

改进粒子滤波在行人跟踪中的应用

陈 旭

(上海电力学院 自动化工程学院,上海 200090)

摘 要:为提高粒子滤波在目标跟踪中的性能,将萤火虫算法(Firefly Algorithm, FA)的优化思想引入粒子滤波,并用自适应差分进化(Self-adaptive Differential Evolution, SaDE)算法代替粒子滤波的重采样,提出一种改进的粒子滤波跟踪算法,并采用新的跟踪特征 HSV-iLBP 进行跟踪.该算法将 FA 用于粒子滤波的重要性采样,通过计算迭代来抽取更加有效的粒子,并将粒子滤波的重采样过程看作求解目标函数的最值问题,通过自适应差分进化算法的迭代寻找最优粒子,改善粒子的退化和贫化问题. HSV-iLBP 模型由于结合了维数低的 HSV 颜色特征和 iLBP 纹理特征,从而在提高跟踪鲁棒性的同时,能有效降低计算复杂度.通过仿真实验,验证了改进算法在行人跟踪上具有更好的精度和速度.

关键词:行人跟踪;粒子滤波;FA;自适应差分进化;HSV-iLBP

中图分类号:TP391.4

文献标志码:A

粒子滤波^[1]是一种基于蒙特卡洛的贝叶斯滤波技术,能很好地处理非高斯、非线性问题,广泛应用于目标跟踪领域^[2-3].粒子滤波的跟踪性能主要取决于重要性采样方法和重采样方法的设计,重要性采样在统计学上表现为用服从一个分布的样本去估计另一个分布的属性的过程,用于产生有效近似系统状态分布的粒子,文献[4]在重要性采样阶段利用无迹卡尔曼滤波(unscented kalman filter, UKF)生成替代分布加以改进,再用 Mean Shift 在粒子的传播范围内进行确定性搜索,提高了目标定位上的精度,但随着粒子数的增加,算法运行时间较长,缺乏实时性,将自适应权重粒子群算法(adaptive weight particle swarm optimization, AWPSO)引入粒子滤波^[5],利用寻优迭代获取粒子,使粒子更加接近似然区域,并加入自适应权重,有效提高跟踪的鲁棒性,能在一定程度上通过减少粒子数来提高跟踪的实时性,但 PSO 的收敛速度较慢,影响粒子抽取速度.

重采样是解决粒子退化的一种有效方法,目前存在多种重采样方法,如系统重采样^[6]、残差重采样^[7]和多项式重采样^[8]等.这些算法主要通过增加粒子的有效性来解决粒子的退化问题,因此重采样后权重高的粒子会被多次选取,导致粒子的多样性变差.使用差分进化算法(differential evolution, DE)代替粒子滤波的重采样^[9],能有效抑制粒子的贫化,但由于参数的选择难度大,参数选择不当容易造成个体多样性变差,从而出现早熟收敛现象,影响跟踪精度.

目标跟踪特征的选取直接影响目标的跟踪精度和速度,常用的颜色特征有 RGB 和 HSV,相比于 RGB, HSV 空间能较好反映人对色彩的感知和鉴别能力^[10].跟踪过程中,一种特征往往不足以充分体现一个目标,当存在背景颜色的干扰时,容易导致目标丢失,将纹理与颜色相结合^[11]可以有效提高目标跟踪的鲁棒性,但却增加了计算的复杂度.

针对以上问题本文将萤火虫算法^[12](firefly algorithm, FA)用于重要性采样部分进行粒子的抽取,并用自适应差分进化^[13](self-adaptive differential evolution, SaDE)算法代替粒子滤波的重采样,提出一种结合 FA、粒子滤波和 SaDE 的目标跟踪算法,并使用 HSV-iLBP 模型作为跟踪特征. FA 相比于 PSO 算法收敛速

收稿日期:2014-09-11;修回日期:2015-03-20.

基金项目:国家自然科学基金(51107080)

作者简介(通信作者):陈旭(1988—),男,广东汕头人,上海电力学院硕士研究生,研究方向为智能控制与自动化装置、运动目标检测、跟踪与识别, E-mail: chenxulife@sina.com.

度快,精度高,利用 FA 算法对粒子进行迭代更新,使提取的粒子更加趋近于系统状态分布,SaDE 在迭代过程中,通过参数的自适应可以防止出现早熟现象,通过变异、交叉和选择操作有效地保持了粒子的多样性和有效性,可以减少在跟踪过程中对粒子数的依赖性,从而实现少量粒子的准确跟踪. HSV 和改进 LBP 相结合作为目标特征,由于 HSV 和 iLBP 维数少,计算量小,在保证跟踪速度的同时提高目标跟踪的鲁棒性.

1 Firefly Algorithm

Firefly Algorithm 是由 Yang^[12]提出的一种仿生群智能算法,通过模拟萤火虫的发光特性并进行相互吸引,使弱光源个体向强光源个体趋近,在移动过程中优胜劣汰,最终获得最优个体,即最优解的过程.

以下 3 式分别表示萤火虫算法的相对荧光亮度、吸引度和位置更新.

$$I = I_0 \times e^{-r_{ij}}, \quad (1)$$

$$\beta = \beta_0 \times e^{-\gamma r_{ij}^2}, \quad (2)$$

$$x_i = x_i + \beta \times (x_j - x_i) + \alpha \times (\xi - 1/2), \quad (3)$$

其中, I_0 为萤火虫自身荧光亮度; γ 为光强吸收系数; r_{ij} 表示萤火虫 i 与 j 之间的空间距离; β_0 为最大吸引度; x_i 和 x_j 为萤火虫 i 和 j 所处的空间位置, α 为 $[0, 1]$ 区间内的常数, ξ 表示 $[0, 1]$ 区间内的均匀随机数. 算法流程^[14]如下:

- 1) 初始化算法, 设置参数 β_0, γ, α , 设萤火虫数目 m , 最大迭代次数 T_{\max} 或达到精度 ϵ ;
- 2) 随机初始化萤火虫初始位置, 通过目标函数计算萤火虫各自的最大荧光亮度 I_0 ;
- 3) 由(1)式和(2)式计算相对亮度 I 和吸引度 β ;
- 4) 用(3)式对萤火虫进行位置更新, 对处在最佳位置的萤火虫进行随机扰动, 并重新计算更新位置后的萤火虫亮度;
- 5) 当达到最大迭代次数或者达到搜索精度时输出全局最优解和最优个体, 否则转到步骤(3)进行下一次搜索.

2 HSV-iLBP

为提高跟踪的鲁棒性, 将纹理与颜色相结合作为跟踪目标的特征, 其中颜色特征使用 HSV 模型^[10], 分为黑色区域、灰色区域、白色区域和彩色区域:

黑色区域: $L = 0, v \in [0, 0.2)$

灰色区域: $L = |(v - 0.2) * 10| + 1, s \in [0, 0.2] \& v \in [0.2, 0.8)$

白色区域: $L = 7, s \in [0, 0.2] \& v \in (0.8, 1.0]$

$$\text{彩色区域: } L = 4H + 2s + v, H = \begin{cases} 0, h \in (0, 22] \\ 1, h \in (22, 45] \\ 2, h \in (45, 70] \\ 3, h \in (70, 155] \\ 4, h \in (155, 186] \\ 5, h \in (186, 278] \\ 6, h \in (278, 330] \end{cases}, s = \begin{cases} 0, s \in (0.2, 0.65] \\ 1, s \in (0.65, 1.0] \end{cases}, v = \begin{cases} 0, v \in (0.2, 0.7] \\ 1, v \in (0.7, 1.0] \end{cases}$$

总维数: $L = 4H + 2s + v + 8$ 共 36 维, 这种量化方法计算简单, 更符合人眼视觉.

纹理特征采用改进 LBP 模型, 取 $LBP_{8,1}^{riu2}$ ^[11] ($LBP_{8,1}^{riu2}$ 表示旋转不变等价 LBP 模型, 圆形邻域半径为 1, 此邻域内所选择的像素点为 8) 主要模式 2 ~ 6 作为纹理特征:

$$iLBP = \begin{cases} \sum_{p=0}^7 s(g_p - g_c + a), U(LBP_{8,1}) \leq 2, \sum_{p=0}^7 s(g_p - g_c + a) \in \{2, 3, 4, 5, 6\}, \\ 0, \text{其它.} \end{cases} \quad (4)$$

$$U(LBP_{P,R}) = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{p-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|, \quad (5)$$

$$s(\cdot) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}, \quad (6)$$

其中, g_c 表示中间像素点, g_p 是与 g_c 相邻的像素点, a 为阈值, R 为圆形邻域半径, P 表示在此邻域内所选择像素点的个数. 模式数由 $LBP_{8,1}^{riu2}$ 的 9 个降为 5 个.

将以上两种目标特征相结合, 建立颜色纹理融合直方图 HSV-iLBP, 维数为 $L \times 5$ 共 180 维, 该直方图模型维数少, 计算简单, 由于结合了颜色和纹理特征, 作为跟踪目标的特征, 鲁棒性强.

在目标行人跟踪上, 通过计算目标模型直方图 q_u 和候选目标模型直方图 p_u 之间的 Bhattachayya 距离 $d = \sum_{u=1}^m \sqrt{p_u(y) \cdot q_u}$ 作为目标函数值, 由于 Bhattachayya 距离表示直方图间的相似程度, 因此该目标函数属于最大化问题.

3 自适应差分进化

自适应差分进化算法是一种基于群体差异的启发式随机搜索算法, 该算法采用实数编码方式, 进化流程主要包括变异、交叉和选择操作, 并且对缩放因子和变异系数采用自适应更新机制, 使变异和交叉产生的个体更加“优秀”. 具体步骤如下^[13]:

1) 初始化种群. 种群初始化, 主要根据实际情况限定种群上下边界, 产生限定范围内的 N 个随机个体 $\{X_i | i = 1, 2, 3, \dots, N\}$, 每个个体有 D 维.

2) 变异操作. 由(9)式进行变异操作, 得到变异个体 V_i .

$$V_i = X_f + F \cdot (X_{r1} - X_{r2}), \quad (7)$$

$$V_i = X_i + F \cdot (X_f - X_i + X_{r1} - X_{r2}), \quad (8)$$

$$V_i = \begin{cases} Eq \cdot (7), & \text{若 } \xi < p \\ Eq \cdot (8), & \text{其它.} \end{cases} \quad (9)$$

其中, $i \neq r1 \neq r2$, X_f 表示最大目标函数值对应的个体(最大化问题), X_{r1} 和 X_{r2} 表示从 N 个随机样本中随机抽取的两个不同的个体, $F = N_i(0.5, 0.3)$ 为缩放因子, 表示均值为 0.5, 标准差为 0.3, 服从高斯分布的随机数, 用于控制样本的变异速度, p 值由式(10)计算, ξ 表示 $[0, 1]$ 之间的随机数,

$$p = \frac{ns_1 \cdot (ns_2 + nf_2)}{ns_2 \cdot (ns_1 + nf_1) + ns_1 \cdot (ns_2 + nf_2)}. \quad (10)$$

经过特定的循环代数^[13]后, 由(7)式和(8)式产生的个体能保留至下一代的次数分别记为 ns_1 和 ns_2 , 不能成功保留至下一代的次数分别记为 nf_1 和 nf_2 , 当 p 更新后, 重置 ns_1, ns_2, nf_1, nf_2 .

3) 交叉操作. 在目标个体 X_i 与变异个体 V_i 之间执行离散交叉操作, 产生试验个体 U_i :

$$U_i(j) = \begin{cases} V_i(j), & \xi \leq C_r \text{ 或者 } j = j_{\text{rand}}, \\ X_i(j), & \text{其它.} \end{cases} \quad (11)$$

其中, j 表示个体的维数, C_r 是交叉系数, 用于控制在哪些变量上采用变异值, 由(12)式和(13)式计算, j_{rand} 表示从 $[1, D]$ 区间内产生的随机整数, 用于防止试验个体完全复制 X_i .

$$C_r = N_i(C_{r_m}, 0.1), \quad (12)$$

$$C_{r_m} = \frac{1}{|C_{r_\xi}|} \sum_{k=1}^{|C_{r_\xi}|} C_{r_\xi}(k), \quad (13)$$

其中, 满足 $\xi \leq C_r$ 所产生的交叉个体如果能保留至下一代, 则将当前 C_r 值记录在 C_{r_ξ} 中, 当 C_{r_ξ} 不为空时, 在特定的循环代数^[13]更新 C_r 值和 C_{r_m} 值, C_{r_m} 值更新时, 重置 C_{r_ξ} .

4) 选择操作. 采用目标函数作为选择指标, 目标函数值小的变量保留至下一代:

$$X'_i = \begin{cases} U_i, f(U_i) \leq f(X_i), \\ X_i, \text{其它}, \end{cases} \quad (14)$$

其中, $f(\cdot)$ 为目标函数(最小化问题), X'_i 代替 X_i 成为下一代子个体。

通过以上操作可得到下一代群体 $\{X'_i | i = 1, 2, \dots, N\}$, 进而可以反复迭代直到满足终止条件或者达到最大迭代次数为止。

4 改进的粒子滤波算法

本文提出的改进粒子滤波算法, 以粒子滤波为框架, 在粒子滤波的重要性采样处使用 FA 对粒子进行迭代寻优, 使得采样出来的粒子更接近真实的后验概率, 并用自适应差分进化算法对粒子进行重采样, 增加粒子的多样性。具体算法流程如下:

1) 粒子初始化: 设粒子个数为 N , 视频帧数为 M , 用 $x = (a, b)$ 表示目标中心点, a 为纵向像素坐标, b 为横向像素坐标, 则粒子集可表示为 $x_i^t (i = 1, 2, \dots, N; t = 0, 1, 2, \dots, M-1)$ 。在视频第一帧指定的区域构建目标模型 q_u , 区域的中心点作为初始粒子值 x_0^i , 权重 $\omega_0^i = \frac{1}{N}$ 。利用 FA 算法, 计算第二帧的目标中心点 $\{x_1^i, \omega_1^i\}_{i=1}^N$ 。

2) 预测: 用前几帧的平均速度来预测下一帧的位置。由(15)式预测视频下一帧的目标中心点 x_t^i 。

$$x_t^i = x_{t-1}^i + \left(\frac{1}{t-1} \cdot \sum_{j=1}^{t-1} (x_j^i - x_{j-1}^i) \right) \cdot v \quad (15)$$

其中, $v = N(0, 1)$ 为服从高斯分布的随机数。

3) 重要性采样: 将预测得到的粒子用 FA 算法进行迭代, 得到新的粒子, 计算每个粒子的 Bhattachayya 距离作为粒子权重, $\{x_t^i, \omega_t^i\}_{i=1}^N$ 由 FA 算法迭代和计算 Bhattachayya 距离得到 $\{x''\}_t$ 。

4) 自适应差分进化重采样算法: 用 \tilde{T} 衡量粒子的退化程度, 其中 $\frac{1}{\tilde{T}} = \sum_{i=1}^N (\omega_i)^2$, 设置一个门限值 T , 当 $\tilde{T} < T$ 时执行以下步骤:

a) 将重要性采样所得的粒子作为初始种群。初始种群及对应权重为 $\{x''\}_t$ 。

b) 将 Bhattachayya 距离作为目标函数, 为最大化问题, 由(7)~(14)式进行变异、交叉和选择操作, 得到下一代群体 $\{x''\}_t^i, \omega''$ 。

c) 计算 \tilde{T} 值, 若 $\tilde{T} < T$, 将 $\{x''\}_t^i, \omega''$ 作为初始群体, 回到步骤 b) 计算下一代群体, 达到门限值时, 结束迭代, 得到最后一代群体 $\{\tilde{x}_t^i, \tilde{\omega}_t^i\}_{i=1}^N$ 。

5) 状态输出: 目标的最优估计由(16)式计算, 返回步骤 2), 继续下一帧的计算。

$$\bar{x}_t = \sum_{i=1}^N \tilde{x}_t^i \cdot \tilde{\omega}_t^i. \quad (16)$$

5 实验仿真

实验 PC 机配置为奔腾双核, 主频 2.5 GHz, 内存 2.0 GB, 软件工具为 Windows XP 操作系统, MATLAB R2013 运行环境。

为验证所用特征在目标跟踪中的鲁棒性, 视频 1 中目标与背景颜色相似, 采用 Mean Shift 分别使用 HSV 和 HSV-iLBP 两种特征进行跟踪。跟踪结果如图 1 所示。

图 1(a) 中仅采用 HSV 特征模型进行跟踪, 当背景与目标颜色相近时, 会导致目标丢失, 而本文将 HSV 与 iLBP 相结合, 可以解决相似背景的干扰, 跟踪鲁棒性高, 如图 1(b) 所示。

在视频 2 中, 采用 Mean Shift 分别使用 RGB(8x8x8) 和 HSV-iLBP 两种特征进行跟踪。效果如表 1 和图 2 所示。

表 1 两种目标特征的性能对比

目标特征	每帧平均处理时间/s
RGB	0.020
HSV-iLBP	0.013

图 2(a)和图 2(b)分别采用 RGB 和 HSV-iLBP 两种特征进行跟踪,均能准确跟踪目标.表 1 中,通过 10 次独立跟踪求平均的方式获得每帧的平均处理时间,其中在使用相同跟踪算法的情况下,本文提出的 HSV-iLBP 特征模型由于其维数少,计算简单,从而在进行每一帧的处理速度上要优于 RGB.

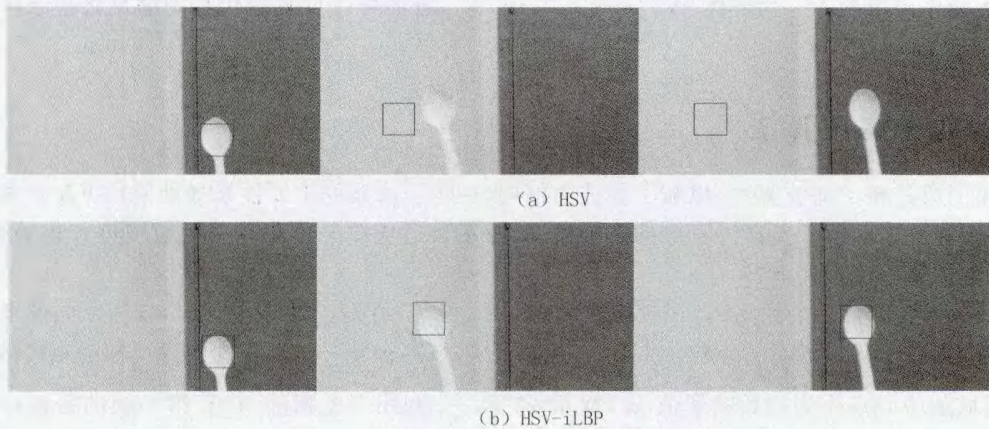


图1 采用不同目标特征的跟踪效果对比

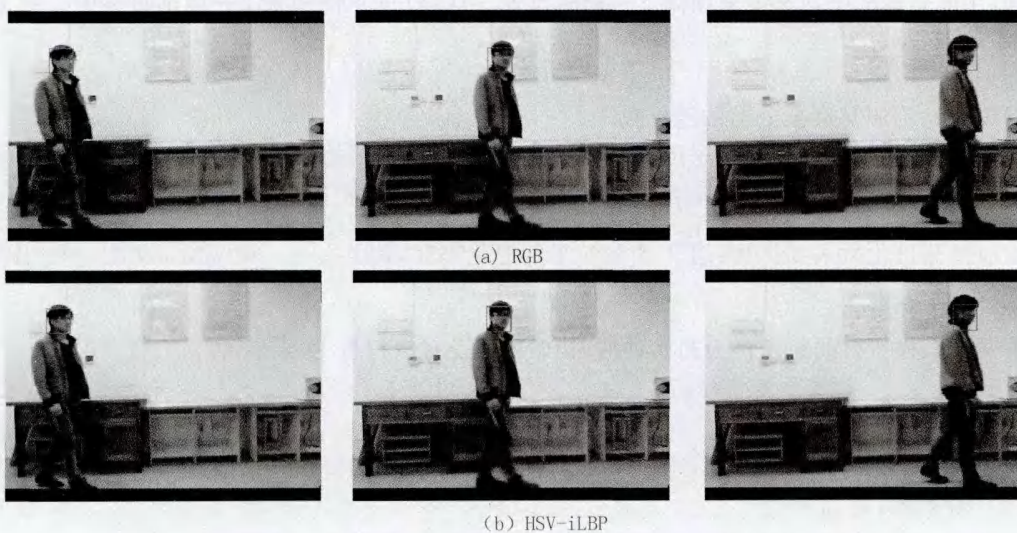


图2 两种目标特征的跟踪效果对比

为验证算法性能,进行计算机模拟跟踪仿真,对 AWPSO-PF^[5],PFDE^[9]和本文算法进行 $NUM=100$ 次独立跟踪试验,粒子数 N 取 10 个,本文算法 Cr 初始值设为 0.5, $\beta_0=1$, $\gamma=0.5$, $\alpha=0.16$,萤火虫数目 $m=50$,最大迭代次数 $T_{max}=200$;AWPSO-PF 粒子数 $m=50$,迭代次数与 T_{max} 相同;PFDE 参数同文献^[9], $RMSE = \sqrt{\frac{1}{NUM} \cdot \sum_{j=1}^{NUM} (x_t^j - \bar{x}_t^j)^2}$, x_t^j 为 t 时间第 j 次模拟跟踪的真实位置, \bar{x}_t^j 为 t 时间第 j 次模拟跟踪的估计位置,仿真结果如图 3 和表 2 所示.

图 3(a)为单次试验的位置估计,图 3(b)和表 2 为 100 次试验各个观测时间的 RMSE 值及 RMSE 对应的均值和方差,其中,PFDE 算法精度最高,但耗时最长,本文算法精度与 PFDE 相接近,实时性明显优于 AWPSO-PF 和 PFDE.

在实际的目标跟踪上,采用行人录像视频对 AWPSO-PF 和本文算法进行仿真试验,从录像视频的第 273 帧开始对目标行人进行跟踪,算法参数不变,实验结果如图 4 和表 3 所示(视频地址: http://groups.inf.ed.ac.uk/vision/CAVIAR/C_AVIAR_DATA1/).

图 4 中本文算法和 AWPSO-PF 均能准确跟踪目标;跟踪过程中每帧的平均误差,由 $\delta' = \frac{1}{M} \cdot \sum_{i=1}^M \delta_i$ 计

算,其中 $\delta_i = \sqrt{(a_i - a'_i)^2 + (b_i - b'_i)^2}$, a_i 和 b_i 为第 i 帧的实际位置, a'_i 和 b'_i 为第 i 帧估计位置,表 3 中,在实际的目标跟踪上,本文算法精度高于 AWPSO-PF,并且具有更好的实时性。

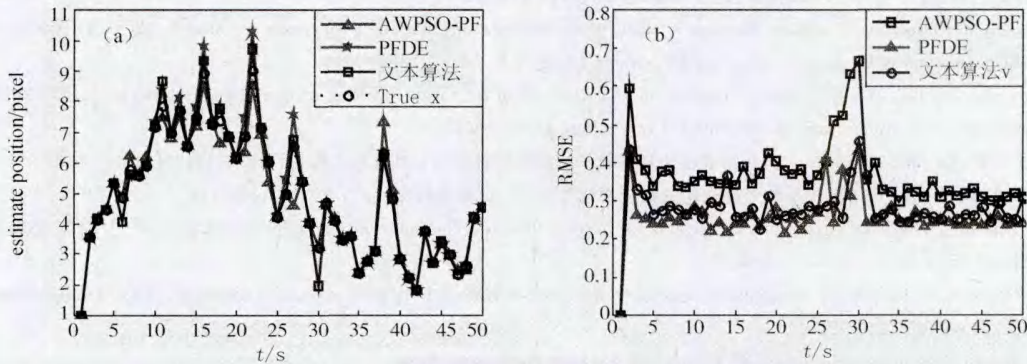


图3 3种算法在模拟跟踪仿真中的对比



图4 行人跟踪仿真

表 2 100 次试验 RMSE 的均值对比

所用算法	RMSE 均值	平均耗时/s
AWPSO-PF	0.364 4	0.043
PFDE	0.216 7	0.521
本文算法	0.252 7	0.031

表 3 跟踪性能对比

跟踪方法	每帧平均误差	每帧平均处理时间/s
AWPSO-PF	9.071 8	0.062 8
本文算法	8.468 1	0.034 1

6 结束语

在粒子滤波的重要性采样过程中,采用 FA 的寻优特性对粒子进行迭代更新,抽取有效的粒子,并在重采样部分使用 SaDE 算法进行替代,能更好地解决粒子的退化问题,改善粒子的多样性和有效性,从而降低对粒子数目的依赖,进而提高跟踪速度.本文提出的 HSV-iLBP 模型,可以克服相似背景的干扰,并且该模型维数低,计算简单。

在仿真试验中,相比于 FA,采用 UPF 进行粒子抽取能得到更加有效的粒子,但消耗时间太长,不利于进行实时跟踪,在重采样部分使用 SaDE 进行粒子寻优,能有效提高粒子的收敛速度,减少算法的运行时间,本文算法在跟踪精度上虽然略差于 PFDE,但拥有更好实时性.在实际行人跟踪仿真中,相比于 AWPSO-PF,本文算法拥有更好的跟踪精度和跟踪速度。

参考文献

- [1] 王法胜,鲁明羽,赵清杰等. 粒子滤波算法[J]. 计算机学报,2014,37(8):1679-1694.
- [2] Jayamohan S, Mathurakani M. Noise tolerance analysis of marginalized particle filter for target tracking[C]. 2013 International Conference on Microelectronics, Communications and Renewable Energy, Kerala,2013.
- [3] Dong Huiying, Cao Bin, Yang Yueping. Application of particle filter for target tracking in wireless sensor networks[C]. 2010 International Conference on Communications and Mobile Computing, Shenzhen,2010.
- [4] 刘璞,王春平,徐艳. 基于 Mean Shift 重要性采样的粒子滤波跟踪算法[J]. 计算机工程与设计,2014,35(3):909-913.
- [5] 刘峰,韩艳丽,王铎. 自适应权重粒子群优化的粒子滤波算法[J]. 计算机仿真,2013,30(11):330-333.
- [6] Bolic M, Djuric P, Hong S. Resampling algorithms for particle filters; a computational complexity perspective[J]. EURASIP Journal on Applied Signal Processing,2004(15):2267-2277.
- [7] Bolic M, Djuric P M, Sangjin H. Resampling algorithms and architectures for distributed particle filters[J]. IEEE Transactions on Signal Processing,2005,53(7):2442-2450.
- [8] Hol J D. Resampling in particle filters[R]. Sweden: Linkoping University,2004.
- [9] Li H W, Wang J, Su H T. Improved particle filter based on differential evolution[J]. Electronics Letters,2011,19(47):1078-1079.
- [10] 程言志,吕天阳,王钰旋,等. 基于表面属性颜色分布的三维模型检索[J]. 仪器仪表学报,2009,30(8):1707-1713.
- [11] 王保云,范保杰. 基于颜色纹理联合特征直方图的自适应 Mean shift 跟踪算法[J]. 南京邮电大学学报:自然科学版,2013,33(3):18-25.
- [12] Yang Xin-she. Firefly algorithms for multimodal optimization[C]. 5th Symposium on Stochastic Algorithms, Foundations and Applications, Sapporo,2009.
- [13] Elsayed S M, Sarker R A, Essam D L. An Improved Self-Adaptive Differential Evolution Algorithm for Optimization Problems [J]. Industrial Informatics, IEEE Transactions on,2013,9(1):89-99.
- [14] 刘长平,叶春明. 一种新颖的仿生群智能优化算法:萤火虫算法[J]. 计算机应用研究,2011,28(9):3295-3297.

Improved Particle Filter Algorithm for Pedestrian Tracking

CHEN Xu

(Automation Engineering College, Shanghai University of Electric Power, Shanghai 200090, China)

Abstract: In order to improve the performance of the particle filter in pedestrian tracking, an improved particle filter algorithm is proposed by combining the firefly algorithm (FA), self-adaptive differential evolution (SaDE) and particle filter. A new feature model called HSV-iLBP is used as the tracking characteristic. For an efficient tracking, FA is used in the important sample of the particle filter to generate the more effective particles by the iteration of optimization. The resampling process of the particle filter is regarded as an optimization problem and replaced by self-adaptive differential evolution (SaDE) to find out the best particles. To improve the robustness of tracking, color and texture are combined as HSV-iLBP tracking characteristic model. The low dimension of HSV-iLBP cause low computational complexity. Experiments show that the accuracy and speed of the pedestrian tracking is improved.

Keywords: pedestrian tracking; particle filter; FA; SaDE; HSV-iLBP