 和 s 的取值越大, $d_B(t, s)$ 的计算速度也就越慢。

3 2维最大类间平均离差阈值选取的两种快速递推算算法

2维最大类间平均离差阈值选取算法是对2维最大类间方差法的一种改进,分割准确,抗噪性能好,但也存在计算量十分庞大,实时性差的缺点,难以在实际中得到广泛应用。为了提高算法的运算速度,进一步优化算法,下面分别给出2维最大类间平均离差法的两种快速递推算算法。

3.1 递推算算法1

将递推思想运用于2维最大类间平均离差法,推导出2维最大类间平均离差法的快速递推公式:

$$\begin{aligned} \omega_0(t, s) &= \sum_{i=0}^t \sum_{j=0}^{s-1} p(i, j) + \sum_{i=0}^t p(i, s) \\ &= \omega_0(t, s-1) + \sum_{i=0}^{t-1} p(i, s) + p(t, s) \\ &= \omega_0(t, s-1) + \omega_0(t-1, s) - \omega_0(t-1, s-1) + p(t, s) \\ \mu_f(t, s) &= \sum_{i=0}^t \sum_{j=0}^{s-1} ip(i, j) + \sum_{i=0}^t ip(i, s) \\ &= \mu_f(t, s-1) + \sum_{i=0}^{t-1} ip(i, s) + tp(t, s) \\ &= \mu_f(t, s-1) + \mu_f(t-1, s) - \end{aligned} \quad (8)$$

$$\mu_f(t-1, s-1) + tp(t, s) \quad (9)$$

同理

$$\begin{aligned} \mu_g(t, s) &= \mu_g(t, s-1) + \mu_g(t-1, s) - \mu_g(t-1, s-1) + sp(t, s) \end{aligned} \quad (10)$$

按以下方式赋初值:

$$\omega_0(t, 0) = \omega_0(t-1, 0) + p(t, 0)$$

$$\omega_0(0, s) = \omega_0(0, s-1) + p(0, s)$$

$$\mu_f(t, 0) = \mu_f(t-1, 0) + tp(t, 0)$$

$$\mu_f(0, s) = \mu_0(0, s-1)$$

$$\mu_g(t, 0) = \mu_g(t-1, 0) + sp(t, 0)$$

$$\mu_g(0, s) = \mu_g(0, s-1) + sp(0, s)$$

$$\omega_0(0, 0) = p(0, 0)$$

$$\mu_f(0, 0) = 0$$

$$\mu_g(0, 0) = 0$$

根据上述递推公式,每计算一次判别函数 $d_B(t, s)$,都无需重新计算 $\omega_0(t, s)$ 、 $\mu_f(t, s)$ 和 $\mu_g(t, s)$,只需分别利用前面得到的 $\omega_0(t, s-1)$ 、 $\omega_0(t-1, s)$ 、 $\omega_0(t-1, s-1)$ 和 $\mu_f(t, s-1)$ 、 $\mu_f(t-1, s)$ 、 $\mu_f(t-1, s-1)$ 以及 $\mu_g(t, s-1)$ 、 $\mu_g(t-1, s)$ 、 $\mu_g(t-1, s-1)$,这样就只要在2维直方图范围内搜索 (t, s) ,因此计算复杂性仅为 $O(L^2)$ 。

这种算法中,每次循环只用到了 $\omega_0(t, \cdot)$ 和 $\omega_0(t-1, \cdot)$ 、 $\mu_f(t, \cdot)$ 和 $\mu_f(t-1, \cdot)$ 及 $\mu_g(t, \cdot)$ 和 $\mu_g(t-1, \cdot)$ 6列数据,因此,只需在每次循环结束的时候将 $\omega_0(t, \cdot)$ 、 $\mu_f(t, \cdot)$ 、 $\mu_g(t, \cdot)$ 赋值于 $\omega_0(t-1, \cdot)$ 、 $\mu_f(t-1, \cdot)$ 及 $\mu_g(t-1, \cdot)$,这样,全部计算所用存储空间仅为 $6 \times L$,而原始算法的存储空间达 $3 \times L^2$ 。

3.2 递推算算法2

若利用与上述不同的另一种递推方式,同样也可以消除原始算法中大量的重复计算。若记

$$\omega_i(s) = \sum_{i=0}^t p(i, s)$$

$$\mu_{fi}(s) = \sum_{i=0}^t ip(i, s)$$

$$\mu_{gi}(s) = s \sum_{i=0}^t p(i, s)$$

则存在如下递推公式:

$$\begin{aligned} \omega_i(s) &= \sum_{i=0}^{t-1} p(i, s) + p(t, s) \\ &= \omega_{i-1}(s) + p(t, s) \end{aligned} \quad (11)$$

$$\mu_{fi}(s) = \sum_{i=0}^{t-1} ip(i, s) + tp(t, s) =$$

$$\mu_{f(t-1)}(s) + tp(t,s) \tag{12}$$

$$\begin{aligned} \mu_{gt}(s) &= s \sum_{i=0}^{t-1} p(i,s) + sp(t,s) \\ &= \mu_{g(t-1)}(s) + sp(t,s) \end{aligned} \tag{13}$$

$$\begin{aligned} \omega_0(t,s) &= \sum_{i=0}^t \sum_{j=0}^s p(i,j) = \sum_{i=0}^t \sum_{j=0}^{s-1} p(i,j) + \sum_{i=0}^t p(i,s) \\ &= \omega_0(t,s-1) + \omega_t(s) \end{aligned} \tag{14}$$

$$\begin{aligned} \mu_f(t,s) &= \sum_{i=0}^t \sum_{j=0}^s ip(i,j) = \sum_{i=0}^t \sum_{j=0}^{s-1} ip(i,j) + \sum_{i=0}^t ip(i,s) \\ &= \mu_f(t,s-1) + \mu_{fi}(s) \end{aligned} \tag{15}$$

$$\begin{aligned} \mu_g(t,s) &= \sum_{i=0}^t \sum_{j=0}^s jp(i,j) = \sum_{i=0}^t \sum_{j=0}^{s-1} jp(i,j) + \sum_{i=0}^t sp(i,s) \\ &= \mu_g(t,s-1) + \mu_{gt}(s) \end{aligned} \tag{16}$$

在计算前需对各点进行如下初始化:

$$\begin{aligned} \omega_0(0) &= p(0,0) \\ \mu_{f0}(0) &= 0 \\ \mu_{g0}(0) &= 0 \\ \omega_0(0,0) &= p(0,0) \\ \mu_f(0,0) &= 0 \\ \mu_g(0,0) &= 0 \\ \omega_t(0) &= \omega_{t-1}(0) + p(t,0) \\ \mu_{fi}(0) &= \mu_{f(t-1)}(0) + tp(t,0) \\ \mu_{gt}(0) &= \mu_{g(t-1)}(0) \\ \omega_0(0,s) &= \omega_0(0,s-1) + p(0,s) \\ \mu_f(0,s) &= \mu_f(0,s-1) \\ \mu_g(0,s) &= \mu_g(0,s-1) + sp(0,s) \end{aligned}$$

这种递推算法同样只需在 2 维直方图范围内搜索 (t,s) , 每次分别按式 (14)、式 (15) 或式 (16) 计算 $\omega_0(t,s)$ 、 $\mu_f(t,s)$ 或 $\mu_g(t,s)$, 只要利用前面得到的 $\omega_0(t,s-1)$ 、 $\mu_f(t,s-1)$ 或 $\mu_g(t,s-1)$ 及当前的 $\omega_t(s)$ 、 $\mu_{fi}(s)$ 或 $\mu_{gt}(s)$; 而 $\omega_t(s)$ 、 $\mu_{fi}(s)$ 和 $\mu_{gt}(s)$ 也可以分别按式 (11)、式 (12) 和式 (13) 通过迭代算出, 不用逐点累加, 因此, 只需做二重循环, 计算复杂性为 $O(L^2)$, 但其占用的存储空间为 $6 \times L^2$, 即以较小的存储容量为代价, 消除了原算法的重复计算现象, 大大提高了运算速度。

4 实验结果和分析

利用上述 2 维最大类间平均离差法的两种快速递推算法, 针对不同类型的灰度图像做了大量实验, 实验结果表明, 分割结果准确, 且运行速度大幅度提

高。先选取其中的 6 幅图像, 分别为车牌、指纹、米粒、螺钉、几何图形及有噪 Lena 图。各幅图像的原始灰度级图像及分别用 2 维最大类间平均离差法和 2 维最大类间方差法分割后得到的二值图像如图 2 所示, 其分割阈值如对表 1 所示。表 2 分别给出了在 P4 1.60 GHz CPU, 512 M 内存的条件下, 相应各幅图像的 2 维最大类间平均离差快速递推算法的运行时间。为了便于比较, 表中同时列出了 2 维最大类间平均离差法的原始算法及其原有快速算法^[10]和 2 维最大类间方差法的原始算法^[4]及其快速算法^[5,8]的运行时间。

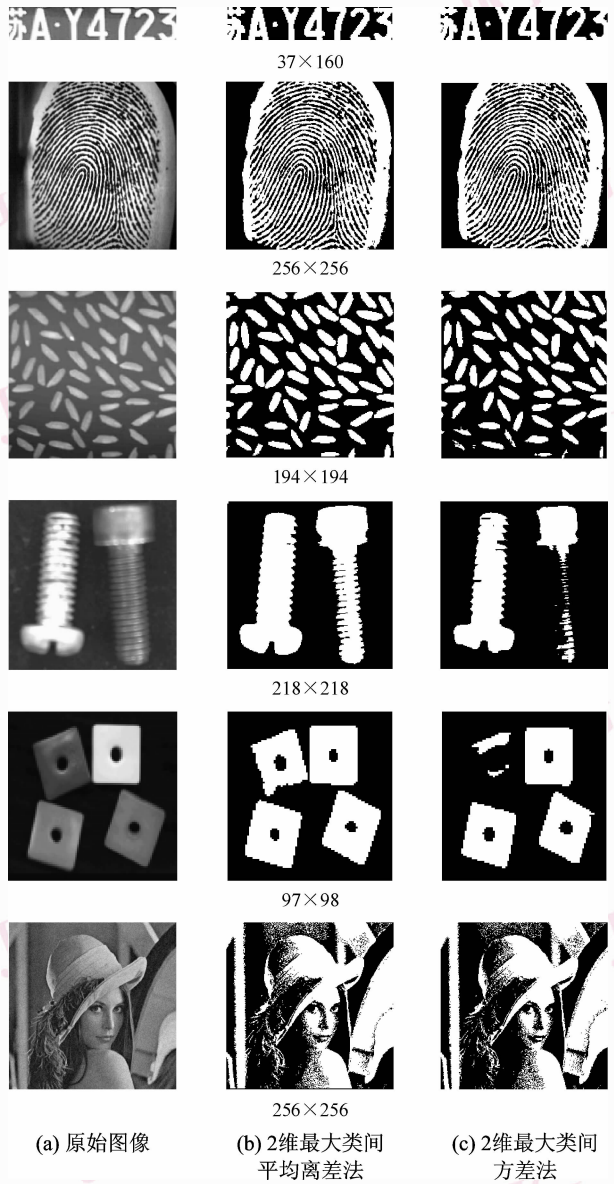


图 2 原始图像和分割后的二值图
Fig. 2 Original and thresholded images

表 1 图像分割的阈值

Tab. 1 Threshold used in image segmentation

图像	车牌	指纹	米粒	螺钉	几何图形	有噪 Lena 图
2 维最大类间平均离差法	160,171	103,125	109,111	91,139	58,94	105,104
2 维最大类间方差法	175,193	111,133	126,125	133,132	90,104	108,107

从图 2 可以看出,对于不同类型及有噪图像,2 维最大类间平均离差法都能选取出较理想的阈值,用该阈值分割后的二值图像区域内部均匀,目标边界形状准确。图 2 中,对于车牌、指纹、有噪 Lena 图,2 维最大类间平均离差法与 2 维最大类间方差

法的分割效果相近,但对于米粒、螺钉、几何图形,2 维最大类间平均离差法的分割结果明显优于 2 维最大类间方差法。

还采用区域内部均匀性评价准则^[11]对 2 维最大类间平均离差阈值分割方法与 2 维最大类间方差法进行了客观定量实验评价。表 3 列出了相应上述 6 幅图像分别采用 2 维最大类间平均离差阈值分割方法和 2 维最大类间方差法所得到的区域内部均匀性比较结果。该结果已进行归一化,越接近于 1 则内部均匀性越好。从表 3 的数据中可以看出,客观定量分析的结果与上述根据分割图像进行主观定性评价所得的结果一致。因此,2 维最大类间平均离差法是一种很好的阈值选取算法,适合广泛的使用。

表 2 图像分割的运行时间

Tab. 2 Processing time spent in image segmentation

单位:s

图	像	车牌	指纹	米粒	螺钉	几何图形	有噪 Lena 图
2 维最大类间平均离差法	原始算法 ^[10]	87.04	87.46	87.43	86.96	87.07	87.38
	原有快速算法 ^[10]	0.620	0.624	0.639	0.627	0.604	0.628
	快速递推算算法一	0.156	0.139	0.132	0.131	0.130	0.142
	快速递推算算法二	0.082	0.091	0.070	0.089	0.081	0.074
2 维最大类间方差法	原始算法 ^[4]	95.68	93.25	95.07	93.34	93.00	97.23
	文献[8]快速算法	0.629	0.676	0.648	0.644	0.638	0.673
	文献[5]快速算法	0.122	0.093	0.119	0.121	0.129	0.115

表 3 阈值分割区域内部均匀性比较

Tab. 3 The internal homogeneous comparison of thresholding

图像	车牌	指纹	米粒	螺钉	几何图形	有噪 Lena 图
2 维最大类间平均离差法	0.928 9	0.569 1	0.802 7	0.774 2	0.973 1	0.689 9
2 维最大类间方差法	0.928 8	0.568 1	0.802 2	0.769 8	0.972 6	0.689 8

从表 2 可知,上述 6 幅图像的 2 维最大类间平均离差法的原始算法运行时间都需近 90 秒,实时性差,难以在实际中应用,而本文所提出的 2 维最大类间平均离差法的两种快速递推算算法都使得运算速度大大提高,运行时间大都在 0.07 至 0.16 秒左右,甚至可以减少到原始算法的 0.1%,仅为原有快速算法的 1/4 左右,这使得 2 维最大类间平均离差法更具实用性。

从本文提出的两种快速递推算算法的相互比较来看,由于算法二涉及的运算量比算法一要少,故算法

二的运行时间不到算法一的 2/3,然而算法二的存储空间是算法一的 L 倍,因此,两种快速递推算算法各有优势。

而从 2 维最大类间平均离差法与 2 维最大类间方差法的运行时间比较来看,无论是原始算法还是快速算法,前者的运行时间都要略短一些,可见 2 维最大类间平均离差法在运行速度上更具优越性。

5 结 论

Otsu 的最大类间方差法在图像阈值分割中应用普遍,但因最大类间方差法 2 维算法的性能较 1 维方法有明显改善而备受关注。而最近提出的 2 维最大类间平均离差法分割准确,抗噪性能好,其效果优于 2 维最大类间方差法。但因 2 维算法普遍存在搜索空间很大的问题,导致运算量按指数增加,运算速度大大减慢,在实际中难以应用。本文提出的 2 维最大类间平均离差法的两种快速递推算算法,消除了类间平均离差计算过程中的重复计算,都可将计算复杂性由 $O(L^4)$ 减少为 $O(L^2)$ 。本文的实验结果

表明,在各种类型图像下,这两种快速递推算法都可以大幅度地提高计算速度,运行时间可减少到原始算法的 0.1%。这两种递推算法各有优势,算法二的计算时间不到算法一的 2/3,但所占用的存储空间却是算法一的 L 倍,因此,可根据实际需要选择相应的快速算法。本文提出的算法将使 2 维最大类间平均离差阈值分割方法在工程中更为实用。目前已被应用于红外目标、车牌、指纹等自动识别系统中。

参考文献 (References)

- 1 Wu Yi-quan, Zhu Zhao-da. 30 Years (1962—1992) of the developments in threshold selection methods in image processing (I) [J]. Journal of Data Acquisition & Processing, 1993, **8**(3): 193-201. [吴一全,朱兆达. 图像处理中阈值选取方法 30 年 (1962—1992) 的进展 (一) [J]. 数据采集与处理, 1993, **8**(3): 193-201.]
- 2 Wu Yi-quan, Zhu Zhao-da. 30 Years (1962—1992) of the developments in threshold selection methods in image processing (II) [J]. Journal of Data Acquisition & Processing, 1993, **8**(4): 268-277. [吴一全,朱兆达. 图像处理中阈值选取方法 30 年 (1962—1992) 的进展 (二) [J]. 数据采集与处理, 1993, **8**(4): 268-282.]
- 3 Otsu N. A threshold selection method from gray-level histogram [J]. IEEE Transactions on Systems Man and Cybernetics, 1979, SMC-9 (1): 62-66.
- 4 Liu Jian-zhuang, Li Wen-qing. The automatic thresholding of gray-level pictures via two-dimensional Otsu method [J]. Acta Automatica Sinica, 1993, **19**(1): 101-105. [刘健庄,栗文青. 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.]
- 5 Gong Jian, Li Li-yuan, Chen Wei-nan. Fast recursive algorithms for two-dimensional thresholding [J]. Pattern Recognition, 1998, **31**(3): 295-300.
- 6 Jing Xiao-jun, Cai An-ni, Sun Jing-ao. Image segmentation based on 2D maximum between-cluster variance [J]. Journal of China Institute of Communications, 2001, **22**(4): 71-76. [景晓军,蔡安妮,孙景鳌. 一种基于二维最大类间方差的图像分割算法 [J]. 通信学报, 2001, **22**(4): 71-76.]
- 7 Liang Guang-ming, Liu Dong-hua, Li Bo, et al. Improvement for two-dimensional adaptive thresholding segmentation algorithm [J]. Techniques of Automation and Applications, 2002, **21**(5): 43-47. [梁光明,刘东华,李波等. 二维 Otsu 自适应阈值分割算法的改进 [J]. 自动化技术与应用, 2002, **21**(5): 43-47.]
- 8 Hao Ying-ming, Zhu Feng. Fast algorithm for two-dimensional Otsu adaptive threshold algorithm [J]. Journal of Image and Graphics, 2005, **10**(4): 484-488. [郝颖明,朱枫. 二维 Otsu 自适应阈值的快速算法 [J]. 中国图象图形学报, 2005, **10**(4): 484-488.]
- 9 Wu Yi-quan, Liu Bing, Wang Hou-shu. Digital image processing of the temperature field on liquid-crystal display [J]. Journal of Data Acquisition & Processing, 1990, **5**(3): 23-30. [吴一全,刘冰,王厚枢. 液晶显示温度场的数字图像处理 [J]. 数据采集与处理, 1990, **5**(3): 23-30.]
- 10 Wu Yi-quan, Hao Hong-jie. Image segmentation based on 2D maximum between-cluster average windage and its fast algorithm [J]. Application Research of Computers, 2007, **24**(supp): 945-946, 955. [吴一全,郝红杰. 二维最大类间离差及其快速分割算法 [J]. 计算机应用研究, 2007, **24**(增): 945-946, 955.]
- 11 Zhang Yu-jin. A classification and comparison of evaluation techniques for image segmentation [J]. Journal of Image and Graphics, 1996, **1**(2): 151-158. [章毓晋. 图像分割评价技术分类和比较 [J]. 中国图象图形学报, 1996, **1**(2): 151-158.]