

改进的 2 维 Otsu 法及混沌粒子群递推的阈值分割

吴一全 张金矿

(南京航空航天大学 信息科学与技术学院, 南京 210016)

摘要 鉴于现常用的灰度级-平均灰度级 2 维直方图区域划分将部分目标和背景点错分成边缘和噪声点这一不足, 为此提出了一种基于灰度级-梯度 2 维直方图的 Otsu 阈值选取新方法, 利用混沌粒子群优化算法来寻找分割阈值, 并提出在迭代过程中, 采用递推方法来大大减少适应度函数的重复计算。实验结果表明, 与最近提出的基于灰度级-平均灰度级 2 维直方图 Otsu 法及粒子群的快速图像分割方法相比, 该新方法由于尽可能地考虑了所有目标点和背景点, 从而使分割后的图像区域内部均匀、边界形状准确、特征细节清晰, 同时运行时间几乎不到现有算法的 1/3, 而且粒子群处理的收敛精度得到了进一步提高。

关键词 图像分割 阈值选取 2 维直方图 Otsu 法 混沌粒子群 递推

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2009)09-1843-07

Thresholding Based on Improved 2D Otsu Method and Chaotic Particle Swarm Optimization

WU Yi-quan, ZHANG Jin-kuang

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

Abstract In view of the shortage of regional division of the commonly used gray level-average gray level two-dimensional histogram, which some object and background inner points are wrongly divided as edge and noise points, an improved Otsu threshold selection method based on gray level-gradient two-dimensional histogram is proposed in this paper. The chaotic particle swarm algorithm is used to search for the best threshold. The repeat computations of the fitness function in iteration are reduced significantly using recursion. Compared with fast image segmentation algorithm based on gray level-average gray level 2-D Otsu method and particle swarm optimization, the experimental results show that the algorithm proposed in this paper not only considers all the object and background inner points and achieves a good segmentation quality in uniform regions, accurate borders and clear details of features, but also the running time is reduced to only 1/3 of that of the existing algorithm. At the same time the convergence property of particle swarm algorithm is further improved.

Keywords image segmentation, threshold selection, two-dimensional histogram, Otsu thresholding, chaotic particle swarm, recursion

1 引言

阈值分割是应用广泛、处理有效的图像分割方法, 其关键是选取合适的阈值, 以获得最佳的分割效果。在诸多的阈值选取方法中^[1-2], 由 Otsu 提出的

1 维最大类间方差法^[3], 以其分割效果较好、简单有效而颇受关注, 且应用最为广泛。1993 年, 刘健庄等人借助 Abutaleb 等人提出的 2 维直方图思想^[4], 将 Otsu 法从 1 维拓展到了 2 维^[5], 由于其利用了图像局部空间信息, 当图像信噪比降低时, 其图像分割效果较 1 维方法有明显改善, 但同时也导致搜索次

基金项目: 国家自然科学基金项目(60872065)

收稿日期: 2008-01-14; 改回日期: 2008-05-19

第一作者简介: 吴一全(1963 ~), 男, 教授。1998 年于南京航空航天大学信息科学与技术学院获信息与通信工程学科博士学位。主要研究方向为图像处理与识别、视频处理与通信、信号处理等。E-mail: gumption_s@yahoo.com.cn

数和运算量大幅增加。为此,人们从不同的观点出发,提出了 2 维 Otsu 法的递推算法、量化直方图算法以及遗传算法等快速算法^[6-9],不同程度地节省了计算时间。而基于群体智能理论的粒子群优化算法(PSO)^[10]则是采用位置-速度模型的动态调整搜索策略,从而避免了交叉和变异操作,是一种更高效的搜索算法。最近,唐英干、何庆元等人就分别将基本粒子群算法和协同带压缩因子粒子群算法用于 2 维 Otsu 阈值分割中^[11-12],进一步减少了计算时间。

但是,迄今为止,上述 2 维 Otsu 阈值选取方法及其快速算法存在下面两个问题:(1)上述算法几乎都是采用分别与灰度级、邻域平均灰度级两坐标轴平行的十字线先将 2 维直方图划分成 4 个矩形区域,然后计算时仅考虑沿对角线的 2 个矩形区域。由于采取了一定的近似,例如忽略了位于阈值向量附近的像素灰度级与其邻域平均灰度级相近的一半区域处的概率分布,但这一假设与实际不符,因此得到的分割结果不够准确;(2)上述用于寻求 2 维 Otsu 阈值的基本粒子群算法^[11]易于陷入局部极值束缚,而协同带压缩因子粒子群算法^[12]虽提高了精度,但难以保证收敛到全局最优解。更引起人们关注的缺陷是这两种算法在迭代过程中,当计算适应度函数时都存在大量的重复计算。

鉴于上述原因,很有必要对基于灰度级-平均灰度级 2 维直方图 Otsu 法及粒子群的阈值分割方法进行改进。本文首先提出了基于灰度级-梯度 2 维直方图的 Otsu 阈值选取方法,以提高分割结果的准确性;然后利用混沌粒子群算法寻找最佳阈值,由混沌的遍历性逃脱局部极值,以提高粒子群算法的收敛精度;最后提出在迭代过程中采用递推方法,以大大减少适应度函数的重复计算。在实验结果和分析中,还对基于常用的灰度级-平均灰度级 2 维直方图和基于改进的 2 维直方图的 Otsu 阈值分割粒子群算法的分割结果、运行时间及收敛精度进行了分析与比较。

2 基于灰度级-梯度 2 维直方图 Otsu 阈值选取

设 $H \times W$ 大小图像的灰度级取 $0, 1, \dots, L-1$, 像素点 (m, n) 的邻域平均灰度级 $g(m, n)$ 定义为

$$g(m, n) = \frac{1}{W} \sum_{(m_1, n_1) \in D} f(m_1, n_1) \quad (1)$$

其中, W 为邻域 D 中的像素点数, D 一般取像素点 (m, n) 的 4-邻域或 8-邻域,可分别由下述模板表征:

$$\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ 或 } \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2.1 常用的灰度级-平均灰度级 2 维直方图

现常用 2 维直方图的横、纵坐标分别采用像素的灰度级 $f(m, n)$ 和邻域平均灰度级 $g(m, n)$ 。它反映了图像中 $(f(m, n), g(m, n))$ 对发生的联合概率。Lena 图(图 1(a))的灰度级-平均灰度级 2 维直方图如图 1(b)所示。目前 2 维直方图阈值选取方法几乎都采纳这种直方图,并在阈值向量 (t, s) 处采用十字线来将图 1(b)所示的 2 维直方图分割成如图 1(c)所示的 4 个矩形区域。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 3 则表示边缘和噪声。计算时仅考虑区域 0 和 1,而假设区域 2 和 3 的 $(f(m, n), g(m, n))$ 发生的概率近似为零。但该区域划分方法存在着明显的缺点,即计算最佳阈值向量时,由于忽略了位于阈值向量附近、且靠近对角线处(像素灰度级与其邻域平均灰度级相近,应属于目标或背景内点)约一半区域(区域 2 的右下角和区域 3 的左上角)的概率分布,因此会将一部分划分成边缘和噪声点的目标内点和背景内点删去。由图 1(b)可见,由于这与实际的 2 维直方图不符,因此基于这种直方图及其区域划分方式计算的最佳阈值必然会出现偏差,分割结果就不够准确。

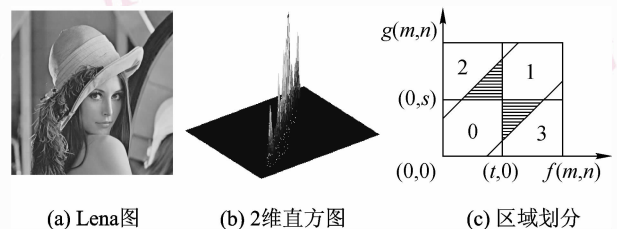


图 1 现常用的灰度级-平均灰度级 2 维直方图

Fig. 1 Commonly used gray level-average gray level two-dimensional histogram

2.2 基于灰度级-梯度的 2 维 Otsu 阈值选取

首先提出改进的 2 维直方图及其区域划分方法。2 维直方图的横坐标依然采用像素的灰度级 $f(m, n)$,纵坐标改用像素的灰度梯度,即灰度级与邻域平均灰度级之差的绝对值 $|f(m, n) - g(m, n)|$ 。当采用 4-邻域时,邻域灰度梯度计算相当于拉普拉

斯算子。如果用 $r(i, j)$ 表示相应的(灰度级 $f=i$, 梯度 $|f-g|=j$) 对出现的频数 ($0 \leq r(i, j) \leq H \times W$), 则可将 $p(i, j)$ 定义为图像中 $(f, |f-g|)$ 对发生的联合概率:

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

即 $\{p(i, j)\}$ 为图像的灰度级-梯度 2 维直方图。利用阈值向量 (t, s) 以及分别与灰度级、梯度两坐标轴平行的十字线, 将 2 维直方图分割成图 2(a) 所示的 4 个矩形区域: 其中区域 0 的像素点的灰度级较小, 其邻域的灰度梯度也较小, 对应于目标点; 区域 1 的像素点的灰度级较大, 其邻域的灰度梯度较小, 对应于背景点; 区域 2 和区域 3 的邻域灰度梯度都较大, 分别表示背景和目标的边缘和噪声。

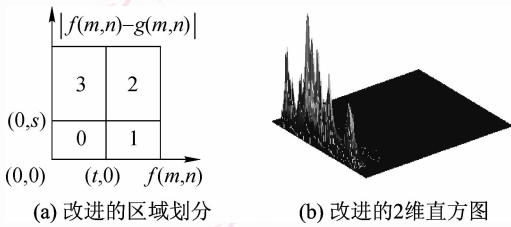


图 2 灰度级-梯度 2 维直方图

Fig. 2 Gray level-gradient two-dimensional histogram

图 2(b) 给出了 Lena 图像的灰度级-梯度 2 维直方图。从图中可看出, 概率分布高峰主要集中在区域 0 和区域 1, 这是由于在整幅图像像素中, 目标和背景的内点区域灰度级比较均匀、梯度较小, 其所占比例很大; 而位于区域 2 和区域 3 的边缘点和噪声点, 则梯度较大, 其所占比例相对很小。在如今常用的灰度级-平均灰度级 2 维直方图中, 由于忽略了位于阈值向量附近, 且靠近对角线处约一半区域的概率分布, 因此会去掉被错误地划分成边缘和噪声点的那部分目标和背景的内点(图 1 中区域 2 的右下角和区域 3 的左上角), 由于这些像素的灰度级与其邻域的平均灰度级相近, 即邻域灰度梯度较小, 因而在图 2 的灰度级-梯度 2 维直方图中, 被正确地划分为理应属于目标和背景的区域。可见这种灰度级-梯度 2 维直方图及其区域划分方式的最大优点是区域 0 和区域 1 能尽可能地包含所有目标点和背景点。

下面将改进的 2 维直方图及区域划分方式运用于 2 维 Otsu 阈值选取, 现导出具体的算法公式。

由图 2 知, 区域 0, 1 发生的概率分别为

$$\omega_0(t, s) = \sum_{i=0}^t \sum_{j=0}^s p(i, j) \quad (3)$$

$$\begin{aligned} \omega_1(t, s) &= \sum_{i=t+1}^{L-1} \sum_{j=0}^s p(i, j) = \sum_{i=0}^{L-1} \sum_{j=0}^s p(i, j) - \sum_{i=0}^t \sum_{j=0}^s p(i, j) \\ &= \omega_0(L-1, s) - \omega_0(t, s) \end{aligned} \quad (4)$$

区域 0 和区域 1 的均值向量分别为

$$\begin{aligned} \mu_0(t, s) &= (\mu_{0,i}, \mu_{0,j}) = \left[\sum_{i=0}^t \sum_{j=0}^s \frac{ip(i, j)}{\omega_0(t, s)}, \sum_{i=0}^t \sum_{j=0}^s \frac{jp(i, j)}{\omega_0(t, s)} \right] \\ &= \left[\frac{\mu_i(t, s)}{\omega_0(t, s)}, \frac{\mu_j(t, s)}{\omega_0(t, s)} \right] \end{aligned} \quad (5)$$

其中,

$$\mu_i(t, s) = \sum_{i=0}^t \sum_{j=0}^s ip(i, j), \mu_j(t, s) = \sum_{i=0}^t \sum_{j=0}^s jp(i, j) \quad (6)$$

$$\begin{aligned} \mu_1(t, s) &= (\mu_{1,i}, \mu_{1,j}) = \left[\sum_{i=t+1}^{L-1} \sum_{j=0}^s \frac{ip(i, j)}{\omega_1(t, s)}, \sum_{i=t+1}^{L-1} \sum_{j=0}^s \frac{jp(i, j)}{\omega_1(t, s)} \right] \\ &= \left[\frac{\mu_i(L-1, s) - \mu_i(t, s)}{\omega_0(L-1, s) - \omega_0(t, s)}, \frac{\mu_j(L-1, s) - \mu_j(t, s)}{\omega_0(L-1, s) - \omega_0(t, s)} \right] \end{aligned} \quad (7)$$

2 维直方图总的均值向量为

$$\mu_{total} = (\mu_{Ti}, \mu_{Tj}) = \left[\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} ip(i, j), \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} jp(i, j) \right] \quad (8)$$

类间协方差矩阵定义为

$$\begin{aligned} \sigma(t, s) &= \sum_{k=0}^1 \omega_k(t, s) [\mu_k(t, s) - \mu_{total}]^T \\ &\quad [\mu_k(t, s) - \mu_{total}] \end{aligned} \quad (9)$$

以矩阵 $\sigma(t, s)$ 的迹 $\text{tr}[\sigma(t, s)]$ 作为类间离散度的度量, 使其达到最大即可求得以下最佳阈值向量:

$$(t^*, s^*) = \text{Arg} \max_{t, s=0, 1, \dots, L-2} \{\text{tr}[\sigma(t, s)]\} \quad (10)$$

可见, 上述给出的基于改进的 2 维直方图的 Otsu 阈值选取公式与如今常用的基于 2 维直方图的 Otsu 阈值选取公式不同, 由于它不涉及区域 2 和区域 3, 因此无需假设区域 2 和区域 3 的概率近似为零。

3 改进的 2 维 Otsu 阈值选取的混沌粒子群递推算法

3.1 基本粒子群优化算法 (PSO)

设在 n 维解空间中,第 o 个粒子的位置为 $\mathbf{x}_o = (x_{o,1}, x_{o,2}, \dots, x_{o,n})$ 和速度为 $\mathbf{v}_o = (v_{o,1}, v_{o,2}, \dots, v_{o,n})$,前者表示问题的解,其对应的目标函数值 F_o 可作为评价该粒子优劣程度的适应度;后者表示粒子从当前位置移动到下一个位置的速度大小。求解时首先对粒子群的位置和速度进行初始化,然后通过迭代方式在解空间中寻找最优解。假设在第 k 次迭代时,第 o 个粒子的最优解为 $\mathbf{p}_o^{\text{best}}(k)$,称为个体极值,相应的个体历史最优适应度为 F_o^{best} ,整个粒子群的最优解为 $\mathbf{g}^{\text{best}}(k)$,称为全局极值。而在 $k+1$ 次迭代时刻,则先按下式更新速度:

$$\mathbf{v}_o(k+1) = w\mathbf{v}_o(k) + c_1r_1[\mathbf{p}_o^{\text{best}}(k) - \mathbf{x}_o(k)] + c_2r_2[\mathbf{g}^{\text{best}}(k) - \mathbf{x}_o(k)] \quad (11)$$

然后以速度 $\mathbf{v}_o(k+1)$ 移动到下一个位置,即

$$\mathbf{x}_o(k+1) = \mathbf{x}_o(k) + \mathbf{v}_o(k+1) \quad (12)$$

式中, k 表示当前迭代次数;通常学习因子 $c_1 = c_2 = 2$; r_1, r_2 是均匀分布在 $(0, 1)$ 上的随机数;设惯性因子 w 随迭代次数增加而线性递减:

$$w = w_{\max} - k \times \frac{w_{\max} - w_{\min}}{K} \quad (13)$$

式中, w_{\max} 和 w_{\min} 分别表示最大和最小惯性因子, K 表示总迭代次数。在迭代过程中,第 o 个粒子的速率限制为 $\mathbf{v}_o \in [\mathbf{v}^{\min}, \mathbf{v}^{\max}]$,并将粒子位置限制在允许范围之内,最后输出的 \mathbf{g}^{best} 即为全局最优解。

3.2 混沌优化算法 (CPSO)

混沌是自然界广泛存在的一种非线性现象,其具有随机性、遍历性、初始条件敏感性等特点。混沌优化算法^[13]的基本思想是先把混沌变量从混沌空间映射到解空间,再利用混沌变量的特点进行搜索。

在粒子群寻优的过程中,若某个粒子停滞,则进入混沌搜索。搜索时,首先产生 n 维随机初始向量

$$\mathbf{y}_0 = (y_{0,1}, y_{0,2}, \dots, y_{0,n}) \quad y_{0,d} \in [0, 1]$$

再用 \mathbf{y}_0 作为迭代初始值,并根据 Logistic 方程

$$y_{m+1,d} = uy_{m,d}(1 - y_{m,d}) \quad d = 1, 2, \dots, n \quad (14)$$

进行迭代来得到第 m 次迭代的混沌序列 $\mathbf{y}_m (m = 1, 2, \dots, M)$, M 为最大混沌迭代次数,由此即可产生局

部最优解周围的多个邻域点;然后通过载波的方式,按下式

$$\tilde{\mathbf{y}}_{m,d} = x_{o,d} + R_{o,d}(2y_{m,d} - 1) \quad (15)$$

将混沌序列 \mathbf{y}_m 映射到解空间,即将第 m 次迭代的混沌序列 \mathbf{y}_m 映射到以第 o 个粒子的位置 \mathbf{x}_o 为中心,以第 o 个粒子映射半径向量 \mathbf{R}_o 为半径的区域上;最后计算当前位置的适应度函数 $F(\tilde{\mathbf{y}}_m)$,并更新混沌迭代过程的第 o 个粒子的历史最优适应度 $\tilde{F}_o^{\text{best}}$ 和第 o 个粒子的历史最优位置 $\tilde{\mathbf{P}}_o^{\text{best}}$,若 $\tilde{F}_o^{\text{best}}$ 优于 F_o^{best} ,则用第 o 个粒子的最优位置 $\tilde{\mathbf{P}}_o^{\text{best}}$ 和速度 $\tilde{\mathbf{v}}_o$ 替换原来的位置和速度,其中

$$\tilde{\mathbf{v}}_o = \frac{\tilde{\mathbf{P}}_o^{\text{best}} - \mathbf{x}_o}{\|\tilde{\mathbf{P}}_o^{\text{best}} - \mathbf{x}_o\|} \quad (16)$$

3.3 改进的 2 维 Otsu 阈值选取的混沌粒子群递推算法 (RCPSO)

为降低 2 维空间的搜索代价,在进行基于改进的 2 维直方图的 Otsu 阈值选取时,利用混沌粒子群算法,并以 $\text{tr}[\boldsymbol{\sigma}(t, s)]$ 作为适应度函数,具体过程如下:

(1) 粒子群初始化。设粒子群规模为 N ,并随机生成 N 个 2 维粒子,将其中由像素灰度级及其邻域梯度构成的有序对作为粒子的位置,其分别在像素灰度级、梯度范围内均匀分布,速度在 $[\mathbf{v}^{\min}, \mathbf{v}^{\max}]$ 内随机产生,设置的参数分别为最大迭代次数 K ,最大惯性因子 w_{\max} 和最小惯性因子 w_{\min} ,同时令 $k = 0$;

(2) 计算粒子的适应度。根据粒子的位置,利用式(3)~式(9)计算各粒子的适应度 $F_o(k)$ 和更新粒子的历史最优适应度 F_o^{best} 、个体最优位置 $\mathbf{p}_o^{\text{best}}$ 、全局最优适应度 F^{best} 和全局位置极值 \mathbf{g}^{best} ;

(3) 采用如下判定准则判断粒子是否停滞

$$\Delta F_o = \frac{F_o - F_o^{\text{best}}}{F_o} \quad (17)$$

分别取 δ 和 n_{chao} 为预先设定的门限和整数,若迭代过程中连续出现 $\Delta F_o < \delta$ 超过 n_{chao} 次,则粒子停滞,进入混沌搜索过程,否则跳至步骤(4);

① 设置 $m = 1$ 和最大混沌迭代次数 M ;

② 生成混沌序列,并将其映射到解空间;

③ 计算适应度 $F(\tilde{\mathbf{y}}_m)$,并更新混沌迭代第 o 个粒子的历史最优适应度 $\tilde{F}_o^{\text{best}}$ 和历史最优位置 $\tilde{\mathbf{P}}_o^{\text{best}}$;

④ 迭代。令 $m = m + 1$,返回步骤②直至 $m = M$;

⑤ 若 $\tilde{F}_o^{\text{best}}$ 优于 F_o^{best} ,则用第 o 个粒子的位置 $\tilde{\mathbf{P}}_o^{\text{best}}$ 和速度 $\tilde{\mathbf{v}}_o$ 替换原来的位置和速度,否则保持

不变;

(4)更新粒子速度,并移动粒子到新位置上;

(5)令 $k = k + 1$,返回步骤(2),直至 $k = K$;

(6)阈值分割。根据迭代后最优解所在的位置 g^{best} ,输出最佳阈值向量 (t^*, s^*) ,并进行图像分割。

上述算法在迭代更新每个粒子适应度的过程中,对每个位置向量 (t, s) 都从 $i = 0, j = 0$ 开始计算 $\omega_0(t, s), \mu_i(t, s)$ 和 $\mu_j(t, s)$,而由粒子更新过程可知,由于每个粒子仅在自身位置的 $[v^{min}, v^{max}]$ 范围内变化,因此存在着大量的重复计算。为此,本文在混沌粒子群优化算法的迭代过程中给出下列快速递推算法,以进一步降低算法的运行时间。假设迭代前第 o 个粒子的位置为 (t, s) ,则更新后的第 o 个粒子的位置为 $(t + \Delta t, s + \Delta s)$,其中 Δt 和 Δs 分别表示粒子位置更新过程的变化量,在 $[v_o^{min}, v_o^{max}]$ 范围内变化。粒子在更新过程中,仅可能“飞向”相对自身的 4 个方向之一,若 $\Delta t > 0, \Delta s > 0$,则依据下列递推算法来计算当前粒子的适应度,其他情况与此类似,据此可大大减少重复计算。由式(3)与式(6)可知:

$$\omega_0(t + \Delta t, s + \Delta s) = \omega_0(t, s) + \sum_{i=t+1}^{t+\Delta t} \sum_{j=s+1}^s p(i, j) + \sum_{i=0}^{t+\Delta t} \sum_{j=s+1}^{s+\Delta s} p(i, j) \quad (18)$$

$$\mu_i(t + \Delta t, s + \Delta s) = \mu_i(t, s) + \sum_{i=t+1}^{t+\Delta t} \sum_{j=0}^s ip(i, j) + \sum_{i=0}^{t+\Delta t} \sum_{j=s+1}^{s+\Delta s} ip(i, j) \quad (19)$$

$$\mu_j(t + \Delta t, s + \Delta s) = \mu_j(t, s) + \sum_{i=t+1}^{t+\Delta t} \sum_{j=0}^s jp(i, j) + \sum_{i=0}^{t+\Delta t} \sum_{j=s+1}^{s+\Delta s} jp(i, j) \quad (20)$$

4 实验结果和分析

利用上述提出的基于灰度级-梯度 2 维直方图 Otsu 法及混沌粒子群递推的阈值选取方法(改进后算法)和常用的基于灰度级-平均灰度级 2 维直方图 Otsu 法及粒子群的阈值选取方法^[11](改进前算法),通过对大量不同类型的灰度图像进行的阈值分割实验发现,改进后算法的分割结果更为准确,收敛精度更高,且运行时间也相应减少。现给出其中 6 幅图像两种算法分割的实验结果(图 3~图 8),相应的阈值列于表 1。为便于比较,针对每一幅图像分别采用现有的基于灰度级-平均灰度级 2 维直方图

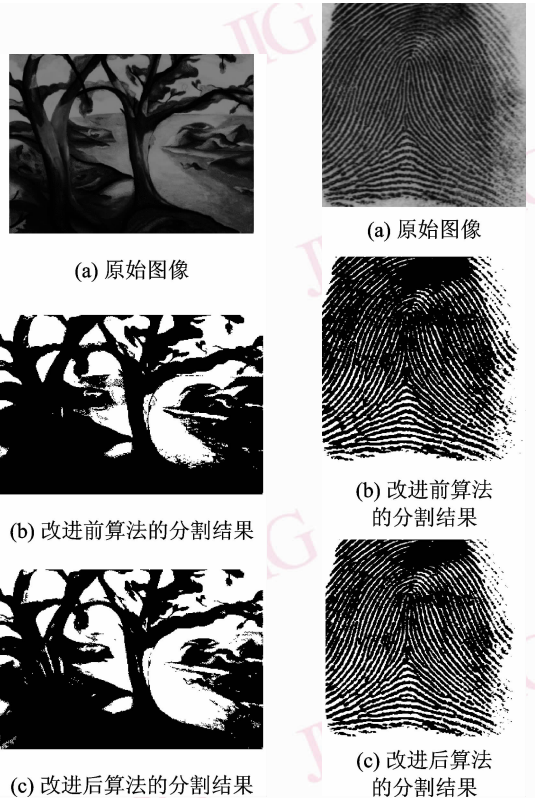


图 4 指纹图像分割结果
Fig. 4 Fingerprint segmentation results

图 3 树木图像分割结果
Fig. 3 Trees segmentation results



图 5 猩猩图像分割结果
Fig. 5 Baboon segmentation results

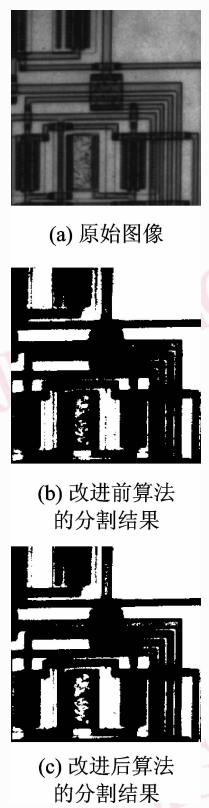


图 6 电路图像分割结果
Fig. 6 Circuit segmentation results



(a) 原始图像 (b) 改进前算法的分割结果 (c) 改进后算法的分割结果

图 7 辣椒图像分割结果

Fig. 7 Peppers segmentation results



(a) 原始图像 (b) 改进前算法的分割结果 (c) 改进后算法的分割结果

图 8 车牌图像分割结果

Fig. 8 Plate segmentation results

表 1 Otsu 阈值分割的 PSO 算法、CPSO 算法、RCPSO 算法的性能比较

Tab. 1 The performance comparisons of PSO, CPSO, RCPSO in Otsu thresholding

方法	树木			指纹			猩猩			电路			辣椒			车牌		
	最优 阈值	阈值 均值	运行时 间(s)	最优 阈值	阈值 均值	运行时 间(s)	最优 阈值	阈值 均值	运行时 间(s)	最优 阈值	阈值 均值	运行时 间(s)	最优 阈值	阈值 均值	运行时 间(s)	最优 阈值	阈值 均值	运行时 间(s)
灰度级-平均 灰度级 PSO 法	82	84.3	0.91	90	92.4	1.54	138	140.9	1.93	93	95.5	1.07	122	125.8	1.61	125	130.7	1.51
灰度级-梯度 CPSO 法	62	62.3	0.98	82	82.2	1.52	133	132.8	1.89	86	86.3	1.18	118	118.3	1.74	128	128.2	1.48
灰度级-梯度 RCPSO 法	62	62.2	0.23	82	82.2	0.51	133	132.7	0.57	86	86.4	0.25	118	117.1	0.52	128	128.3	0.42

及粒子群的算法、基于灰度级-梯度 2 维直方图及混沌粒子群算法与带递推的混沌粒子群算法来寻求最佳阈值向量,并对每种算法独立运行 20 次,以比较其收敛精度和运行时间,在 Intel Celeron M 1.4 GHz CPU、512 M 内存、MATLAB7.1 环境中所得到的运行时间与阈值均值也分别列于表 1。其中主要参数为粒子群规模 $N = 30$, 总迭代次数 $K = 30$, $v^{\min} = -v^{\max} = -[9 \ 9]$, $R_0 = [10 \ 10]$, $u = 4$, $M = 5$ 。

若仔细对照辨别图 3 ~ 图 8 可以看出,基于灰度级-梯度 2 维直方图 Otsu 法能使分割后的图像区域的内部均匀、边界形状更准确、细节特征更清晰。这是由于改进的 2 维直方图与如今常用的灰度级-平均灰度级 2 维直方图相比,已尽可能地考虑了所有目标点和背景点,致使目标和背景内点、边缘及噪声点的错分大为减少的缘故。通过进一步采用区域

内部均匀测度^[14]从客观上定量评价分割结果可见,均匀测度值越大,相应的算法分割效果越好,表 2 列出了基于灰度级-平均灰度级 2 维直方图 Otsu 法和基于灰度级-梯度 2 维直方图 Otsu 法所得到的分割图像的区域内部均匀测度值。从表 2 可以明显地看出,本文提出的算法性能优于如今常用的算法。

表 2 两种 2 维 Otsu 阈值分割算法区域内部均匀测度比较
Tab. 2 The uniformity comparisons of two Otsu methods

方法	树木	指纹	猩猩	电路	辣椒	车牌
灰度级-平均灰 度级法	0.987	0.981	0.982	0.981	0.981	0.984
灰度级-梯度法	0.992	0.992	0.988	0.990	0.988	0.993

分析表 1 中的数据可以看出,本文提出的基于混沌粒子群优化的阈值选取方法与基于基本粒子群

算法的阈值选取方法^[11]相比,由于引入了混沌序列重新初始化,其产生的局部最优解的邻域点,不仅可帮助“惰性”粒子逃离束缚,并可快速搜寻到全局最优解,因此收敛精度得到进一步提高;而带递推的混沌粒子群算法在计算适应度时,由于是根据飞行方向在自身基础上进行递推,从而大大减少了迭代过程中适应度函数的重复计算,其运行时间几乎不到文献[11]算法的1/3。

5 结 论

本文首先提出了一种基于灰度级-梯度的改进2维直方图 Otsu 法的阈值选取算法,提高了分割结果的准确性;然后利用混沌粒子群优化算法寻找最佳阈值,由于可帮助陷入“早熟”的粒子逃离束缚,从而提高了算法的收敛精度;最后提出在迭代过程中,采用递推的算法大大减少了适应度函数的重复计算。实验结果表明,与文献[11]提出的基于灰度级-平均灰度级2维直方图 Otsu 法及粒子群的快速图像分割方法相比,本文提出的方法不仅使分割后的图像区域的内部均匀、边界形状准确、特征细节清晰,同时运行时间几乎不到文献[11]算法的1/3,且粒子群处理的收敛精度得到进一步提高。因此本文提出的基于改进的2维直方图 Otsu 法及带递推的混沌粒子群的图像阈值分割算法在分割效果、收敛精度和运行速度等几方面都有很明显的优势。

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 (1) [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 (2) [J]. Journal of Data Acquisition & Processing, 1993, **8**(4): 268-282. [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, **9**(1): 62-66.

- 4 Abutaleb A S. Automatic thresholding of gray-level picture using two-dimensional entropies [J]. Pattern Recognition, 1989, **47**(1): 22-32.
- 5 Liu Jian-zhuang, Li Wen-qing. The automatic threshold of gray-level pictures via two-dimensional Otsu method [J]. Acta Automatica Sinica, 1993, **19**(1): 101-105. [刘健庄,栗文青. 灰度图象的二维 Otsu 自动阈值分割方法[J]. 自动化学报, 1993, **19**(1): 101-105.]
- 6 Gong J, Li L Y, Chen W N. Fast recursive algorithm for two-dimensional thresholding [J]. Pattern Recognition, 1998, **31**(3): 295-300.
- 7 Jing Xiao-jun, Cai An-ni, Sun Jing-ao. Image segmentation based on 2-D maximum between-cluster variance [J]. Journal of China Institute of Communication, 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.]
- 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. 2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 Zheng Hong, Pan Li. The automatic selection of image threshold on the basis of genetic algorithms [J]. Journal of Image and Graphics, 1999, **4**(4): 327-330. [郑宏,潘励. 基于遗传算法的图像阈值的自动选取[J]. 中国图象图形学报, 1999, **4**(4): 327-330.]
- 10 Kennedy J, Eberhart R C. Particle swarm optimization [A]. In: Proceedings of IEEE International Conference on Neural Networks [C], Piscataway, NJ, USA, 1995: 1942-1948.
- 11 Tang Ying-gan, Liu Dong, Guang Xin-ping. Fast image segmentation based on particle swarm optimization and two-dimension Otsu method [J]. Control and Decision, 2007, **22**(2): 202-205. [唐英干,刘冬,关新平. 基于粒子群和二维 Otsu 方法的快速图像分割[J]. 控制与决策, 2007, **22**(2): 202-205.]
- 12 He Qing-yuan, Han Chuan-jiu. Image thresholding segmentation with Otsu based on particle swarm optimization algorithm [J]. Journal of Guilin University of Electronic Technology, 2006, **26**(5): 355-358. [何庆元,韩传久. 基于粒子群算法的 Otsu 法图像阈值分割[J]. 桂林电子科技大学学报, 2006, **26**(5): 355-358.]
- 13 Meng Hong-ji, Zheng Peng, Mei Guo-hui, et al. Particle swarm optimization algorithm based on chaotic series [J]. Control and Decision, 2006, **21**(3): 263-266. [孟红记,郑鹏,梅国晖等. 基于混沌序列的粒子群优化算法[J]. 控制与决策, 2006, **21**(3): 263-266.]
- 14 Zhang Yu-jin. A classification and comparison evaluation technique for image segmentation [J]. Journal of Image and Graphics, 1996, **1**(2): 151-158. [章毓晋. 图象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.]