

# 基于 WFSOA 的 2D-Otsu 钢轨缺陷图像分割方法



曹义亲 段也钰 武丹

华东交通大学软件学院 南昌 330013

(yqcao@ecjtu.jx.cn)

**摘要** 针对二维最大类间方差阈值法(2D-Otsu)抗噪性较弱、计算时间较长的问题,文中提出了一种基于随机权重及异步价值因子取值的人群搜索算法,并将其应用于2D-Otsu中对钢轨缺陷图像进行分割。该算法采用随机权重加快收敛速度,采用异步价值因子提高搜索能力,有利于全局收敛到最优值。根据测试函数分析,WFSOA算法能够快速收敛,寻优值结果精度高,收敛时间短,算法稳定性好。在钢轨缺陷图像分割中,将2D-Otsu的迹函数作为WFSOA的目标函数,实验结果表明图像检测实时性高,对表面灰度不匀或生锈的钢轨缺陷分割结果清晰,有效降低了钢轨缺陷误检率和漏检率,在计算时间上仅占2D-Otsu算法的2%,可满足实际工程对实时性的需求。

**关键词:** 图像分割; 阈值; 2D-Otsu; 改进人群搜索算法; 钢轨缺陷

中图分类号 TN391.41

## 2D-Otsu Rail Defect Image Segmentation Method Based on WFSOA

CAO Yi-qin, DUAN Ye-yu and WU Dan

School of Software, East China Jiaotong University, Nanchang 330013, China

**Abstract** Aiming at the problem that the two-dimensional maximum inter-class variance threshold method (2D-Otsu) had weak anti-noise and long calculation time, a seeker optimization algorithm based on random weight and asynchronous value factor is proposed. The algorithm is applied to the image segmentation of rail defects in 2D-Otsu. The random weight is used to speed up the convergence speed of the algorithm, and the asynchronous value factor is used to improve the algorithm's search ability, which is conducive to global convergence to the optimal value. Through the test function analysis, the WFSOA algorithm can converge quickly, the precision value of the optimization value is high, the convergence time is small, and the algorithm has good stability. In the image segmentation of rail defects, the 2D-Otsu trace function is used as the objective function of WFSOA. The experimental results show that the image detection has high real-time performance, the segmentation result of the rail defects is clear, and the false detection rate and missed detection rate of the rail defects are effectively reduced. Time is only 2% of the 2D-Otsu algorithm, which meets the needs of actual engineering.

**Keywords** Image segmentation, Threshold, 2D-Otsu, Improved seeker optimization algorithm, Rail defect

### 1 引言

图像分割是图像识别与分类的基础。该技术由于能够有效简化图像,突出目标区域,降低图像分析难度,因此在图像处理领域中得到了广泛应用<sup>[1]</sup>。在铁路运输领域,由于钢轨伤损程度直接影响列车行车安全,因此基于图像处理的钢轨图像缺陷分割技术成为了研究的焦点<sup>[2]</sup>。

常见的图像分割方法主要有区域分割法<sup>[3-4]</sup>、边缘分割法<sup>[5-6]</sup>、阈值分割法<sup>[7-8]</sup>3种。其中,阈值分割法因实现简单、运算效率高、性能稳定而得到众多学者的广泛研究。阈值分割法中最为经典的Otsu法由大津于1979年提出;Liu等<sup>[9]</sup>、

Gong等<sup>[10]</sup>在此基础上利用像素灰度级和邻域平均灰度,将其扩展到2D-Otsu,有效改善了算法的抗噪性能。但2D-Otsu因计算复杂度高,分割时间延长,不能满足钢轨实时检测的要求;且2D-Otsu的全局寻优能力差,难以得到全局最佳阈值<sup>[1]</sup>。

针对传统2D-Otsu二维直方图存在错分、阈值求解不准确的缺陷,国内外相关学者做了诸多改进。Fan等<sup>[11]</sup>充分考虑了边界区域信息对分割结果的影响,提出了2D-Otsu曲线阈值法,将递归算法和小波变换相结合,达到了分割速度快、求取阈值精度高的效果,但存在难以准确选取阈值曲线的问题。Zhang等<sup>[12]</sup>采用快速递归算法,对二维直方图主对角线

到稿日期:2019-02-15 返修日期:2019-09-06 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61663009);江西省科技支撑计划重点项目(20161BBE50081)

This work was supported by the National Natural Science Foundation of China(61663009), Key Project of Jiangxi Science and Technology Support Plan(20161BBE50081).

通信作者:段也钰(1958194676@qq.com)

的目标和背景两区域分别求取阈值。该方法在降低计算复杂度、准确分割、增强抗噪性等方面取得了一定的成效。Wu 等<sup>[13]</sup>采用灰度级-梯度直方图及其区域划分方法,导出基于该直方图区域划分的快速迭代算法公式,求取最佳阈值时无须遍历整个直方图,达到了分割准确、用时短的效果,但此方法的普遍性不强。Chen 等<sup>[14]</sup>重新计算二维直方图对角线两个区域的概率,采用递归的方法求取阈值。该方法阈值精度高、抗噪性能强,但计算复杂。Yuan 等<sup>[15]</sup>通过求出的双阈值重新划分有效区域,在二维直方图上进行一维 Otsu 方法划分。实验表明,该方法具有较强的抗噪性能,分割效果较好。

上述算法需研究整个二维直方图,算法的复杂性高,计算量较大。为此,许多学者将 2D-Otsu 分割方法与智能算法相结合。Cheng 等<sup>[16]</sup>用个体最优适应值与群最优适应值之间的距离修正惯性因子的大小,使个体粒子在远离群最优值时加快进化速度,而接近群最优值时降低进化速度,从而快速求出 2D-Otsu 阈值并进行分割。Zhou 等<sup>[17]</sup>通过自适应权重优化的萤火虫算法搜寻 2D-Otsu 阈值,解决了萤火虫之间吸引力与相互距离成反比而使得算法局部搜索能力降低引起极值点附近震荡、无法寻优的问题,缩短了图像分割的运行时间。Cao 等<sup>[18]</sup>针对 2D-Otsu 穷尽搜索阈值导致分割时间长、分割效果差的问题,引入了狼群算法来获取 2D-Otsu 最佳阈值,缩短了分割时间,提高了图像分割精度。

上述方法能在一定程度上提高图像分割的精度或缩短分割时间,然而铁路环境多变,钢轨图片复杂,阈值分割法在钢轨缺陷分割过程中还存在以下难点:1)一般来说,钢轨缺陷的灰度比其他部分的灰度低,钢轨生锈区域的灰度与钢轨缺陷的灰度接近;2)长期使用的钢轨的轨顶摩擦面光亮而两侧相对暗淡,钢轨图像灰度分布不均,其灰度直方图呈多峰状态;3)轨检车速度较快,钢轨缺陷图像分割算法需满足实时性。

人群搜索算法(Seeker Optimization Algorithm, SOA)是一种新的智能算法,已被应用于数字图像处理<sup>[19]</sup>、PID 参数优化<sup>[20]</sup>、IIR 数字滤波器设计<sup>[21]</sup>、电力系统优化等诸多领域。SOA 在处理图像时具有收敛速度快、全局搜索能力强等优点,能够迅速、准确地求得 2D-Otsu 最佳阈值,进而进行图像分割。本文提出了一种基于随机权重策略及异步价值因子取值的人群搜索算法(Weight Factor Seeker Optimization Algorithm, WFSOA)。该算法采用随机权重加快收敛速度,采用异步价值因子提高搜索能力,并与 2D-Otsu 结合后用于钢轨缺陷图像的分割。实验结果验证了本文提出的改进方法对函数寻优时收敛代数更少,效率更高,且全局搜索能力更强,更易收敛到最优值;能对钢轨图像缺陷进行有效分割,分割的图像清晰,算法抗噪性能强;解决了钢轨图像复杂、钢轨缺陷灰度及钢轨表面灰度分布不均等问题,同时满足了钢轨缺陷检测对实时性的要求。

## 2 理论基础

### 2.1 2D-Otsu 算法

设有一幅图像  $f(x, y)$ , 其  $k$  阶邻域均值图像为  $g(x, y)$ , 灰度级均为  $L$ 。二维直方图中以  $f(x, y)$  灰度级为横坐标, 以

$g(x, y)$  灰度级为纵坐标, 则图像中任一像素可表示为像素灰度值  $i$  和邻域灰度值  $j$  构成的二维单元组  $(i, j)$ 。设该图像有  $M$  个像素, 则二维单元组  $(i, j)$  出现的概率为:

$$P_{ij} = \frac{f_{ij}}{M} \quad (1)$$

其中,  $f_{ij}$  为符合该像素的二维单元组的个数。

$$\begin{cases} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} f_{ij} = M \\ \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ij} = 1 \end{cases} \quad (2)$$

根据上述描述, 当任意设定一个阈值向量  $(u, v)$  时, 可将图像的二维直方图分割成图 1 所示的 4 个区域, 其中区域 I 代表图像的目标类, 区域 II 代表图像的噪声, 区域 III 代表图像的背景类, 区域 IV 代表图像中的边缘点。

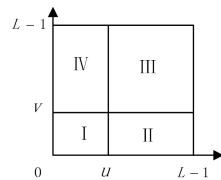


图 1 二维直方图

Fig. 1 Two-dimensional histogram

设该二维直方图中目标类和背景类出现的概率  $\omega_0$  和  $\omega_1$  分别为:

$$\begin{cases} \omega_0 = \sum_{i=0}^{u-1} \sum_{j=0}^{v-1} p_{ij} \\ \omega_1 = \sum_{i=u}^{L-1} \sum_{j=v}^{L-1} p_{ij} \end{cases} \quad (3)$$

目标类和背景类对应的均值矢量  $\eta_0$  和  $\eta_1$  为:

$$\eta_0 = (\eta_{0i}, \eta_{0j})^T = \left[ \sum_{i=0}^{u-1} \sum_{j=0}^{v-1} i p_{ij}, \sum_{i=0}^{u-1} \sum_{j=0}^{v-1} j p_{ij} \right]^T \quad (4)$$

$$\eta_1 = (\eta_{1i}, \eta_{1j})^T = \left[ \sum_{i=u}^{L-1} \sum_{j=v}^{L-1} i p_{ij}, \sum_{i=u}^{L-1} \sum_{j=v}^{L-1} j p_{ij} \right]^T \quad (5)$$

总灰度均值矢量  $\bar{\eta}$  为:

$$\bar{\eta} = (\bar{\eta}_i, \bar{\eta}_j)^T = \left[ \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} i p_{ij}, \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} j p_{ij} \right]^T \quad (6)$$

类间离散度矩阵为:

$$S = \sum_{k=0}^1 \omega_k [(\eta_k - \bar{\eta})(\eta_k - \bar{\eta})^T] \quad (7)$$

离散度矩阵的迹表示背景类和目标类的距离测度函数。迹函数为:

$$r_{tr}(S) = \frac{\sum_{k=0}^1 \omega_k [(\eta_{ki} - \bar{\eta}_i)^2 + (\eta_{kj} - \bar{\eta}_j)^2]}{\omega_0(1 - \omega_0)} \quad (8)$$

距离测度函数  $r_{tr}(S)$  取得最大值时的阈值向量  $(i, j)$  即为最佳阈值  $(u, v)$ 。

### 2.2 SOA 算法

SOA 算法是一种基于种群的启发式搜索算法, 能够直接应用于函数寻优, 效仿人类在搜索活动中的智能行为, 结合了搜索与进化的思想。该算法分析研究人的利己行为、利他行为、自组织聚集行为、预动行为和不确定性推理行为, 并通过对其建模来确定搜索步长和搜索方向。

#### 2.2.1 搜索步长的确定

基本 SOA 算法将种群中所有个体的适应度函数值降序排列并对应转换成从 1 到  $s$  的自然数, 则当前目标函数值  $i$  的隶属度为:

$$\mu_i = \mu_{\max} - \frac{s - I_i}{s - 1} (\mu_{\max} - \mu_{\min}); i = 1, 2, \dots, s \quad (9)$$

其中,  $\mu_i$  为目标函数值  $i$  的隶属度;  $I_i$  为目标函数值  $i$  排序后的序列编号;  $\mu_{\max}$  代表最佳位置, 取值为 1.0;  $\mu_{\min}$  代表最差位置, 取值为 0.0111。当  $\mu_i \in (0.0111, 1.0)$  时, 代表其他位置。

$$\mu_{ij} = \text{rand}(\mu_i, 1); j = 1, 2, \dots, D \quad (10)$$

其中,  $\mu_{ij}$  为  $j$  ( $1 \leq j \leq D$ ) 维搜索空间中目标函数值  $i$  的隶属度,  $\text{rand}(\mu_i, 1)$  表示在区间  $[\mu_i, 1]$  上均匀分布的随机数。根据  $\mu_{ij}$  不确定推理的行为得出步长:

$$\alpha_{ij} = \delta_{ij} \sqrt{-\ln(\mu_{ij})} \quad (11)$$

其中,  $\alpha_{ij}$  为  $j$  维搜索空间的搜索步长,  $\delta_{ij}$  为隶属度函数的参数, 其值由式(12)确定:

$$\vec{\delta}_{ij} = \omega \cdot \text{abs}(\vec{x}_{\min} - \vec{x}_{\max}) \quad (12)$$

$$\omega = (T_{\max} - t) / T_{\max} \quad (13)$$

其中,  $x_{\min}$  和  $x_{\max}$  分别为同一种群中具有最小和最大函数值的位置;  $\omega$  为惯性权重, 其取值随进化代数  $t$  的增大而线性递减, 取值范围为  $[0.1, 0.9]$ , 最大进化代数为  $T_{\max}$ 。

### 2.2.2 搜索方向的确定

SOA 算法的搜索者对搜索路径有一定的记忆功能, 并能进行自身预测及与他人沟通, 综合利己方向、利他方向和预动方向确定搜索者的位置方向。

利己方向:

$$\vec{d}_{i,\text{ego}}(t) = \vec{p}_{i,\text{best}} - \vec{x}_i(t) \quad (14)$$

利他方向:

$$\vec{d}_{i,\text{alt}}(t) = \vec{g}_{i,\text{best}} - \vec{x}_i(t) \quad (15)$$

预动方向:

$$\vec{d}_{i,\text{pro}}(t) = \vec{x}_i(t_1) - \vec{x}_i(t_2) \quad (16)$$

其中,  $\vec{x}_i(t_1)$  和  $\vec{x}_i(t_2)$  分别为  $\{\vec{x}_i(t-2), \vec{x}_i(t-1), \vec{x}_i(t)\}$  中的最佳位置和最差位置;  $\vec{g}_{i,\text{best}}$  为搜索者  $i$  所在邻域集体的历史最佳位置,  $\vec{p}_{i,\text{best}}$  为第  $i$  个搜索者搜索过程中的个体最佳位置。综合以上 3 个方向的随机加权确定 SOA 算法的搜索方向。

$$\vec{d}_i(t) = \text{sign}(\omega \vec{d}_{i,\text{pro}} + \varphi_1 \vec{d}_{i,\text{ego}} + \varphi_2 \vec{d}_{i,\text{alt}}) \quad (17)$$

其中,  $\omega$  为惯性权重, 其值由式(13)确定;  $\varphi_1$  和  $\varphi_2$  为价值因子, 是  $[0, 1]$  范围内的随机实数;  $\text{sign}(\cdot)$  为符号函数。

### 2.2.3 搜索位置的更新

搜索方向确定后, 进行搜索位置的更新。通过搜索方向和步长确定搜索者更新的位置, 从而迭代得到最佳位置。

$$\Delta x(t+1) = \alpha(t) d(t) \quad (18)$$

$$x(t+1) = x(t) + \Delta x(t+1) \quad (19)$$

其中,  $\Delta x(t+1)$  为搜索者  $i$  在  $D$  维空间的移动量;  $x(t)$  代表搜索者  $i$  的当前位置;  $x(t+1)$  代表搜索者  $i$  的更新位置。

通过搜索方向和步长确定搜索者更新的位置, 从而迭代得到最佳位置。

## 3 基于 WFSOA 的 2D-Otsu 钢轨缺陷图像分割

### 3.1 随机权重策略

在传统 SOA 算法中, 搜索步长与搜索方向能使搜索者的

位置得到更新。式(13)中惯性权重  $\omega$  随着进化代数  $t$  的增大而线性递减, 以平衡 SOA 算法的全局和局部搜索能力; 通过式(17)影响搜索者的预动方向, 增强搜索者目标导向的行为。

$\omega$  一般采用线性递减取值, 但存在两方面不足: 1) 迭代初期算法局部搜索能力弱, 当搜索结果接近全局最优点时, 因取值较大容易跳出, 减缓了算法的收敛速度, 且易错过最优解; 2) 迭代后期算法全局搜索能力弱, 易陷入局部最优解。

若将惯性权重设定为服从某种分布的随机数, 则可以打破线性递减取值的局限性, 在不同的取值情况下快速找到最优值。1) 当在进化初期, 搜索者接近最佳位置时, 若  $\omega$  随机产生较小的值, 则有利于加快收敛速度; 若  $\omega$  随机产生较大的值, 得到的适应函数值次于历史最佳值时, 则将其淘汰进行重新选择, 反之将其保存进行重新选择, 这有利于增大最优值的选取概率。2) 当在进化初期, 搜索者稍远离最佳位置时, 若  $\omega$  随机产生较大的值, 则同样有利于加快收敛速度; 若  $\omega$  随机产生较小的值, 得到的适应度函数值次于历史最佳值时, 则将此随机权重淘汰, 重新选择权重, 反之更新搜索者的最佳位置, 直到收敛到最佳的效果。

本文提出通过添加随机权重调整惯性权重, 避免算法陷入局部最优, 这有利于增加种群多样性并提高算法的全局搜索能力。 $\omega$  的改进公式如下:

$$\begin{cases} \omega = \omega^* + \sigma \cdot N(0, 1) \\ \omega^* = \omega_{\min}^* + (\omega_{\max}^* - \omega_{\min}^*) \cdot \text{rand}(0, 1) \end{cases} \quad (20)$$

其中,  $\omega$  代表随机权重;  $\omega_{\min}^*$  为随机权重的最小值;  $\omega_{\max}^*$  为随机权重的最大值;  $\text{rand}(0, 1)$  表示在区间  $[0, 1]$  上的随机数;  $N(0, 1)$  表示标准正态分布的随机数;  $\sigma$  为随机权重取值与其历史取值均值的偏离程度, 即方差, 用于控制其取值的权重误差。随机权重的第一项能在一定程度上避免算法陷入局部最优解, 第二项保证了其取值向期望权重方向进化。

### 3.2 价值因子的改进

人群搜索算法中, 通过利己行为、利他行为以及预动行为的随机加权确定搜索方向。式(17)中第一部分预动行为  $\omega \vec{d}_{i,\text{pro}}$  可理解为搜索者智能展现目标导向, 通过过去行为的影响来预测下一步的行为; 第二部分利己行为  $\varphi_1 \vec{d}_{i,\text{ego}}$  和第三部分利他行为  $\varphi_2 \vec{d}_{i,\text{alt}}$  分别遵循自我优先原则和群体优先原则, 并通过  $\varphi_1$  和  $\varphi_2$  加快算法的收敛速度。价值因子  $\varphi$  与随机权重  $\omega$  共同维护搜索者对全局和局部搜索能力的平衡, 价值因子较小, 可能使搜索者在远离目标的区域内徘徊; 价值因子较大, 可使搜索迅速向目标区域移动, 甚至超越目标区域。

$\varphi_1$  和  $\varphi_2$  的不同搭配, 将影响 SOA 算法的性能。本文采用异步变化的取值方式对价值因子赋值。初始阶段,  $\varphi_1$  和  $\varphi_2$  的取值有较强的利己行为能力和较弱的利他行为能力, 有益于提升全局搜索能力; 后期阶段,  $\varphi_1$  和  $\varphi_2$  的取值是较弱的利己行为能力和较强的利他行为能力, 有益于收敛到全局最优。价值因子的变化公式为:

$$\varphi_1 = \varphi_{1,\text{ini}} + \frac{\varphi_{1,\text{fin}} - \varphi_{1,\text{ini}}}{t_{\max}} \cdot t \quad (21)$$

$$\varphi_2 = \varphi_{2,\text{ini}} + \frac{\varphi_{2,\text{fin}} - \varphi_{2,\text{ini}}}{t_{\max}} \cdot t \quad (22)$$

其中,  $\varphi_{1,ini}$  和  $\varphi_{2,ini}$  分别为  $\varphi_1$  和  $\varphi_2$  的初始值,  $\varphi_{1,fin}$  和  $\varphi_{2,fin}$  分别代表  $\varphi_1$  和  $\varphi_2$  的迭代终值。

### 3.3 基于 WFSOA 算法的 2D-Otsu 钢轨缺陷图像分割流程

由 2D-Otsu 图像分割可知,该算法的本质是在二维灰度直方图中找到一个最佳阈值向量  $(u, v)$ , 以满足距离测度函数最大, 实现图像分割的最佳效果。由于传统的 2D-Otsu 算法的计算量大, 计算时间长, 不满足实时性的要求, 本文将 WFSOA 应用于 2D-Otsu 中, 以逆函数作为 WFSOA 算法的适应度函数, 阈值变量  $(u, v)$  作为搜索者的位置, 其最佳位置即为 2D-Otsu 的最佳阈值, 以此进行图像分割。

基于 WFSOA 算法的 2D-Otsu 钢轨缺陷图像分割的流程如图 2 所示。

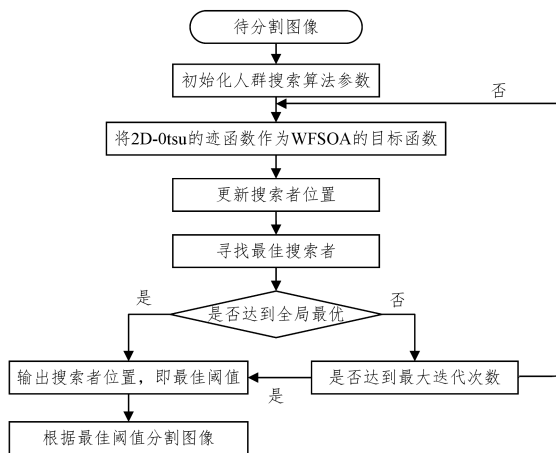


图 2 基于 WFSOA 算法的 2D-Otsu 钢轨缺陷图像分割流程图

Fig. 2 Flow chart of 2D-Otsu rail defect image segmentation based on WFSOA algorithm

## 4 实验结果及分析

### 4.1 测试函数及实验环境

为了对改进算法的收敛速度、求解精度和全局搜索能力进行测评, 将其与 ISOA<sup>[20-21]</sup>、粒子群算法 (Particle Swarm Optimization, PSO)<sup>[17]</sup>、WFPSO<sup>[22]</sup> 进行对比实验, 并通过 4 个典型测试函数来验证该算法的有效性。其中,  $F_1, F_2$  为多峰函数, 能够检验算法的全局搜索能力和求解精度;  $F_3, F_4$  是单峰函数, 用于检测算法的收敛速度和求解精度。在本文的对比实验中, 具体参数设置如表 1 所列, 相应函数的详细信息如表 2 所列。

本文对上述 4 种经典函数进行算法测试, 函数变量维数  $D=2$ , 种群规模  $N=40$ , 最大迭代次数为  $T_{max}=100$ 。实验环境为: Intel i5, CPU 2.50 GHz, RAM 4.00 GB, Window 7 操作系统, MATLAB 2014a。

表 1 参数设置

Table 1 Parameter setting

算法	参数设置
PSO	$\omega \in [0.4, 0.9], c_1 = c_2 = 1.4945$
WFPSO	$\omega, \omega^* \in [0.5, 0.8], \varphi, \varphi_{ini} = 2.5, \varphi_{fin} = 0.5$
ISOA	$\delta \in [0.0111, 1], \omega \in [0.1, 1], \varphi \in [0, 1]$
WFSOA	$\delta \in [0.0111, 1], \omega, \omega^* \in [0.5, 0.8], \varphi, \varphi_{ini} = 2.5, \varphi_{fin} = 0.5$

表 2 4 个测试函数

Table 2 4 test functions

函数名	测试函数	变量范围	最优值
Rosenbrock	$F_1 = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-2.05, 2.05]$	0
Ackley	$\rho_B = N_B/N \times 100\%$	$[-8, 8]$	0
Schafer	$\rho_C = N_C/N \times 100\%$	$[-10.0, 10.0]$	0
Rastrigin	$F_4 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{j=1}^n x_j^2}\right) - \exp\left(\frac{1}{n} \sum_{j=1}^n \cos(2\pi x_j)\right) + 20$	$[-5.12, 5.12]$	0

### 4.2 对比实验及结果分析

实验统计包括迭代曲线图、迭代次数、优化收敛结果以及收敛速度。

图 3—图 6 给出了各算法的适应度曲线对比结果。从收敛曲线可以直观地看出, WFSOA 算法的适应度曲线最为平稳, 说明该算法的稳定性最高; WFSOA 算法的测试函数适应度最早收敛于函数寻优值, 说明该算法的收敛速度最快, 求解精度高, 全局搜索能力相对较强。接下来从统计学的角度为适应度曲线图做进一步的数值分析。

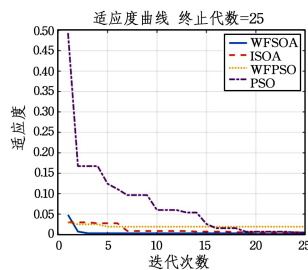


图 3  $F_1$  适应度曲线  
Fig. 3 Fitness curve( $F_1$ )

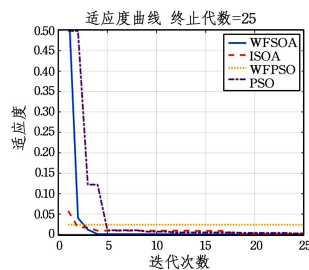


图 4  $F_2$  适应度曲线  
Fig. 4 Fitness curve( $F_2$ )

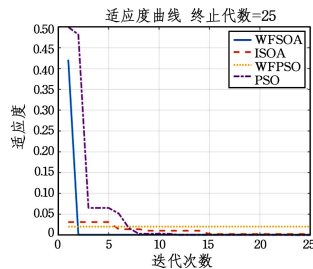


图 5  $F_3$  适应度曲线  
Fig. 5 Fitness curve( $F_3$ )

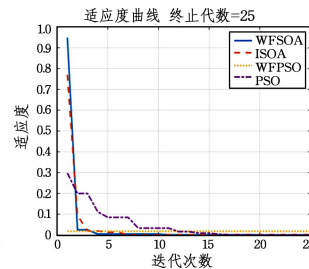


图 6  $F_4$  适应度曲线  
Fig. 6 Fitness curve( $F_4$ )

表 3 列出了不同改进算法对函数收敛代数的统计结果, 其中加粗数据表示各算法在相同测试函数中的最优结果。

表 3 收敛代数统计

Table 3 Statistics of convergence algebraic

算法	$F_1$		$F_2$		$F_3$		$F_4$	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
WFSOA	<b>3.3</b>	1.902	2.7	1.224	3.5	2.787	2.8	1.540
ISOA	4.2	2.508	2.9	1.451	6.6	3.8	3.1	2.049
WFPSO	4.5	3.684	3.3	3.016	<b>3.4</b>	<b>2.504</b>	5.1	3.534
PSO	8.9	3.447	7.3	3.462	12.1	3.922	8.8	4.474

对比表 3 中的数据得出, WFSOA 算法在测试函数  $F_1, F_2$  和  $F_4$  中的迭代次数均值和标准差最小, 说明此算法的收

敛代数少,稳定性强;在测试函数  $F_3$  中,与 WFPSO 算法相比,WFSOA 算法的收敛代数均值仅落后 0.1,标准差仅落后 0.2,考虑到参数取值的随机性,WFSOA 算法与 WFPSO 算法在测试函数  $F_3$  上的表现相似。本次实验结果表明,WFSOA 算法在同类算法中的收敛代数少,稳定性较高,性能较好。

表 4 中的数据为各算法对测试函数的寻优结果。对比表

4 中的数据得出,WFSOA 算法在  $F_1 - F_3$  测试函数的寻优结果和标准差最小,说明在对比实验中 WFSOA 算法的收敛性能表现更好,稳定性更强,寻优精度最高;ISOA 算法在测试函数  $F_4$  中的收敛结果略优于 WFSOA 算法,但其稳定性差。综合比较,WFSOA 算法更优。该实验结果表明,WFSOA 算法在测试函数求解收敛值时,收敛性能最佳,算法寻优效果突出。

表 4 收敛寻优结果统计

Table 4 Statistics of convergence and optimization results

算法	$F_1$		$F_2$		$F_3$		$F_4$	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
WFSOA	$0.109 \times 10^{-4}$	$0.989 \times 10^{-5}$	$0.101 \times 10^{-4}$	$0.921 \times 10^{-5}$	$0.931 \times 10^{-5}$	$0.105 \times 10^{-4}$	$0.223 \times 10^{-4}$	$0.253 \times 10^{-4}$
ISOA	$0.257 \times 10^{-4}$	$0.782 \times 10^{-4}$	$0.193 \times 10^{-4}$	$0.199 \times 10^{-4}$	$0.258 \times 10^{-4}$	$0.288 \times 10^{-4}$	$0.122 \times 10^{-4}$	$0.892 \times 10^{-4}$
WFPSO	$0.233 \times 10^{-4}$	$0.219 \times 10^{-4}$	$0.796 \times 10^{-1}$	0.273	$0.286 \times 10^{-4}$	$0.168 \times 10^{-4}$	$0.54 \times 10^{-2}$	$0.67 \times 10^{-4}$
PSO	$0.796 \times 10^{-4}$	$0.782 \times 10^{-4}$	$0.793 \times 10^{-4}$	$0.96 \times 10^{-4}$	$0.89 \times 10^{-4}$	$0.103 \times 10^{-3}$	$0.925 \times 10^{-4}$	$0.824 \times 10^{-4}$

为了对比各算法的收敛速度,本文对 4 种测试函数进行了 100 次实验,统计各算法收敛到最优值的时间。对比表 5 中的收敛时间得出,与同类算法相比,WFSOA 算法在 4 种测试函数中的迭代收敛时间最短,收敛速度最快。

表 5 收敛时间统计

Table 5 Statistics of convergence time

算法	(单位:s)			
	$F_1$	$F_2$	$F_3$	$F_4$
WFSOA	2.2301	1.6981	2.3689	1.7553
ISOA	2.3816	2.1432	3.7351	1.8380
WFPSO	3.6990	3.1553	3.0697	5.1577
PSO	9.5906	7.6266	13.4070	9.7434

从图 3—图 6 和表 3—表 5 可以看到,对于 4 种测试函数,WFSOA 算法能够快速收敛,寻优值结果精度高,收敛时间最短,稳定性最好。与同类智能算法相比,WFSOA 更值得推广应用。

### 4.3 钢轨缺陷图像分割结果及分析

实际钢轨图像常常存在轨面灰度不匀和钢轨表面生锈等情况,从而造成传统算法分割效果不佳,该问题的解决一直是钢轨缺陷图像分割研究的难点和重点。为了验证本文算法对钢轨缺陷图像分割的有效性,将 WFSOA 算法和 ISOA,PSO,WFPSO 算法与 2D-Otsu 算法相结合,并与 2D-Otsu 进行对比。本文对 RSDDs (Rail Surface Discrete Defects) dataset<sup>[23]</sup> 中的所有图像进行了分割,选取 Type-I RSDDs dataset 中的 rail\_1.jpg, rail\_19.jpg 和 rail\_37.jpg 进行展示,其分别代表钢轨普通缺陷、钢轨光照不均和钢轨生锈 3 种不同缺陷类型的图像。分割结果分别如图 7—图 9 所示,图中从左到右依次为原始图像、2D-Otsu、ISOA、PSO、WFPSO、WFSOA 图像分割结果及 GroundTruth 图像。



图 7 普通缺陷图像及分割图

Fig. 7 Ordinary defect image and segmentation image

由图 7 中的原始图像可知,钢轨普通缺陷图像因轮对摩擦等原因,同样存在灰度不匀的现象,图 7 与图 8 在钢轨缺陷图像分割时都受此影响。

由图 7 中的 2D-Otsu 分割图像可知,受轨面灰度不匀的影响,分割结果中包含灰度较低的钢轨区域,2D-Otsu 无法判断缺陷与轨面区域,分割效果不理想。而智能算法与 2D-Otsu 相结合使得分割效果有了较大的改善。由实验结果对比可知,用 PSO 算法求得的阈值进行分割,效果有所改善,但依然存在小片边缘区域检测不佳的问题;ISOA 算法与 WFPSO 算法的效果相似,都有较好的分割结果,但仍存在部分噪声;WFSOA 算法求得的阈值所分割的钢轨图像缺陷部分清晰,钢轨边缘平直,基本消除了离散而细小的噪声。因此,WFSOA 算法应用于 2D-Otsu 钢轨缺陷图像分割时,能够解决钢轨表面不匀的难题,分割效果最好,抗噪性能最强。



图 8 光照不均图像及分割图

Fig. 8 Uneven light image and segmentation image

图 8 中的钢轨图像由于右侧有光照,灰度较左侧更亮。从 2D-Otsu 算法的分割结果可见,该方法受灰度不匀的影响较大,钢轨图像中阴面灰度暗处被检测为缺陷,未能检测识别真正的缺陷。与智能算法结合的 2D-Otsu 算法得到了较大的改善,能够有效减弱灰度不匀的影响。本文算法与对比算法相比,检测结果与 GroundTruth 中的缺陷大小、形状相似度最高,所含噪声相对较少。

由图 9 的分割结果可知,受钢轨表面生锈与缺陷灰度相似的影响,2D-Otsu 算法的分割结果模糊,没有分割出连续的轨面区域,难以检测到钢轨的边缘,分割效果较差;将 WFSOA 算法的分割结果与其他算法进行对比,分割结果显示边缘清晰,分割结果中所含离散而细小的噪声点少。由此可见,WFSOA 算法与 2D-Otsu 结合进行生锈的钢轨缺陷图像分割,能够消除钢轨生锈的影响,分割效果较好,抗噪性能更强。



图9 钢轨生锈图像及分割图

Fig. 9 Rail rusty image and segmentation image

为了进一步定量评价本文算法对钢轨缺陷检测的效果,本文针对 Type-I RSDDs dataset 数据集中的 67 幅含缺陷的钢轨图像,从缺陷像素检测和缺陷个数检测两个方面对分割结果进行评价。

#### 1) 缺陷像素检测

缺陷像素通过精确率、召回率及综合指标评价 3 个评价标准进行检测。各指标的计算公式如下:

$$PRE = TP / (TP + FP) \times 100\% \quad (23)$$

$$REC = TP / (TP + FN) \times 100\% \quad (24)$$

$$F = 2 \times PRE \times REC / (PRE + REC) \quad (25)$$

式(23)表示精确率,其中  $PRE$  代表检测的钢轨缺陷像素中正检个数的占比,  $TP$  为正检像素个数,  $FP$  表示误检像素个数;式(24)表示召回率,其中  $REC$  代表正检数与 Ground-Truth 样本中钢轨缺陷像素总数的比值,  $FN$  表示漏检像素个数;式(25)表示综合指标评价,其中  $F$  是综合考虑精确率和召回率的加权调和均值,其值越大说明算法的检测效果越好。

#### 2) 缺陷个数检测

缺陷个数通过对缺陷的漏检率和误检率两个评价标准进行检测。各指标的计算公式如下:

$$\rho_B = N_B / N \times 100\% \quad (26)$$

$$\rho_C = N_C / N \times 100\% \quad (27)$$

式(26)为漏检率,其中  $\rho_B$  为漏检率,  $N_B$  为漏检个数,  $N$  为缺陷总数;式(27)为误检率,其中  $\rho_C$  为误检率,  $N_C$  为误检个数。

本实验的研究样本为 Type-I RSDDs 数据集中的 67 幅钢轨缺陷图像,该数据集中的缺陷是由铁路表面检测领域的专家观察标记的。将原始图像进行二值化处理,并将缺陷区域的像素值设为 255,背景区域的像素值设为 0,从而突出缺陷区域,形成 GroundTruth 数据集。

在 Matlab 中,用 regionprops 函数对 GroundTruth 中每个缺陷的区域属性 Area 进行统计,从而得到该数据集中的 273 个缺陷。缺陷总数、尺寸以检测的缺陷区域属性为准。

采用上文提及的 5 种分割方法对 67 幅钢轨缺陷图像进行阈值分割后,得到钢轨缺陷图像的分割结果。将检测结果图像  $A$  与 GroundTruth 数据集图像  $B$  取交集,得到图像  $C$ ,可以统计正检缺陷数量  $N_A$  及其属性;图像  $A$  与图像  $C$  取差集,可以统计误检缺陷数量  $N_C$  及其属性;图像  $B$  与图像  $C$  取差集,可以统计漏检缺陷数量  $N_B$  及其属性。以漏检率  $\rho_B$ 、误检率  $\rho_C$  和钢轨缺陷图像分割的平均时间为评价指标,定量分析 WFSOA 算法的优势。

由图 7—图 9 所示的分割结果可以看出,2D-Otsu 算法无

法消除钢轨轨面灰度不匀的影响,甚至无法检测出钢轨图像中光照区域的缺陷,因此不对该方法所检测的正检缺陷与误检缺陷的个数做统计,仅从检测缺陷像素个数角度来说明 2D-Otsu 算法的检测效果。

由表 6 可知,2D-Otsu 算法对钢轨缺陷检测的准确率、召回率及  $F$  值都远小于另外 4 种方法,说明 2D-Otsu 算法在进行钢轨缺陷分割时,由于灰度不匀而导致误检像素点过多,从而造成准确率、召回率低、分割效果差。而 WFSOA+2D-Otsu 算法的准确率为 72.48%,召回率为 68.82%, $F$  值为 36.90%,3 个评价指标都高于其他对比算法,说明 WFSOA+2D-Otsu 算法在进行钢轨缺陷检测时,正检像素多,误检像素少,漏检率低,在钢轨缺陷检测上的效果最好。

表 6 缺陷像素级检测结果统计

Table 6 Statistics of defect pixel-level detection results

(单位:%)

算法	PRE	REC	F
2D-Otsu	9.04	29.85	4.62
ISOA+2D-Otsu	46.61	64.98	35.59
PSO+2D-Otsu	49.11	67.64	36.18
WFPSO+2D-Otsu	47.34	64.89	35.80
WFSOA+2D-Otsu	72.48	68.82	36.90

另外,对比表 7 中 4 种方法的误检率、漏检率数据可知,本文方法在漏检率上比传统 PSO 算法的漏检率高出 1.47%,其原因是钢轨图像存在灰度值接近、对比度不高的特点。PSO+2D-Otsu 算法的抗噪性能相比不强,所分割的图像噪声多,分割图像结果噪声点与缺陷同时被检测,因此该算法检测时的误判率高达 27.84%。PSO+2D-Otsu 算法虽能分割缺陷,但其误检率过高,不利于接下来的缺陷辨别与分类工作,故不能应用于钢轨图像检测。而本文的 WFSOA 算法的漏检率为 3.67%,相比其他算法有了明显的提高,且误检率远低于其他算法;钢轨缺陷分割的时间也是最短的,仅占 2D-Otsu 算法的 2%,能满足现代铁路钢轨缺陷检测的实时性要求。

表 7 缺陷个数检测及收敛时间结果统计

Table 7 Statistics of defect number detection and convergence

time results

算法	$N_A$	$N_B$	$N_C$	$\rho_B/\%$	$\rho_C/\%$	时间/s
2D-Otsu	—	—	—	—	—	2.3814
ISOA+2D-Otsu	251	22	28	8.42	10.26	0.0643
PSO+2D-Otsu	267	6	76	2.20	27.84	0.1026
WFPSO+2D-Otsu	255	18	39	6.59	14.29	0.0948
WFSOA+2D-Otsu	263	10	7	3.67	2.56	0.0485

综上所述,本文 WFSOA 算法在钢轨缺陷图像分割的应用中能够清晰地分割缺陷图像,减弱钢轨灰度不匀及生锈钢轨的影响,分割时间缩短,满足实时性的要求,并且正检缺陷像素点占比较高,误检率和漏检率较低,能够在钢轨缺陷图像分割中进行推广应用。

**结束语** 钢轨缺陷图像分割对于铁路安全运输、钢轨缺陷识别等有着重要的意义,为了改善图像分割的效果,优化阈值求取精度,本文提出 WFSOA 算法,并将其用于 2D-Otsu 算法的最佳阈值寻优,以实现钢轨缺陷图像的最佳分割效果。该算法的性能优于传统的 2D-Otsu 算法及其他各对比算法,

稳定性高,全局搜索能力强,求解最佳阈值时的精度高,分割时间明显缩短,钢轨缺陷检测实时性高。对于轨面灰度不均匀或生锈的钢轨缺陷,本文算法进行图像分割时的结果清晰,正检缺陷像素点的占比高,漏检率和误检率相对于其他算法有了较大的改善,能满足实际工程的需要。

### 参 考 文 献

- [1] WANG K, ZHANG Y. Infrared Image Segmentation Based on Improved Dimensional OTSU and Genetic Algorithm[J]. Journal of System Simulation, 2017, 29(6): 1229-1236.
- [2] LIU Z, WANG W, ZHANG X, et al. Inspection of Rail Surface Defects Based on Image Processing [C] // International Asia Conference on Informatics in Control. IEEE, 2010: 472-475.
- [3] DUBEY A K, JAFFERY Z A. Maximally Stable Extremal Region Marking-Based Railway Track Surface Defect Sensing[J]. IEEE Sensors Journal, 2016, 16(24): 9047-9052.
- [4] HOU Y, LI B L, LIU J J, et al. Railway Fasteners Positioning Feature Extraction Based on Gray-scale Invariance[J]. Computer Applications and Software, 2015, 32(11): 193-196.
- [5] MA K, VICENTE T F Y, SAMARAS D, et al. Texture classification for rail surface condition evaluation [C] // Applications of Computer Vision. IEEE, 2016: 1-9.
- [6] YU X. Adaptive Canny Operator Based Digital Image Processing Scheme for Photoshop Filter Plugin Unit [C] // International Conference on Intelligent Transportation, 2018.
- [7] MIN Y Z, YUE B, MA H F, et al. Rail Surface Defects Detection Based on Gray Scale Gradient Characteristics of Image [J]. Chinese Journal of Scientific Instrument, 2018, 39(4): 220-229.
- [8] YUAN X C, WU L S, CHEN W H. Rail Image Segmentation Based on Otsu Threshold Method [J]. Optics and Precision Engineering, 2016, 24(7): 1772-1781.
- [9] LIU J Z, LI W Q. Two-Dimensional Otsu Automatic Threshold Segmentation Method for Grayscale Images [J]. Acta Automatica Sinica, 1993, 19(1): 101-105.
- [10] GONG J, LI L, CHEN W. A Fast-Recursive Algorithm for Two-Dimensional Thresholding [C] // International Conference on Signal Processing. IEEE, 1998: 1155-1158.
- [11] FAN J L, ZHAO F. Two-Dimensional Otsu's Cure Thresholding Segmentation Method for Gray-Level Images [J]. Acta Automatica Sinica, 2007, 35(4): 751-755.
- [12] ZHANG X M, SUN Y J, ZHENG Y B. Precise Two - Dimensional Otsu's Image Segmentation and Its Fast Recursive Realization [J]. Acta Automatica Sinica, 2011, 39(8): 1778-1784.
- [13] WU Y Q, FAN J. Fast iterative algorithm for image segmentation based on an improved two-dimensional Otsu thresholding [J]. Journal of Electronic Measurement and Instrument, 2011, 25(3): 218-225.
- [14] CHEN Q, ZHAO L, LU J, et al. Modified two-dimensional Otsu image segmentation algorithm and fast realization [J]. Iet Image Processing, 2012, 6(4): 426-433.
- [15] YUAN J, CHENG G T. Rapid Otsu Method Based on Two-Dimensional Histogram of Double Slope [J]. Application Research of Computers, 2017, 34(6): 1905-1908.
- [16] CHENG W S, ZANG X J, ZHAO J, et al. Modified strategy to inertia weight in PSO for searching threshold of Otsu rule [J]. Optics and Precision Engineering, 2008, 16(10): 1907-1912.
- [17] ZHOU C H, TIAN L W, ZHAO H W, et al. Two-Dimensional Otsu Image Segmentation Based on Improved Firefly Algorithm [J]. Journal of Shenyang University (Natural Science), 2016, 28(1): 45-50.
- [18] CAO S, AN J C. A Fast Two-Dimensional Otsu Image Segmentation Algorithm Based on Wolf Pack Algorithm Optimization [J]. Computer Engineering & Science, 2018, 40(7): 1221-1226.
- [19] PARVANEH H, DIZGAH S M, SEDIGHIZADEH M, et al. Load Frequency Control of A Multi-Area Power System by Optimum Designing of Frequency-based PID Controller Using Seeker Optimization Algorithm [C] // Thermal Power Plants, 2016.
- [20] SAHA S K, KAR R, MANDAL D, et al. Digital Stable IIR Band Pass Filter Design Using Seeker Optimization Technique [J]. Advanced Materials Research, 2014, 905: 406-410.
- [21] DAI C H, CHEN W R, ZHU Y F, et al. IIR Digital Filter Design Via Seeker Optimization Algorithm [J]. Journal of Southwest Jiaotong University, 2009, 44(6): 871-876.
- [22] HE L W, YUAN Y, WANG Y S, et al. Placement Strategy Of Cloud Virtual Machine Based On WFSOA Algorithm [J]. Application Research of Computers, 2017, 34(2): 591-594.
- [23] GAN J, LI Q, WANG J, et al. A Hierarchical Extractor-Based Visual Rail Surface Inspection System [J]. IEEE Sensors Journal, 2017, PP(99): 1-1.



**CAO Yi-qing**, born in 1964, professor, is a member of China Computer Federation. His main research interests include image processing, pattern recognition.



**DUAN Ye-yu**, born in 1994, master's degree. Her main research interests include image processing and so on.