

# 二维直方图准分的 Otsu 图像 分割及其快速实现

张新明, 孙印杰, 郑延斌

(河南师范大学计算机与信息技术学院, 河南新乡 453007)

**摘要:** 传统二维 Otsu 法主要由于对二维直方图采用主对角线区域概率和近似为 1 的假设等原因, 以致分割结果不够准确. 针对此问题, 提出了一种二维直方图准分的 Otsu 快速图像分割方法. (1) 准确选择邻域模板构建二维直方图并将 Otsu 阈值法用于此直方图上以便提高分割性能; (2) 对二维直方图主对角线上的目标和背景两区域的 Otsu 公式中对应量准确取值使阈值选取更准确; (3) 对二维直方图投影进行分析得到其特性, 并证明三个定理的存在, 利用此特性和三个定理导出新型、快速的递推算法来降低计算复杂度. 实验结果表明: 与当前二维 Otsu 阈值法相比, 本文提出的方法分割更准确和抗噪性更强, 而且其运行时间少, 与当前二维 Otsu 斜分递推算法的运行时间相近.

**关键词:** 图像分割; 阈值法; 二维 Otsu 法; 递推算法; 准分

**中图分类号:** TN911.73; TP391.41 **文献标识码:** A **文章编号:** 0372-2112 (2011) 08-1778-07

## Precise Two-Dimensional Otsu's Image Segmentation and Its Fast Recursive Realization

ZHANG Xin-ming, SUN Yin-jie, ZHENG Yan-bin

(College of Computer and Information Technology, Henan Normal University, Xinxiang, Henan 453007, China)

**Abstract:** In view of the inaccurate segmentation in the traditional two-dimensional (2-D) Otsu's thresholding method mainly owing to the supposition that the sum of probabilities of main-diagonal districts in 2-D histogram is approximately one, a fast and precise 2-D Otsu's image thresholding method is presented in this paper. A 2-D histogram was created with the select neighborhood and Otsu's method was used on the 2-D histogram in order to obtain better segmentation performance. The probabilities and the mean gray levels in the objects and the background of 2-D histogram main-diagonal districts were calculated precisely to get a more accurate threshold. The 2-D histogram projection was analyzed to get its features, three theorems were proved, and a novel recursive approach was inferred with the features and the theorems to reduce the computational complexity. Experimental results show that the proposed method achieves more accurate segmentation results and more robust anti-noise than the current 2-D Otsu's thresholding methods, and that its running time is much less, almost the same as that of the current Otsu's recursive algorithm based on 2-D histogram oblique segmentation.

**Key words:** image segmentation; thresholding method; two-dimensional Otsu's method; recursive algorithm; precise segmentation

### 1 引言

阈值分割因简单有效在图像识别与计算机视觉等领域得到广泛的应用, 常用的阈值法有最大类间方差法、最大熵法和最小交叉熵法等<sup>[1]</sup>. 其中最大类间方差法<sup>[2]</sup>是 Otsu 最先提出, 它以目标和背景类间方差最大为阈值选取准则, 依据一维直方图选取阈值, 由于没有利用图像的局部空间信息, 当存在噪声干扰或灰度分布交叉时分割效果不佳. 因此, 刘健庄等<sup>[3]</sup>将其拓展到

了二维, 分割效果有明显改善, 但同时导致运算量大幅度增加, 难以实时处理. 为此, 人们提出了多种基于二维 Otsu 阈值选取的快速算法<sup>[4~6]</sup>. 然而, 上述二维方法及其快速算法中都将二维直方图分成四个矩形区域, 计算阈值时仅考虑两个沿对角线的矩形区域进行分割, 由于引入了近似所作的假设与实际不符等原因, 分割结果不够准确. 最近, 范九伦等和吴一全等分别提出了二维 Otsu 曲线阈值法<sup>[7]</sup>和二维直方图区域斜分法<sup>[8]</sup>, 其基本原理相同, 均摒弃了近似的假设, 采用与主对角线垂直的

斜线按灰度级与邻域平均灰度级之和的大小进行阈值分割,分割效果得到了较大的提高,尤其是其快速递推算法极大地提高了分割速度(这两种方法在本文中统称二维 Otsu 斜分法,简称斜分法 Oblique Segmentation Method, OSM),但斜分法普遍性不强<sup>[9]</sup>.另外,构建二维直方图的邻域图像对分割结果影响较大,一般在构建二维直方图时,采用的是每个像素点处计算其邻域的平均灰度值得到邻域图像,即邻域图像是均值滤波得到的,而均值滤波会使图像变得模糊,细节丢失.鉴于以上原因,本文提出了一种二维直方图准分的 Otsu 快速图像分割方法.既丢弃近似的假设和对邻域模板进行改进,又利用二维直方图灰度分布的特点推出新型的递推算法来降低计算复杂度,以便获得一个分割性能好、运行速度快的图像分割算法.

## 2 二维 Otsu 阈值法

### 2.1 二维直方图

设一幅  $M \times N$  的图像  $f(x, y)$  ( $1 \leq x \leq M, 1 \leq y \leq N$ ), 在每个像素点处计算  $n \times m$  邻域的平均灰度值,得到一幅平滑的图像  $g(x, y)$ , 二者灰度级都为  $0, 1, \dots, L$ .  $n \times m$  的邻域可由下面模板表征:

$$\frac{1}{D} \begin{bmatrix} d_{1,1} & \cdots & d_{1,j} & \cdots & d_{1,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{i,1} & \cdots & d_{i,j} & \cdots & d_{i,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{n,1} & \cdots & d_{n,j} & \cdots & d_{n,m} \end{bmatrix}$$

其中,  $L$  一般取 255,  $n$  和  $m$  取大于 1 的奇数, 它们可以相等也可以不等. 邻域模板的方式对二维阈值化方法的分割结果有影响, 多数文献都将  $n$  和  $m$  取 3,  $d_{i,j} = 1$ ,

$D = 9$  也就是 8-邻域模板:  $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ ; 而文献[10]采

用 4-角域模板:  $\frac{1}{4} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ , 即  $n$  和  $m$  取 3, 但  $d_{1,1}$ ,

$d_{1,3}$ ,  $d_{3,1}$  和  $d_{3,3}$  取值为 1, 其它为 0,  $D = 4$ ; 与 8-邻域模板不同, 此种模板构建的二维直方图由于能更好地体现邻域中心像素的灰度值与所参考像素灰度值的差异, 所以能提高分割性能<sup>[10]</sup>. 设  $r(i, j)$  为图像  $f(x, y)$  中灰度级为  $i$  且  $g(x, y)$  灰度级为  $j$  的像素对的个数, 那么, 可以定义二元组  $(i, j)$  在图像和其邻域滤波图像中出现的联合概率:  $p(i, j) = r(i, j) / (M \times N)$ ,  $i, j = 0, 1, \dots, L$ ; 显然,  $0 \leq p(i, j) \leq 1$  且  $\sum_{i=0}^L \sum_{j=0}^L p(i, j) = 1$ . 这样得到了图像的二维直方图, 它是一个  $(L+1) \times (L+1)$  的矩阵, 如图 1 所示. 假设阈值向量  $(t, s)$  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, 其灰度值与邻域灰度值是相似的, 而对于目标和背景边缘处的像素, 其灰度值与邻域灰度值有很大的不同. 区域 1 和 2 代表目标或背景, 区域 3 和 4 表示边缘点及噪声. 由于边缘点和噪声点占少数, 在传统的二维阈值法中都假设二维直方图中远离对角线的分量近似为零<sup>[3~6]</sup>, 即  $p(i, j) \approx 0$  ( $t < i < L, 0 \leq j \leq s$ ) 和  $(0 \leq i \leq t, s < j < L)$ .

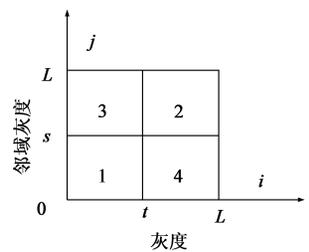


图 1 二维直方图

### 2.2 二维 Otsu 的阈值选择方法

设在二维直方图中存在两个类如图 1 中的区域 1 和 2 所示, 则两类概率分别为  $p_1(t, s)$  和  $p_2(t, s)$ , 相应的类内均值向量分别为:  $[m_{i1}(t, s)/p_1(t, s), m_{j1}(t, s)/p_1(t, s)]^T$  和  $[m_{i2}(t, s)/p_2(t, s), m_{j2}(t, s)/p_2(t, s)]^T$ , 其中:

$$p_1(t, s) = \sum_{i=0}^t \sum_{j=0}^s p(i, j), p_2(t, s) = \sum_{i=t+1}^L \sum_{j=s+1}^L p(i, j) \quad (1)$$

$$m_{i1}(t, s) = \sum_{i=0}^t \sum_{j=0}^s i p(i, j), m_{j1}(t, s) = \sum_{i=0}^t \sum_{j=0}^s j p(i, j) \quad (2)$$

$$m_{i2}(t, s) = \sum_{i=t+1}^L \sum_{j=s+1}^L i p(i, j), m_{j2}(t, s) = \sum_{i=t+1}^L \sum_{j=s+1}^L j p(i, j) \quad (3)$$

二维直方图的总均值向量  $m_T$  为:

$$m_T = (m_{Ti}, m_{Tj})^T = \left[ \sum_{i=0}^L \sum_{j=0}^L i p(i, j), \sum_{i=0}^L \sum_{j=0}^L j p(i, j) \right]^T \quad (4)$$

如果用  $p_1, p_2$  等来简写  $p_1(t, s), p_2(t, s)$  等, 那么类间方差测度为:

$$tr\sigma_B = p_1[(m_{i1}/p_1 - m_{Ti})^2 + (m_{j1}/p_1 - m_{Tj})^2] + p_2[(m_{i2}/p_2 - m_{Ti})^2 + (m_{j2}/p_2 - m_{Tj})^2] \quad (5)$$

选取的最佳阈值向量应满足:

$$(t^*, s^*)^T = \text{Arg} \max_{0 < t < L, 0 < s < L} \{tr\sigma_B(t, s)\} \quad (6)$$

传统二维 Otsu 方法作了如下的假设: 假设二维直方图的副对角线区域图像的概率忽略不计, 于是有:

$$p_1 + p_2 \approx 1, m_{Ti} \approx m_{i1} + m_{i2}, m_{Tj} \approx m_{j1} + m_{j2} \quad (7)$$

将式(7)代入式(5), 有:

$$tr\sigma_B \approx [(m_{Ti} p_1 - m_{i1})^2 + (m_{Tj} p_1 - m_{j1})^2] / [p_1(1 - p_1)] \quad (8)$$

以上方法称作二维 Otsu 近分法, 简称近分法 (Approximate Segmentation Method, ASM). 对于近分法, 每对

$(t, s)$  计算两类先验概率和类内均值的计算复杂度  $O((L+1)^2)$ , 且共有  $(L+1)^2$  对  $(t, s)$ , 因而计算二维 Otsu 的总计算复杂度为  $O((L+1)^4)$ , 运算量惊人. Gong 等人提出了二维 Otsu 近分法的快速递推算法<sup>[4]</sup>, 给出了  $p_1(t, s)$ 、 $m_{i1}(t, s)$  和  $m_{j1}(t, s)$  的快速递推公式, 计算复杂度从  $O((L+1)^4)$  减少为  $O((L+1)^2)$ . 但由于将副对角线区域图像的概率忽略不计这种假设与实际不符等原因, 因此分割结果不够准确.

### 3 二维直方图准分的 Otsu 阈值分割法

#### 3.1 准确选择构建二维直方图的邻域模板

由于用邻域图像构建的二维直方图决定二维 Otsu 法阈值的获取, 另外邻域图像还直接用来分割图像, 所以邻域图像质量好坏直接影响分割效果. 邻域图像一般由低频滤波得到, 而低频滤波方式通常由邻域模板来决定, 正如上文所述, 常见的邻域图像都是由均值滤波得到的, 均值滤波除了能有效消除高斯噪声外, 存在不足. 为了提高二维 Otsu 法的分割效果, 本文准确选择了邻域模板. 该模板是基于中值滤波技术, 而中值滤波法是一种非线性平滑技术, 它是通过从图像中的某个采样窗口取出  $n$  个数据进行排序, 然后用排序后的中值取代要处理的数据, 中值滤波的优点是在较好地去除噪声(尤其是椒盐噪声)的同时能够较好地保持目标图像的细节<sup>[11]</sup>. 本文选择 8-去心邻域窗口, 即

$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ , 此窗口滤波的工作原理是: 用图像中每个

像素点周围的 8 个像素的灰度值首先进行排序, 然后对排序后第四和第五值的均值作为输出. 如此窗口有两个特点: (1) 采用了 8-邻域点的灰度值, 舍弃了邻域中心的灰度值, 尽量发挥每一个邻域点的作用, 提高邻域中心像素的灰度值与所参考像素灰度值的差异, 更好地构建二维直方图, 提高分割效果<sup>[10]</sup>; (2) 从工作原理上看, 这种滤波窗口兼有中值滤波和均值滤波的作用, 所以在有效消除椒盐噪声的同时, 又能对高斯噪声有一定的抑制, 因此, 此模板有较强的适应性, 因为实际图像不可能只含一类噪声.

#### 3.2 准确计算二维 Otsu 的阈值

准确计算二维 Otsu 的阈值, 就是对式(5)中的六个量  $p_1(t, s)$ 、 $m_{i1}(t, s)$ 、 $m_{j1}(t, s)$ 、 $p_2(t, s)$ 、 $m_{i2}(t, s)$  和  $m_{j2}(t, s)$  不作近似计算, 而是进行准确计算, 这种方法称为二维 Otsu 准分法, 简称准分法 (Precise Segmentation Method, PSM). 为了快速实现准分法, 文献[12]使用了递推算法, 但此算法递推有误. 本文参照文献[4]获得准分法的一般递推算法, 主要过程描述如下:

#### 3.2.1 $p_1(t, s)$ 、 $m_{i1}(t, s)$ 和 $m_{j1}(t, s)$ 的递推

(1) 若  $t=0$  且  $s>0$ , 则  $m_{i1}(0, s)=0$ ,

$$p_1(0, s) = p_1(0, s-1) + \mathbf{p}(0, s) \quad (9)$$

$$m_{j1}(0, s) = m_{j1}(0, s-1) + \mathbf{sp}(0, s) \quad (10)$$

(2) 若  $t>0$  且  $s=0$ , 则  $m_{j1}(t, 0)=0$ ,

$$p_1(t, 0) = p_1(t-1, 0) + \mathbf{p}(t, 0) \quad (11)$$

$$m_{i1}(t, 0) = m_{i1}(t-1, 0) + \mathbf{tp}(t, 0) \quad (12)$$

(3) 若  $t>0$  且  $s>0$ , 则:

$$p_1(t, s) = p_1(t, s-1) + p_1(t-1, s) - p_1(t-1, s-1) + \mathbf{p}(t, s) \quad (13)$$

$$m_{i1}(t, s) = m_{i1}(t, s-1) + m_{i1}(t-1, s) - m_{i1}(t-1, s-1) + \mathbf{tp}(t, s) \quad (14)$$

$$m_{j1}(t, s) = m_{j1}(t, s-1) + m_{j1}(t-1, s) - m_{j1}(t-1, s-1) + \mathbf{sp}(t, s) \quad (15)$$

#### 3.2.2 $p_2(t, s)$ 、 $m_{i2}(t, s)$ 和 $m_{j2}(t, s)$ 的递推

(1) 若  $t=L$  且  $s<L$ , 则:

$$p_2(L, s) = p_2(L, s+1) + \mathbf{p}(L, s) \quad (16)$$

$$m_{i2}(L, s) = m_{i2}(L, s+1) + \mathbf{lp}(L, s) \quad (17)$$

$$m_{j2}(L, s) = m_{j2}(L, s+1) + \mathbf{sp}(L, s) \quad (18)$$

(2) 若  $t<L$  且  $s=L$ , 则

$$p_2(t, L) = p_2(t+1, L) + \mathbf{p}(t, L) \quad (19)$$

$$m_{i2}(t, L) = m_{i2}(t+1, L) + \mathbf{tp}(t, L) \quad (20)$$

$$m_{j2}(t, L) = m_{j2}(t+1, L) + \mathbf{lp}(t, L) \quad (21)$$

(3) 若  $t<L$  且  $s<L$ , 则:

$$p_2(t, s) = p_2(t, s+1) + p_2(t+1, s) - p_2(t+1, s+1) + \mathbf{p}(t, s) \quad (22)$$

$$m_{i2}(t, s) = m_{i2}(t, s+1) + m_{i2}(t+1, s) - m_{i2}(t+1, s+1) + \mathbf{tp}(t, s) \quad (23)$$

$$m_{j2}(t, s) = m_{j2}(t, s+1) + m_{j2}(t+1, s) - m_{j2}(t+1, s+1) + \mathbf{sp}(t, s) \quad (24)$$

此准分递推算法的时间复杂度, 同二维 Otsu 近分递推法一样为  $O((L+1)^2)$ , 但比近分递推算法需要更多的时间, 因为在准分递推算法中,  $p_2(t, s)$ 、 $m_{i2}(t, s)$  和  $m_{j2}(t, s)$  还需要递推. 另外它的空间复杂度高: 除  $\mathbf{p}(i, j)$  外, 共需要  $6(L+1)^2 + 2$  个存储单元, 如此计算复杂度并不能满足实际分割的要求, 因此, 需要对准分递推算法进行改进. 为了得到改进的递推算法, 首先依据二维直方图见图 1, 证明如下的三个定理:

**定理 1** 在直方图中, 设

$$p_{13} = \sum_{i=0}^t \sum_{j=0}^L \mathbf{p}(i, j), p_{14} = \sum_{i=0}^L \sum_{j=0}^s \mathbf{p}(i, j),$$

$$p_{1L} = \sum_{i=0}^L \sum_{j=0}^L \mathbf{p}(i, j), \text{ 则:}$$

$$p_2(t, s) = p_{1L} - p_{13} - p_{14} + p_1(t, s) \quad (25)$$

**定理 2** 在直方图中, 设

$$m_{i13} = \sum_{i=0}^t \sum_{j=0}^L ip(i, j), m_{i14} = \sum_{i=0}^t \sum_{j=0}^s ip(i, j), \text{则:}$$

$$m_{i2}(t, s) = m_{Ti} - m_{i13} - m_{i14} + m_{i1}(t, s) \quad (26)$$

**定理 3** 在直方图中, 设

$$m_{j13} = \sum_{i=0}^t \sum_{j=0}^L jp(i, j), m_{j14} = \sum_{i=0}^t \sum_{j=0}^s jp(i, j), \text{则:}$$

$$m_{j2}(t, s) = m_{Tj} - m_{j13} - m_{j14} + m_{j1}(t, s) \quad (27)$$

以上三个定理证法相同, 下面以定理 2 为例加以证明.

**证明**

$$\begin{aligned} m_{Ti} &= \sum_{i=0}^L \sum_{j=0}^L ip(i, j) = \sum_{i=0}^L \left[ \sum_{j=0}^s ip(i, j) + \sum_{j=s+1}^L ip(i, j) \right] \\ &= \sum_{i=0}^L \sum_{j=0}^s ip(i, j) + \sum_{i=0}^L \sum_{j=s+1}^L ip(i, j) \\ &= \sum_{i=0}^L \sum_{j=0}^s ip(i, j) + \sum_{i=0}^t \sum_{j=s+1}^L ip(i, j) + \sum_{i=t+1}^L \sum_{j=s+1}^L ip(i, j) \\ &= m_{i14} + \sum_{i=0}^t \sum_{j=s+1}^L ip(i, j) + \sum_{i=0}^t \sum_{j=0}^s ip(i, j) \\ &\quad - \sum_{i=0}^t \sum_{j=0}^s ip(i, j) + m_{i2}(t, s) \\ &= m_{i14} + \sum_{i=0}^t \sum_{j=0}^L ip(i, j) - \sum_{i=0}^t \sum_{j=0}^s ip(i, j) + m_{i2}(t, s) \\ &= m_{i14} + m_{i13} - m_{i1}(t, s) + m_{i2}(t, s) \end{aligned}$$

所以  $m_{i2}(t, s) = m_{Ti} - m_{i13} - m_{i14} + m_{i1}(t, s)$ , 证毕.

然后  $p_2(t, s)$ 、 $m_{i2}(t, s)$  和  $m_{j2}(t, s)$  可以通过式(25)、(26)和式(27)来计算. 那么结合三个定理, 可以得到一种新型快速递推算法.

## 4 新型快速递推算法

在导出新型递推算法之前, 需找出需要递推的量. 因为对于同一幅图像,  $p_{LL} = 1$ 、 $m_{Ti}$  和  $m_{Tj}$  是固定的, 又依据式(5)、式(25)到式(27), 所以实际需要递推  $p_1(t, s)$ 、 $m_{i1}(t, s)$ 、 $m_{j1}(t, s)$ 、 $p_{13}$ 、 $p_{14}$ 、 $m_{i13}$ 、 $m_{i14}$ 、 $m_{j13}$  和  $m_{j14}$  就可得到最佳阈值向量. 而对于  $p_1(t, s)$ 、 $m_{i1}(t, s)$  和  $m_{j1}(t, s)$  三个量与  $p_{13}$ 、 $p_{14}$ 、 $m_{i13}$ 、 $m_{i14}$ 、 $m_{j13}$  和  $m_{j14}$  六个量递推的方式不同, 故分别进行讨论.

### 4.1 $p_1(t, s)$ 、 $m_{i1}(t, s)$ 和 $m_{j1}(t, s)$ 三个量的递推

(1) 分析一下直方图投影见图 2, 在此图中, 沿着主

对角线存在灰度概率分布, 而远离主对角线其概率为零. 从投影图以及大量的实验上可以得出二维直方图的两大特性:

第一、二维直方图中, 零元素较多, 因此在递推

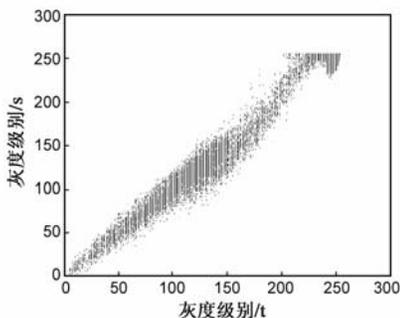


图2 二维直方图投影

$p_1(t, s)$  等与二维直方图有关量时, 必然存在着重复值. 即当  $p(i, j) = 0$  时,  $p_1(t, s)$  有可能是重复值, 而从式(2)可以看出,  $m_{i1}(t, s)$  和  $m_{j1}(t, s)$  也与二维直方图分布情况有关, 且与  $p_1(t, s)$  有相同的重复值分布, 那么可以通过  $p_1(t, s)$  来确定  $m_{i1}(t, s)$  和  $m_{j1}(t, s)$  的值是否需要计算获取. 另外, 当  $p(i, j) = 0$ ,  $m_{i1}(t, s)$  和  $m_{j1}(t, s)$  中的乘法无需运算.

(2) 只有主对角线的左上角和右下角这两个区域最有可能有重复值. 可以分  $j > i$  和  $j < i$  两种情况讨论, 具体递推算法如下:

赋初值:  $p_1(0, 0) = p(0, 0)$ ,

$$m_{i1}(0, 0) = 0, m_{j1}(0, 0) = 0.$$

分三种情况进行递推:

(a) 在  $t = 0$  且  $s > 0$  时,  $m_{i1}(0, s) = 0$ .

if  $p(i, j) = 0$ ,  $p_1(0, s)$  和  $m_{j1}(0, s)$  保持不变.

else 计算式(9)和式(10).

(b) 在  $s = 0$  且  $t > 0$  时,  $m_{j1}(t, 0) = 0$ .

if  $p(i, j) = 0$ ,  $p_1(t, 0) = p_1(t - 1, 0)$ ,

$$m_{i1}(t, 0) = m_{i1}(t - 1, 0).$$

else 计算式(11)和式(12).

(c) 在  $t > 0$  且  $s > 0$  时,

if  $p(i, j) = 0$ , 如果当  $j > i$  且  $p_1(t, s) = p_1(t, s - 1)$  时, 则  $m_{i1}(t, s)$  和  $m_{j1}(t, s)$  保持不变. 如果当  $j < i$  且  $p_1(t, s) = p_1(t - 1, s)$  时, 则  $m_{i1}(t, s) = m_{i1}(t - 1, s)$ ,  $m_{j1}(t, s) = m_{j1}(t - 1, s)$ . 否则:

$$p_1(t, s) = p_1(t, s - 1) + p_1(t - 1, s) - p_1(t - 1, s - 1) \quad (28)$$

$$m_{i1}(t, s) = m_{i1}(t, s - 1) + m_{i1}(t - 1, s) - m_{i1}(t - 1, s - 1) \quad (29)$$

$$m_{j1}(t, s) = m_{j1}(t, s - 1) + m_{j1}(t - 1, s) - m_{j1}(t - 1, s - 1) \quad (30)$$

else 计算式(13)、(14)和式(15).

最后由式(5)和式(6)逐次计算逐次比较求出最佳阈值向量.

从上面的算法可以看出: 虽然要递推三个量, 但三个量递推次数都大大小于  $(L + 1)^2$ , 由于利用了直方图的分布特性, 其中乘法运算次数就是直方图中非零元素的个数, 式(9)至式(15)和式(28)至式(30)的计算次数就是  $p_1(t, s)$  的非重复值的个数, 一般直方图中非零元素的个数仅仅是  $(L + 1)^2$  的五分之一以下, 甚至达到十分之一以下, 而乘法运算是 Otsu 递推算法中是最耗时的; 而非重复值的个数一般是  $(L + 1)^2$  的三分之一以下, 这样节约了时间. 而空间复杂度为: 在这三个量中每个量需要  $L + 1$  个临时存储单元存储  $s$  从 0 到  $L$  的  $x(t - 1, s)$  每个值, 需要两个临时变量分别保存  $x(t, s)$

和  $x(t-1, s-1)$ , 加上逐次计算逐次比较需要两个临时变量, 每个变量都需一个存储单元, 总共需要  $3(L+1)+8$  个存储单元。

#### 4.2 $p_{13}$ 、 $p_{14}$ 、 $m_{i13}$ 、 $m_{i14}$ 、 $m_{j13}$ 和 $m_{j14}$ 六个量的递推

这六个量主要用来计算  $p_2(t, s)$ 、 $m_{i2}(t, s)$  和  $m_{j2}(t, s)$  的值, 与上文描述的准分一般递推算法比较, 初看起来, 似乎要递推的量多了, 计算复杂度更高了; 但实际上, 用这六个量来替换前三个量, 计算复杂度大幅下降, 原因是前三个量都需要递推  $(L+1)^2$ , 而这六个量都仅仅需要递推  $(L+1)$  次。而且都采用上面介绍的方法来减少乘法运算次数和有关算式的计算次数。下面针对六个量不同的情况采用不同的计算方式, 描述如下:

对于  $p_{13}$ 、 $m_{i13}$  和  $m_{j13}$  三个量都是在  $s=L$  时计算  $t$  从 0 到  $L$  对应直方图中的每一列值, 总共有  $L+1$  列, 因为在上文递推算法中可以同时计算目标函数式(5), 所以这三个量需要在逐次计算逐次比较的递推算法之前进行并保存在三个一维数组中, 以备读取; 此外, 由  $m_{i13}$  和  $m_{j13}$  递推的结果可以直接得到  $m_{Ti}$  和  $m_{Tj}$ , 不需要另外再计算, 这样也降低了计算复杂度。对于  $p_{14}$ 、 $m_{i14}$  和  $m_{j14}$  三个量是在  $t=L$  时计算  $s$  从 0 到  $L$  对应直方图中的每一行值, 总共有  $L+1$  行, 因为在逐次计算逐次比较的递推算法中是从第一列到最后一列, 从第一行到最后一行的顺序进行递推, 所以, 这三个量无需先递推保存在数组中, 直接在此算法中进行递推。

从上面的分析可以看出: 这六个量的时间复杂度为最大为  $O(L+1)$ , 而空间复杂度为: 前三个量的每一个量需要  $L+1$  个存储单元, 共有  $3(L+1)$  个; 后三个量的每一个量只需要一个存储单元, 共 3 个, 两者加起来

为  $3(L+1)+3$  个存储单元。那么本文提出新型递推算法总的空间复杂度为  $6(L+1)+11$  个存储单元。

## 5 实验结果及分析

为了验证本文算法的有效性, 用其分割图像, 限于篇幅本文只选用了二幅图片作为示例说明见图 3(a) 和图 4(a), 前者叠加上椒盐噪声, 其强度为 0.01, 后者叠加上均值为 0, 方差为 0.01 的高斯噪声, 两幅图像的直方图见图 3(b) 和图 4(b)。算法采用 MATLAB 6.5 语言实现, 所有实验都在 AMD Athlon 64 X2 主频为 2.7G 的 CPU 和内存为 2GB DDR RAM 的机器上进行, 操作系统使用 Windows XP。

(1) 分割性能对比实验。在本实验中, 将本文方法的分割结果与一维 Otsu 法<sup>[2]</sup>、二维 Otsu 近分法和二维 Otsu 斜分法<sup>[8]</sup>分割的结果进行比较, 其中近分法和斜分法以及本文准分法的邻域模板都采用改进模板, 四种方法获取的阈值或阈值向量见表 3。两幅图片的四种方法分割结果如图 3 和图 4(e)、(f)、(g) 和 (h) 所示。从分割结果图的视觉效果上可以看出, 本文方法明显优于其它三种方法, 在这两幅图片的分割实验中, 其他三种方法都没能很好地将目标对象分割出来, 而本文的方法将目标对象从背景中清晰地分割出来了见图 3(e) 和图 4(e) 的四个方块和五个瓶子。

(2) 邻域模板对图像分割影响实验。在此实验中, 列举了三种邻域模对直方图的影响, 这三种模板是: 常见的均值 8-邻域模板(8-Neighborhood Mask, 8-NM)、均值 4-角域模板(4-Angle Mask, 4-AM)和本文提出的改进模板(ImprovedMask, IM)。表 1 是不同邻域构建的直方图

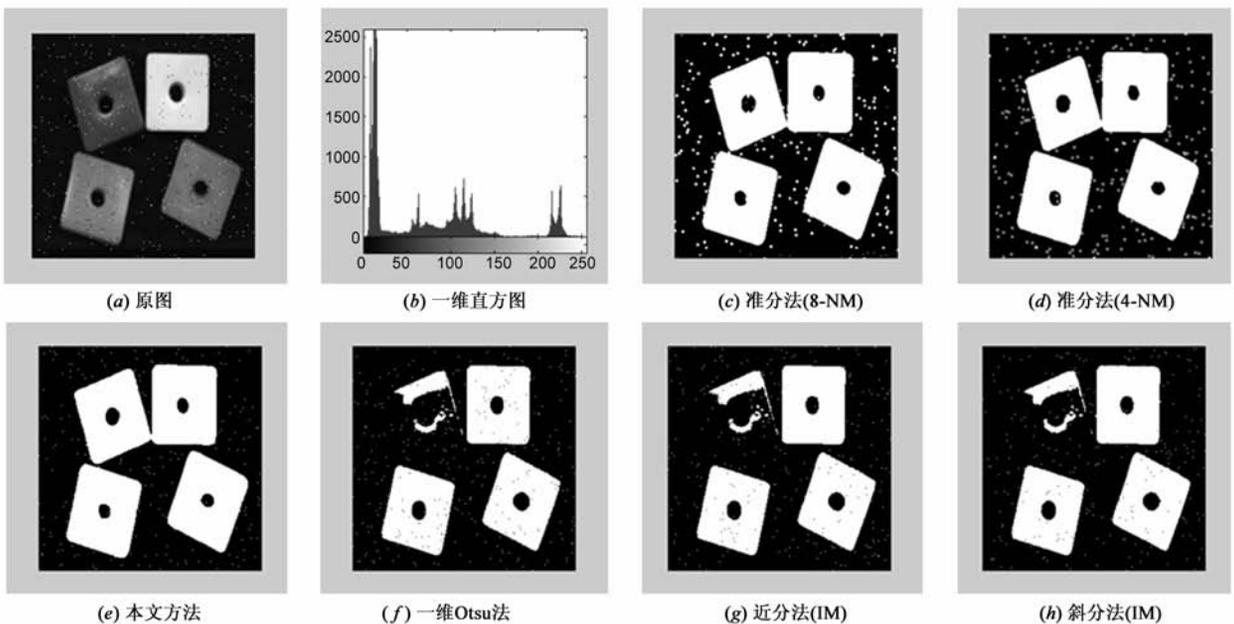


图3 Blocks图片及其分割结果

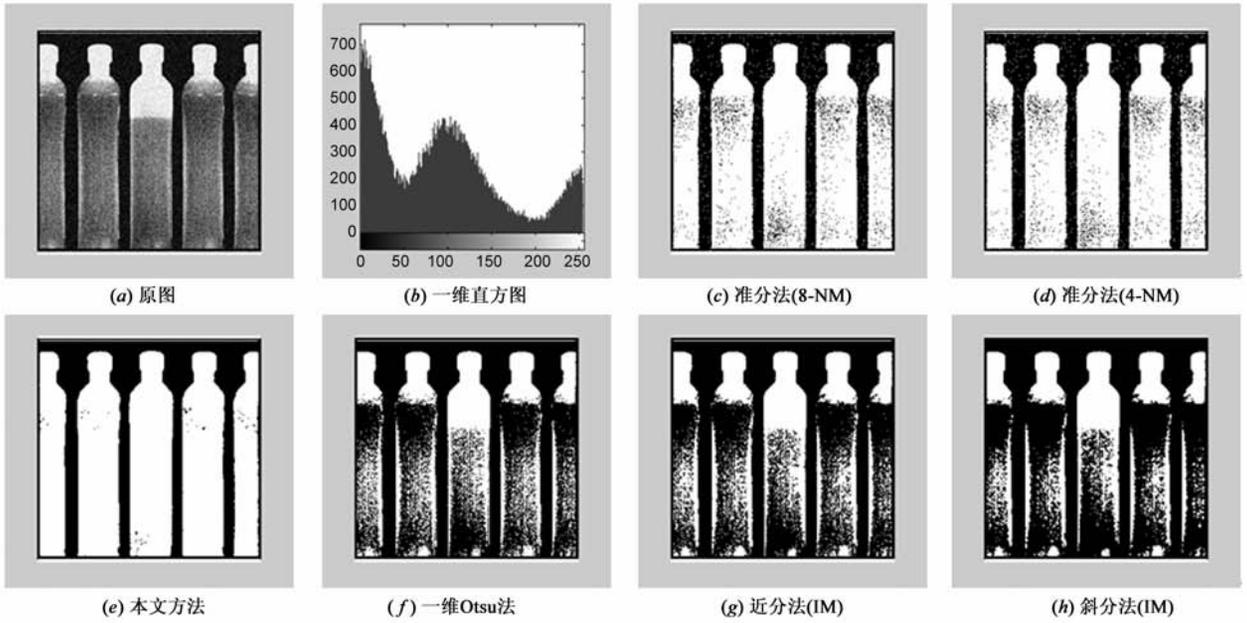


图4 Bottles图片及其分割结果

对应的非零数(乘法运算次数)和非重复值次数(有关算式的计算次数).从此表可以看出:(a)一旦邻域模板确定,其二维直方图的灰度概率分布就确定了,与采用的二维阈值方法无关;(b)非零数越多,乘法运算次数越多,对应的非重复次数越多,计算相关表达式的次数越多,时间花费越多;(c)在三种邻域模板中,本文提出的改进模板非零数最少,4-角域模板非零数最多,这说明改进模板乘法运算的次数最少,如表中两幅图片非零数的平均(7333)是 65536(256 × 256)的 11%,非重复次数的平均(18344)是 65536 的 28%,利用此特性可以大大降低计算复杂度;(d)待分割的图像有无噪声对直方图的非零数会有影响见表 1,有噪声时非零数增加.表 3 的最后三列是不同邻域模板在准分法下获取的阈值向量.从此表可以看出:邻域模板不同,阈值向量不同,会产生不同的分割结果.用准分法在三种模板下对两幅图片进行分割,其分割结果如图 3 和图 4(c)、(d)和(e)所示,从这些分割结果看,本文提出的改进模板在抗噪性和分割效果上都优于常用的 8-邻域和 4-角域均值模板.综上所述,本文提出的改进模板提高了分割效果和分割速度,优于常用的均值邻域模板.

(3)运行时间对比实验.表 2 是二维 Otsu 准分法在穷举搜索、递推搜索和本文提出的新型搜索方法以及二维 Otsu 近分递推算法和二维 Otsu 斜分递推算法<sup>[8]</sup>的运行时间.其中,本文的穷举法和递推法对传统的穷举方法和递推算法进行了改进,尽量减少穷举和递推时间.从此表可以看出:穷举方法耗时最多,平均约为 147 秒,这是因为它的时间复杂度是  $O((L + 1)^4)$ ;在文献[8]斜分递推算法中,虽然目标函数计算次数为  $2(L +$

1),次数少,但在其递推算法中,求解  $p_1(t, s)$ 、 $m_{i1}(t, s)$  和  $m_{j1}(t, s)$  并没有消去大量重复无效的表达式计算,包括较为耗时的乘法运算,仍然需要  $(L + 1)^2$  次,而且还要耗时寻找斜线上的点;准分递推法和近分递推法耗时原因上文已经交代过,在这里不再赘述.本文的快速算法:(a)建立在一般递推算法的基础之上,不仅消去了大量的重复求和、乘法等运算,而且使得计算  $p_1(t, s)$ 、 $m_{i1}(t, s)$  和  $m_{j1}(t, s)$  表达式的次数和其乘法运算次数都大为降低;(b)使用三个定理使  $p_2(t, s)$ 、 $m_{i2}(t, s)$  和  $m_{j2}(t, s)$  的计算复杂度大幅度下降;综上所述两点,因此新算法的运行时间较少,与穷举算法相比减少 4 个数量级,与文献[8]的斜分递推算法相近.另外,本文提出的新型搜索算法不仅时间复杂度大为降低,而且空间复杂度也显著下降,约为准分一般递推算法的  $L + 1$  分之一,总之,本文提出的快速算法是有效的.

表 1 不同邻域直方图对应的非零数和有关算式所计算的次数

测试图像	噪声	8-邻域模板		改进模板		4-角域模板	
		次数	非 0 数	次数	非 0 数	次数	非 0 数
Blocks. jpg	有	14215	4044	12810	2657	19244	4600
Blocks. jpg	无	3794	2962	3205	2330	5442	3783
Bottles. bmp	有	29355	16307	38996	17346	39657	19401
Bottles. bmp	无	18403	7915	18366	6999	27281	9544
平均次数		16442	7807	18344	7333	22906	9332

表 2 不同搜索算法运行时间对比/秒

测试图像	准分穷举法	近分递推法	斜分递推法	本文的方法	准分递推法
Blocks. jpg	132.282	0.1314	0.0223	0.0227	0.2091
Bottles. bmp	161.687	0.1332	0.0224	0.0241	0.2117
平均时间	146.985	0.1323	0.0224	0.0234	0.2104

表 3 四种方法及本文方法在不同的模板下所获取的阈值向量或阈值

测试图像	一维	改进模板(不同的二维方法)			准分法(不同的模板)	
	Otsu	近分法	斜分法	准分法	8-邻域	4-角域
Blocks.jpg	83	(82, 144)	164	(130, 33)	(129, 38)	(131, 38)
Bottles.bmp	109	(113, 122)	229	(162, 51)	(52, 150)	(52, 152)

## 6 结论

(1)传统的二维 Otsu 法由于对二维直方图采用近似处理,再加之其他原因,分割欠佳.由此本文提出了二维直方图准分的 Otsu 阈值分割方法:(a)准确选择构建二维直方图的邻域模板;(b)对二维直方图主对角线上的目标和背景两区域的 Otsu 公式中对应量准确取值,此方法是有效的,其分割结果更准确和抗噪性更强.

(2)本文通过分析二维直方图得到其特性和三个定理,利用此特性和定理提出了一种新型的快速算法大大降低了计算复杂度,能较好运用到实时应用的场合,此算法也可以推广到其它二维阈值法中.

## 参考文献

- [1] Sczgin M, Sankur B. Survey over image thresholding techniques and quantitative performance evaluation[J]. Journal of Electronic Imaging, 2004, 13(1): 146 - 168.
- [2] Otsu N. A threshold selection method from gray-level histogram [J]. IEEE Transaction on Systems, Man and Cybernetics, 1979, 9(1): 62 - 66.
- [3] 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.  
Liu Jian-zhuang, Li Wen-qing. Automatic thresholding using the Otsu algorithm based on the two-dimensional gray image[J]. Acta Automatica Sinica, 1993, 19(1): 101 - 105. (in Chinese)
- [4] Gong J, Li L Y, Chen W N. Fast recursive algorithm for two-dimensional thresholding[J]. Pattern Recognition, 1998, 31(3): 295 - 300.
- [5] 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)
- [6] 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.  
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. (in Chinese)
- [7] 范九伦, 赵凤. 灰度图像的二维 Otsu 曲线阈值分割法[J]. 电子学报, 2007, 35(4): 751 - 755.  
Fan Jiu-lun, Zhao Feng. Two-dimensional Otsu's curve thresh-

- olding segmentation method for gray-level images [J]. Acta Electronica Sinica, 2007, 35(4): 751 - 755. (in Chinese)
- [8] 吴一全, 潘 ■, 吴文怡. 二维直方图区域斜分阈值分割及快速递推算法[J]. 通信学报, 2008, 29(4): 77 - 83.  
Wu Yi-quan, Pan Zhe, Wu Wen-yi. Image thresholding based on two-dimensional histogram oblique segmentation and its fast recurring algorithm [J]. Journal on Communications, 2008, 29(4): 77 - 83. (in Chinese)
  - [9] 吴一全, 张金矿. 二维直方图  $\theta$ -划分最大平均离差阈值分割算法[J]. 自动化学报, 2010, 36(5): 634 - 643.  
Wu Yi-quan, Zhang Jin-kuang. Image thresholding based on two-dimensional histogram  $\theta$ -division and maximum between-cluster deviation criterion [J]. Acta Automatica Sinica, 2010, 36(5): 634 - 643. (in Chinese)
  - [10] 张云飞, 张晔. 利用二维熵自动确定图像分割的阈值 [J]. 哈尔滨工程大学学报, 2006, 27(3): 353 - 356.  
Zhang Yun-fei, Zhang Ye. Automatic threshold of image segmentation using 2-D entropy [J]. Journal of Harbin Engineering University, 2006, 27(3): 353 - 356. (in Chinese)
  - [11] 姚敏. 数字图像处理 [M]. 北京: 机械工业出版社, 2008.  
Yao Min. Digital Image Processing [M]. Beijing, China: Machine Press, 2008. (in Chinese)
  - [12] 陈琪, 熊博莅, 陆军, 等. 改进的二维 Otsu 图像分割方法及其快速实现 [J]. 电子与信息学报, 2010, 32(5): 1100 - 1104.  
Chen Qi, Xiong Bo-li, Lu Jun et al. Improved two-dimensional Otsu image segmentation method and fast recursive realization [J]. Journal of Electronics & Information Technology, 2010, 32(5): 1100 - 1104. (in Chinese)

## 作者简介



张新明 男, 1963 年出生, 湖北省孝感市人, 硕士, 副教授, 硕士生导师, 主要研究方向: 数字图像处理、智能优化算法和模式识别等。  
E-mail: xinmingzhang@126.com



孙印杰 男, 1964 年出生, 河南省新野县人, 硕士, 教授级高级工程师, 硕士生导师, 主要研究方向: 图像通信安全和多媒体技术及其应用等。  
E-mail: sunyinjie@126.com

郑延斌 男, 1964 年出生, 河南省内乡县人, 博士, 教授, 硕士生导师, 主要研究方向: 图形图像技术、虚拟现实和人工智能等。