

基于 DE 蝙蝠算法优化粒子滤波的目标跟踪^①



李龙龙, 周武能, 闫斯瑶

(东华大学 信息科学与技术学院, 上海 201620)

通讯作者: 李龙龙, E-mail: dlhwxj@163.com

摘要: 在目标跟踪领域, 粒子滤波技术有处理非线性非高斯问题的优势, 但是标准粒子滤波在利用重采样方法解决退化现象时, 会产生粒子贫化问题, 导致滤波精度不稳定. 针对这种问题, 本文算法采用了差分进化蝙蝠算法对粒子滤波进行改进. 本文算法将粒子表征为蝙蝠个体, 蝙蝠种群通过调节频率、响度、脉冲发射率, 伴随当前最优蝙蝠个体在目标图像区域进行搜索, 并且可以动态决策是采用全局搜索还是进行局部搜索, 从而提高粒子整体的质量和合理的分布; 引进的差分进化策略可以增强蝙蝠个体跳出局部最优的能力. 为了验证本文算法的优化性能, 将本文算法和标准粒子滤波算法进行性能分析对比. 实验结果表明本文算法滤波性能优于标准粒子滤波算法.

关键词: 粒子滤波; 粒子贫化; 蝙蝠算法; 差分进化; 目标跟踪

引用格式: 李龙龙, 周武能, 闫斯瑶. 基于 DE 蝙蝠算法优化粒子滤波的目标跟踪. 计算机系统应用, 2019, 28(2): 24-32. <http://www.c-s-a.org.cn/1003-3254/6759.html>

Target Tracking Based on DE Bat Algorithm for Particle Filter Optimization

LI Long-Long, ZHOU Wu-Neng, LYU Si-Yao

(School of Information Science and Technology, Donghua University, Shanghai 201620, China)

Abstract: In the field of target tracking, particle filter technology has the advantage of dealing with nonlinear non-Gaussian problems. However, when the standard particle filter solves the degradation phenomenon by using the resampling method, the particle depletion problem will occur, resulting in unstable filter precision. To solve this problem, the algorithm uses the differential evolution bat algorithm to improve the particle filter. In this study, the particle is characterized as a bat individual. The bat population adjusts the frequency, loudness, and pulse emissivity, and the current optimal bat individual searches in the target image area, and can dynamically decide whether to use global search or local search to improve the particle. The overall quality and reasonable distribution; the introduction of differential evolution strategies can enhance the ability of bat individuals to jump out of local optimum. In order to verify the optimization performance of the proposed algorithm, the performances of the proposed algorithm and the standard particle filter algorithm are compared. Experimental results show that the filter performance of the proposed algorithm is better than the standard particle filter algorithm.

Key words: particle filter; particle depletion; bat algorithm; differential evolution; target tracking

1 引言

目标跟踪是计算机视觉中的一个重要主题, 它具

有许多真实世界的应用, 包括流量管理^[1], 机器人定位^[2]

和人机界面^[3]. 跟踪系统由两个主要部分组成: 视觉表

① 基金项目: 国家自然科学基金 (61573095)

Foundation item: National Natural Science Foundation of China (61573095)

收稿时间: 2018-07-12; 修改时间: 2018-08-09, 2018-08-30; 采用时间: 2018-09-11; csa 在线出版时间: 2019-01-28

示^[4]和运动估计^[5]. 在跟踪的运动估计步骤中, 对物体动态的不确定性或非线性进行建模, 以在视频序列中定位物体. 跟踪方法可以分为两组, 即运动估计的确定性^[6]和随机^[7]方法. Meanshift 算法是一个确定性的跟踪器, 它利用一个区域来迭代地最大化相似性度量. 该算法可能在遮挡, 前景和背景的相似颜色分布以及长视频序列中漂移^[8]. 相比之下, 随机跟踪器采用统计来定位目标跟踪的目标位置. 卡尔曼滤波方法是针对线性和高斯观测噪声^[9]而开发的, 作为一种顺序随机方式, 不能用于非线性运动. 粒子滤波方法保持了视觉跟踪中模型评估和分析步骤的非线性和不确定性^[10].

但是标准粒子滤波方法 (PF) 存在权值退化的问题, 标准粒子滤波在利用重采样方法解决退化现象时, 又会产生粒子贫化现象. 针对该问题, 文献^[11]提出了一种根据权值来进行选择的粒子滤波方法, 该算法在粒子群体中选取一些权值相对较大的粒子用于进行估计下一时刻的状态, 这样就能在一定程度上减轻粒子的贫化问题程度. 文献^[12]提出了一种确定性的重采样粒子滤波, 该粒子滤波方法提出了一种重采样的确定性优化思想, 这样增加了粒子状态的多样性. 文献^[13]提出了一种饱和粒子滤波方法, 这种方法是按照系统的不同特性挑选出相应比较适合的重采样方法, 使得粒子更加贴近真实状态. 但前面提出来的几种方法仍然是在传统重采样思想的基础上, 并不能从源头上解决粒子贫化的问题现象.

基于群智能优化算法的粒子滤波是现代粒子滤波领域中一个较为新颖的发展方向^[14]. 因为群智能算法改进粒子滤波主要是对粒子群体的分布状态展开迭代并寻优^[15,16], 其中没有涉及到对权值较低的粒子进行直接舍弃, 所以可以在源头上来解决掉粒子的贫化现象.

文献^[17]通过融合蜂群算法建立箱式粒子贫化解决策略, 结果较好. 文献^[18]通过将粒子群算法和蚁群算法融合在粒子滤波算法中, 实现粒子集间的信息共享, 增强算法的全局寻优能力. 文献^[19]通过融合萤火虫算法优化粒子滤波, 使用亮度和吸引力动态优化粒子集.

蝙蝠算法 (Bat Algorithm, BA)^[20]在 2010 年被剑桥大学 Yang 教授提出, 其实现智能寻优的方法是模拟现实世界蝙蝠的捕食行为, 类似于粒子群算法, 蝙蝠算法同样是一种基于群体的随机策略进行搜索来寻优的机制, 不同点在于蝙蝠算法在随机性拥有更强的优势. 文

献^[20]已证明, 蝙蝠算法的综合寻优能力优于粒子群算法、蚁群算法等主流群智能优化算法.

文献^[21]提出的蝙蝠算法优化粒子滤波, 并验证该算法优于其它智能算法, 但是带来计算效率低下的问题. 文献^[22]使用 Lévy 的飞行路线特征来改进蝙蝠算法优化, 增强了算法跳出局部极值的能力. 文献^[23]使用拉丁正交策略构建蝙蝠群位置初始策略, 来使得搜索区间分散均衡, 以提升性能.

上述的改进蝙蝠算法的过程中, 更多考虑的是算法参数和初始化策略改进, 能够获得一定的性能提升. 与上述算法的改进策略不同, 本文主要针对搜索区域进行策略调整.

本文在原有的蝙蝠优化算法中加入差分进化算法, 以改进算法的搜索能力, 并将其用于基于粒子滤波器的目标跟踪中, 在执行重采样之前进行差分进化蝙蝠优化搜索, 可以避免重采样本身的粒子贫化现象.

2 基本的粒子滤波算法

在标准的粒子滤波器中, 在时间 t 下, 目标状态矢量为 x_t , 考虑通过观测状态矢量为 z_t 来找到目标状态. 观测器从第一帧到第 t 帧为 $z_{1:t}$. 在粒子滤波器中, 后验分布可以使用查普曼-科莫高洛夫方程的近似模型来定义, 粒子集表示为 $\{x_1^t, x_2^t, \dots, x_N^t\}$, 权重集为 $\{w_1^t, w_2^t, \dots, w_N^t\}$.

$$w_i^t = w_{i-1}^t \frac{p(z_t | x_t^i) p(x_t^i | x_{t-1}^i)}{q(x_t^i | x_{t-1}^i, z_t)} \quad (1)$$

其中, q 是重要性密度函数. 前一次的后验分布是 $p(x_{t-1} | z_{1:t-1}) \approx \sum_{i=1}^N w_i^{t-1} \delta(x_{t-1} - x_{t-1}^i)$, 其中 $\delta(\cdot)$ 为一个狄拉克函数, 函数的约束条件是在前一时刻有 $\sum_{i=1}^N w_i^{t-1} = 1$, 且 $1 \leq i \leq N$.

3 蝙蝠算法

基于蝙蝠启发式的元启发式优化算法, 即蝙蝠算法, 由 Yang^[20]基于蝙蝠的回声定位的能力提出. 在标准的蝙蝠算法中, 蝙蝠的回声定位特征抽象化为如下规则^[20].

(1) 蝙蝠种群都通过回声定位来获得距离, 并使用某种特殊的方法“明白”目标和非目标两者之间存在的不同特征点;

(2) 蝙蝠在位置 x_i 处伴随着速度 v_i 以一定的速度

飞行频率和响度来搜寻目标. 他们可以自动调节发射脉冲的波长(或频率), 并根据目标的接近程度调节脉冲发射速率.

(3) 虽然响度可以按照多种方式变化, 但是通常是假设响度从大(正) A_0 变化到最小设定数 A_{\min} .

但是蝙蝠算法存在后期收敛速度较慢的、收敛精度不高、容易陷入局部最优点等不足. 本文提出的基于差分蝙蝠算法优化粒子滤波(DEBA-PF), 将差分算法融合到蝙蝠算法中, 使蝙蝠种群个体之间具有变异、交叉、选择机制, 从而达到增加蝙蝠个体的多样性, 提高算法的全局寻优能力和局部搜索能力, 能够避免陷入局部最优从而得到全局最优. 本文实验结果也证明 DEBA 效果更佳.

4 蝙蝠算法融合差分进化算法 DEBA

由于差分进化算法(DE)的有效性和简单性, 而使得差分进化算法获得很多的关注. 差分进化算法(DE)^[24]的基本思想在于应用当前种群个体的差异来重组种群进而得到中间种群, 然后应用父代个体与子代个体竞争来获得下一代种群. DEBA 算法的步骤如下:

Step 1. 初始化 DEBA 算法各个参数;

Step 2. 根据蝙蝠个体适应度函数得到各个蝙蝠权重和初始种群当前最优解;

Step 3. 模拟蝙蝠种群的全局搜索寻优行为, 根据本文中的等式(4)–(6)更新各个蝙蝠个体的对应的位置和速度;

Step 4. 模拟蝙蝠个体的局部搜索行为, 以及对对应生成一个均匀分布的随机数 $rand$, 如果存在 $rand > r_i$ 的情况, 则下一步将进行式(7);

Step 5. 生成另外一个独立的随机数, 若 $rand < A_i^g$ 且 $I(x_i^g) > I(x_i^{g+1})$, 则蝙蝠当前的位置为 x_i^g , 否则蝙蝠的当前位置为 x_{new} ;

Step 6. 对蝙蝠的局部搜索能力进行调整, 利用式(9)更新脉冲频率和响度;

Step 7. 利用差分进化算法以每一个蝙蝠位置为初始点进行变异、交叉、选择操作, 得到新的蝙蝠位置;

Step 8. 计算并对比适应度函数值, 更新全局最优值;

Step 9. 当算法符合设定的精度阈值 ϵ 时, 或者迭代次数达到设定值时, 这个时候就停止算法优化过程, 否则就要进入 Step 3.

从上述的 DEBA 算法步骤可以看到, BA 和

DEBA 不同点在于 DEBA 在每一次的进化过程中, 通过进化之后的蝙蝠个体位置 x_i^g 不是紧接着进行下一次的迭代操作, 首先是在蝙蝠个体之间进行变异、交叉、选择操作, 获得新的蝙蝠个体位置后, 再进行下一次迭代. 在本文的 DEBA 算法迭代的初期, 由于种群中蝙蝠个体的差异较大, 因此进行的变异操作可以使得算法拥有更强的全局搜索能力, 当过程进入到了迭代的后期, 趋于收敛的时候, 种群中蝙蝠个体的差异就会变小, 结果也使得算法拥有更强的局部搜索能力. 综上所述, DEBA 将 DE 和 BA 的各自优点结合在一起, 使得 DEBA 算法同时拥有更强的全局和局部搜索能力, 进而克服了 BA 后期收敛速度较慢的、收敛精度不高、容易陷入局部最优点等不足.

5 本文算法目标灰度描述和目标函数设计

本文选用灰度分布进行目标的描述, 建立系统观测模型. 设视频图像跟踪区域的中心为 $X = (x, y)$, 区域尺寸为 $h = (h_x, h_y)$, 跟踪区域内像素位置设为 $X_i = (x_i, y_i)$, 因此, 执行核概率密度估计以执行目标的灰度分布描述. 假设目标灰度分布被离散化为 B 个 bin 区间, 则定义灰度级量化函数 $b(X_i): R^2 \rightarrow \{1, \dots, B\}$ 表明将位置 X_i 处的像素灰度量级并分配给对应的灰度级 bin 中, 这里 B 表示灰度的量化等级, 由此定义目标的灰度分布为:

$$p_x^u = \frac{\sum_{i=1}^M k\left(\left\|\frac{X - X_i}{h}\right\|\right) \delta(b(X_i) - u)}{\sum_{i=1}^M k\left(\left\|\frac{X - X_i}{h}\right\|\right)} \quad (2)$$

其中, $\delta(\cdot)$ 为 Kronecker Delta 函数; 跟踪区域内的像素总数记为 M , $k(\cdot)$ 为所采用的高斯核函数.

本文算法设计的目标函数式如下所表示:

$$I = \exp\left[-\frac{1}{2R_g} (z_{\text{New}} - z_{\text{Pred}}(i))\right] \quad (3)$$

6 DEBA-PF 算法设计

6.1 DEBA-PF 全局搜索最优过程

对于每一个蝙蝠设为 i , 它的位置 x_i^{g-1} 和速度 v_i^{g-1} 定义在一个 d 维搜索空间中, 并且在迭代期间随后被更新. 在迭代 g 中更新的解决方案 x_i^g 和速度 v_i^g 是通过使用全局搜索策略来计算的, 公式如下:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (4)$$

$$v_i^{g+1} = v_i^g + (x_i^g - x^*)f_i \quad (5)$$

$$x_i^{g+1} = x_i^g + v_i^{g+1} \quad (6)$$

其中, $\beta \in [0, 1]$ 是一个从均匀分布中得到的随机矢量. x^* 是当前发现的全局最佳解决方案. 频率 f_{\min} 和 f_{\max} 的值取决于感兴趣问题的域大小.

6.2 DEBA-PF 局部搜索最优过程

对于局部搜索部分, 一旦从当前最佳解决方案中选择了解决方案, 则使用随机游走模型在局部生成每个蝙蝠的新的解决方案, 本文设置的局部搜索方法如下:

若 $rand > r_i$, 则

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A^g \quad (7)$$

若 $rand < r_i$, 则

$$x_{\text{new}} = x_i^{g+1} \quad (8)$$

其中, $\varepsilon \in [-1, 1]$ 是一个随机数, x_{old} 是当前最优解决方案集中的解决方案, $A^g = \langle A_i^g \rangle$ 是所有蝙蝠在迭代 g 时的平均响度.

6.3 DEBA-PF 全局和局部搜索寻优调整过程

通过响度 A_i 和脉冲率 r_i 的更新以反映如果找到目标, 那么蝙蝠就降低响度并增加脉冲率, 通过下式计算:

$$A_i^{g+1} = \omega A_i^g, r_i^{g+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (9)$$

其中, r_i^0 是初始脉冲率, ω 是脉冲频率增加系数, γ 是脉冲幅度衰减系数. 对于任何 $0 < \omega$ 和 $\gamma < 1$, 我们有以下:

$$A_i^g \rightarrow 0, r_i^g = r_i^0, \text{当 } g \rightarrow \infty \quad (10)$$

用差分进化算法以每个蝙蝠位置为初始点进行变异、交叉、选择, 得到新的蝙蝠位置.

变异操作过程: 从蝙蝠种群中随机选取 3 个蝙蝠个体: $x_a^g(t)$ 、 $x_b^g(t)$ 、 $x_c^g(t)$, 且有 $a \neq b \neq c \neq i, a, b, c \in \{1, 2, \dots, N\}$ 中的随机数. F 为缩放因子, $F \in [0, 2]$.

$$v_{ij}(t) = x_a^g(t) + F(x_b^g(t) + x_c^g(t)) \quad (11)$$

交叉操作过程: 交叉操作过程可以增多蝙蝠种群的多样性, 经过下式对第 g 代蝙蝠种群 $\{x_i^g(t)\}$ 和其变异中间个体 $\{v_i^{g+1}(t)\}$ 进行蝙蝠个体之间的交叉操作.

$$u_{ij}^{g+1}(t) = \begin{cases} v_{ij}^g(t), & \text{if } (rand(0, 1)) \leq CR \\ & \text{or } j = rand(1, n) \\ x_{ij}^g(t), & \text{if } (rand(0, 1)) > CR \\ & \text{or } j \neq rand(1, n) \end{cases} \quad (12)$$

在上式中, $CR \in [0, 1]$ 表示的是交叉概率, $rand(1, n) \in [1, 2, \dots, d]$ 的随机数, $j = 1, 2, \dots, d$. 此交叉操作策略能够保证 $u_{ij}^g(t)$ 中至少有一个分量是由 $v_i^g(t)$ 贡献.

选择操作过程: 通过适应度函数来判断蝙蝠个体 $u_{ij}^g(t)$ 是否优于 $x_i^g(t)$ 来产生新一代的蝙蝠个体 $x_i^{g+1}(t)$:

$$x_i^{g+1}(t) = \begin{cases} u_{ij}^{g+1}(t), & \text{if } (f(u_{ij}^{g+1}(t)) < f(x_i^g(t))) \\ x_i^g(t), & \text{else} \end{cases} \quad (13)$$

7 本文算法步骤

Step 1. 输入初始的测试图像, 并设定所要进行跟踪的目标对象;

Step 2. 随机生成 N 个蝙蝠个体 $\{x_i(0), i = 1, \dots, N\}$ 作为算法的初始蝙蝠个体, 初始化 DEBA-PF 算法参数. $x_i(t)$ 服从重要性密度函数:

$$x_i(t) \sim q[x_i(t) | x_i(t-1), z(t)] \quad (14)$$

Step 3. 计算蝙蝠个体的灰度直方图特征;

Step 4. DEBA 算法操作;

Step 5. 计算重要性权值:

$$w_i^t \approx w_i^{t-1} p(z^g | x_i^g) \quad (15)$$

Step 6. 进行归一化:

$$w_i^t = w_i^t / \sum_{i=1}^N w_i^t \quad (16)$$

Step 7. 根据蝙蝠权值大小重采样, 得到 N 个新样本;

Step 8. 状态输出, 对新样本求平均确定跟踪点:

$$\tilde{x}_t = \sum_{i=1}^N w_i^t x_i^g \quad (17)$$

上述算法过程充分利用了整个蝙蝠种群的有效信息, 帮助蝙蝠个体跳出局部最优, 避免了状态值变化不大的情况下还会继续进行迭代过程, 使得算法终止优化, 更多地是因为算法达到设定精度停止阈值, 从而降低算法直到达到设定的最高迭代次数才会终止下来的概率, 提高算法计算效率.

8 仿真实验和结果分析

8.1 性能测试分析

本文选取单变量非静态增长模型, 实验测试对象的过程模型和观测模型如下:

过程模型:

$$x(t) = 0.5x(t-1) + \frac{25x(t-1)}{1 + [x(t-1)]^2} + 8 \cos[1.2(t-1)] + n_{et}(t) \quad (18)$$

观测模型:

$$z(t) = \frac{x(t)^2}{20} + w_{et}(t) \quad (19)$$

上述等式中, $n_{et}(t)$ 和 $w_{et}(t)$ 为均值为零的高斯噪声. 在本文假定系统噪声方差 $Q=1$ 和 $Q=10$, 观测噪声方差 $R=10$, 滤波时间步数为 50, 初始化时脉冲响度为 $A_0=0.5$, 初始化脉冲率 $r_0=0.5$, $f_{max}=2$, $f_{min}=0$, 使用标准粒子滤波、粒子群优化粒子滤波和本文算法差分进化蝙蝠算法优化粒子滤波对该非线性系统进行状态估计与跟踪.

在本文算法中所使用的均方根误差公式:

$$R_{MSE} = \left[\frac{1}{T} \sum_{t=1}^T (x_t - \hat{x}_t)^2 \right]^{\frac{1}{2}} \quad (20)$$

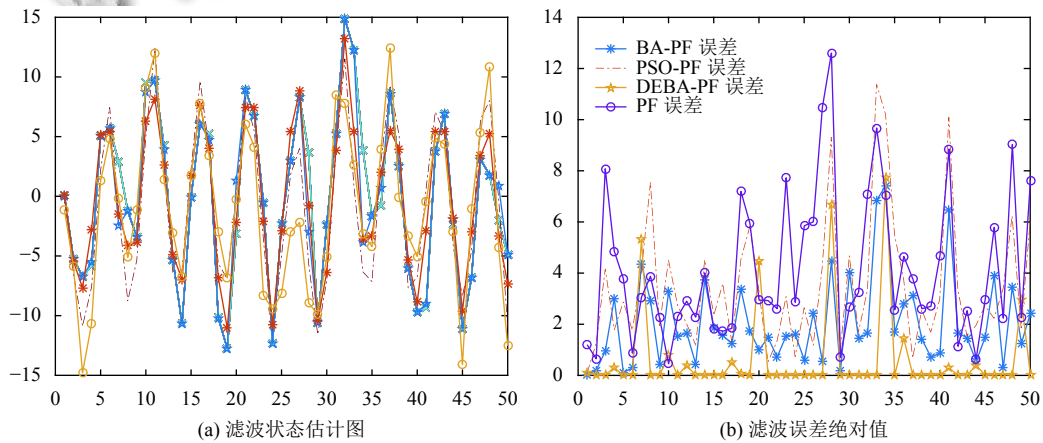


图1 $N=20$ 、 $Q=10$ 时滤波状态估计图和误差绝对值

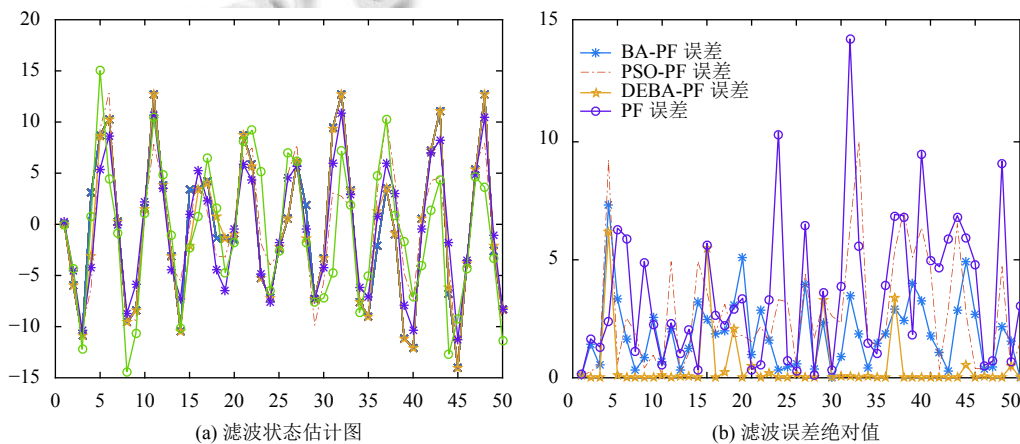


图2 $N=50$ 、 $Q=10$ 时滤波状态估计图和误差绝对值

(1) 当蝙蝠数为 $N=20$ 、 $Q=10$ 时, 仿真结果如图 1 所示.

(2) 当蝙蝠数为 $N=50$ 、 $Q=10$ 时, 仿真结果如图 2 所示.

(3) 当蝙蝠数为 $N=100$ 、 $Q=10$ 时, 仿真结果如图 3 所示.

(4) 当蝙蝠数为 $N=20$ 、 $Q=1$ 时, 仿真结果如图 4 所示.

(5) 当蝙蝠数为 $N=50$ 、 $Q=1$ 时, 仿真结果如图 5 所示.

(6) 当蝙蝠数为 $N=100$ 、 $Q=1$ 时, 仿真结果如图 6 所示.

在下示仿真图中, 用 x 表示真实点, *表示 BA 估计点, \circ 表示 PF 估计点, \cdot 表示 PSO 估计点, \star 表示 DEBA 估计点, 分别用实线连接.

从图1~图6可以看出,差分进化蝙蝠算法优化粒子滤波具有更高的状态值预测精度,这是因为差分进化算法优化粒子滤波的在粒子滤波的基础上,模拟蝙蝠个体和种群搜索目标的过程中,可以对全局搜索和

局部寻优能力进行精确控制,同时融合了差分进化算法使得本文算法能够较大幅度提高算法的收敛速度、鲁棒性和寻优能力.

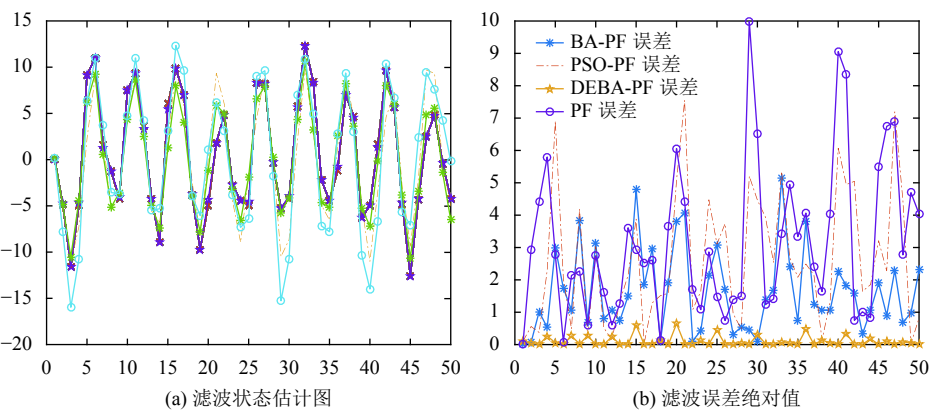


图3 $N=100$ 、 $Q=10$ 时滤波状态估计图和误差绝对值

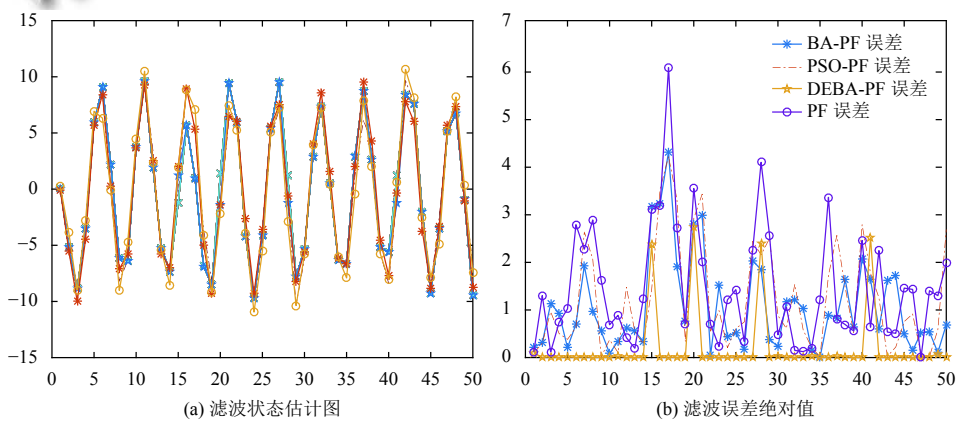


图4 $N=20$ 、 $Q=1$ 时滤波状态估计图和误差绝对值

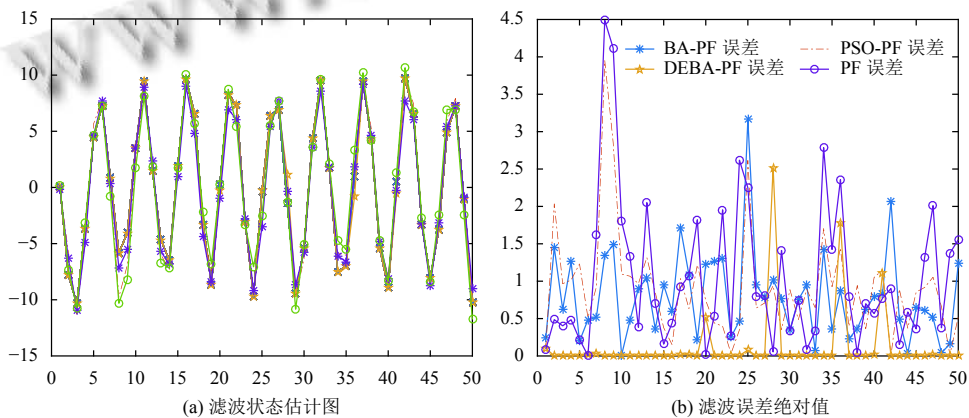


图5 $N=50$ 、 $Q=1$ 时滤波状态估计图和误差绝对值

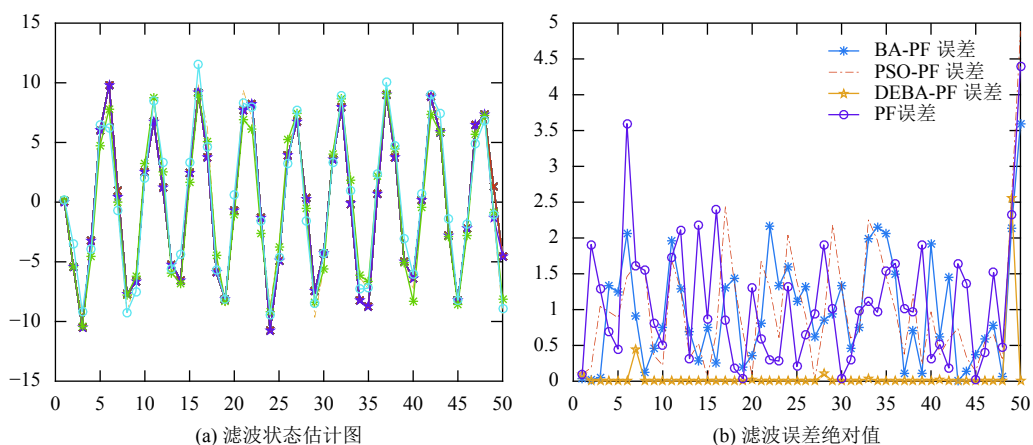


图6 $N=100, Q=1$ 时滤波状态估计图和误差绝对值

从表1中可以看出,在高噪声的影响下,差分进化蝙蝠算法粒子数 $N=20$ 时精度依然比标准粒子滤波中粒子数 $N=100$ 时的精度高,这样即表明,本文所用的差分进化算法在非线性和高噪声的环境会较标准的粒子滤波更好的跟踪目标。

8.2 目标跟踪测试

为了测试本研究算法的性能,本文在多个测试视频上进行了对比实验,如图7至图12所示。

表1 实验结果均方根误差对比

参数	R_{MSE}			
	PF	DEBA-PF	PSO-PF	BA-PF
$N=20, Q=10$	5.0320	1.8150	4.3163	2.6642
$N=50, Q=10$	4.6727	1.3900	3.6210	2.4146
$N=100, Q=10$	3.9321	0.1912	3.3076	2.0793
$N=20, Q=1$	1.9187	0.7116	1.6513	1.4303
$N=50, Q=1$	1.4480	0.4699	1.1585	0.9807
$N=100, Q=1$	1.4181	0.3684	1.3462	1.2440



图7 标准粒子滤波跟踪结果1(第50、60、70、80、90、100帧)



图8 差分进化蝙蝠算法跟踪结果1(第50、60、70、80、90、100帧)

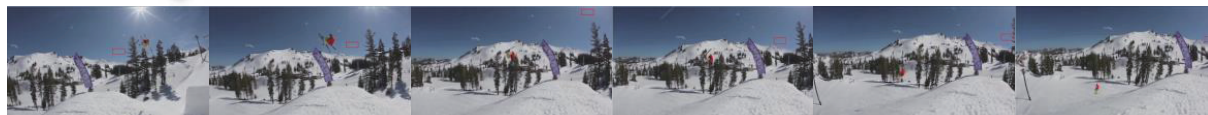


图9 标准粒子滤波算法跟踪结果2(第20、30、40、50、60、70帧)



图10 差分进化蝙蝠算法跟踪结果2(第20、30、40、50、60、70帧)



图 11 标准粒子滤波跟踪结果 3(第 50、60、70、80、90、100 帧)



图 12 差分进化蝙蝠算法跟踪结果 3(第 50、60、70、80、90、100 帧)

从以上实验结果可以看到, 在非线性下、高斯噪声环境下, 标准粒子滤波算法跟踪效果较差, 鲁棒性不佳, 而本文所采用的差分进化蝙蝠算法在总体上来说具有良好的跟踪效果和鲁棒性。

9 结语

本文提出了一种基于差分进化蝙蝠算法智能优化粒子滤波的目标跟踪方法, 同时具备全局搜索和局部搜索能力, 提高跟踪算法的跟踪性能。测试视频图像的实验结果也表明, 本文方法在复杂背景下的跟踪具有较好的鲁棒性和准确性。在后面工作中, 将对其它信号源图像的目标跟踪问题进行深入研究, 对跟踪过程中的模板更新、目标遮挡等问题做进一步优化。

参考文献

- 1 Yilmaz A, Javed O, Shah M. Object tracking: A survey. *ACM Computing Surveys*, 2006, 38(4): 13. [doi: [10.1145/1177352](https://doi.org/10.1145/1177352)]
- 2 Mekonnen AA, Lerasle F, Herbulot A. Cooperative passers-by tracking with a mobile robot and external cameras. *Computer Vision and Image Understanding*, 2013, 117(10): 1229–1244. [doi: [10.1016/j.cviu.2012.12.004](https://doi.org/10.1016/j.cviu.2012.12.004)]
- 3 Xu BL, Lu ML. An ant-based stochastic searching behavior parameter estimate algorithm for multiple cells tracking. *Engineering Applications of Artificial Intelligence*, 2014, 30: 155–167. [doi: [10.1016/j.engappai.2013.11.010](https://doi.org/10.1016/j.engappai.2013.11.010)]
- 4 Tian CN, Gao XB, Wei W, *et al.* Visual tracking based on the adaptive color attention tuned sparse generative object model. *IEEE Transactions on Image Processing*, 2015, 24(12): 5236–5248. [doi: [10.1109/TIP.2015.2479409](https://doi.org/10.1109/TIP.2015.2479409)]
- 5 Ait Abdelali H, Essannouni F, Essannouni L, *et al.* Fast and robust object tracking via accept-reject color histogram-based method. *Journal of Visual Communication and Image Representation*, 2016, 34: 219–229. [doi: [10.1016/j.jvcir.2015.11.010](https://doi.org/10.1016/j.jvcir.2015.11.010)]
- 6 Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, 25(5): 564–577. [doi: [10.1109/TPAMI.2003.1195991](https://doi.org/10.1109/TPAMI.2003.1195991)]
- 7 Sun X, Yao HX, Lu XS. Dynamic multi-cue tracking using particle filter. *Signal, Image and Video Processing*, 2014, 8(S1): 95–101. [doi: [10.1007/s11760-014-0674-z](https://doi.org/10.1007/s11760-014-0674-z)]
- 8 Mazinan AH, Amir-Latifi A. Applying mean shift, motion information and Kalman filtering approaches to object tracking. *ISA Transactions*, 2012, 51(3): 485–497. [doi: [10.1016/j.isatra.2012.02.002](https://doi.org/10.1016/j.isatra.2012.02.002)]
- 9 Ait Abdelali H, Essannouni F, Essannouni L, *et al.* An adaptive object tracking using Kalman filter and probability product kernel. *Modelling and Simulation in Engineering*, 2016, 2016: 2592368.
- 10 Yi SY, He ZY, You XG, *et al.* Single object tracking via robust combination of particle filter and sparse representation. *Signal Processing*, 2015, 110: 178–187. [doi: [10.1016/j.sigpro.2014.09.020](https://doi.org/10.1016/j.sigpro.2014.09.020)]
- 11 张琪, 胡昌华, 乔玉坤. 基于权值选择的粒子滤波算法研究. *控制与决策*, 2008, 23(1): 117–120. [doi: [10.3321/j.issn:1001-0920.2008.01.026](https://doi.org/10.3321/j.issn:1001-0920.2008.01.026)]
- 12 Li TC, Sattar TP, Sun SD. Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters. *Signal Processing*, 2012, 92(7): 1637–1645. [doi: [10.1016/j.sigpro.2011.12.019](https://doi.org/10.1016/j.sigpro.2011.12.019)]
- 13 Stano PM, Lendek Z, Babuška R. Saturated particle filter: Almost sure convergence and improved resampling. *Automatica*, 2013, 49(1): 147–159. [doi: [10.1016/j.automatica.2012.10.006](https://doi.org/10.1016/j.automatica.2012.10.006)]
- 14 Yu YH, Zheng XY. Particle filter with ant colony optimization for frequency offset estimation in OFDM systems with unknown noise distribution. *Signal Processing*,

- 2011, 91(5): 1339–1342. [doi: [10.1016/j.sigpro.2010.12.009](https://doi.org/10.1016/j.sigpro.2010.12.009)]
- 15 Zhong J, Fung YF. Case study and proofs of ant colony optimisation improved particle filter algorithm. *IET Control Theory & Applications*, 2012, 6(5): 689–697.
- 16 Xian WM, Long B, Li M, *et al.* Prognostics of lithium-ion batteries based on the verhulst model, particle swarm optimization and particle filter. *IEEE Transactions on Instrumentation and Measurement*, 2014, 63(1): 2–17. [doi: [10.1109/TIM.2013.2276473](https://doi.org/10.1109/TIM.2013.2276473)]
- 17 Leinonen M, Codreanu M, Juntti M. Distributed joint resource and routing optimization in wireless sensor networks via alternating direction method of multipliers. *IEEE Transactions on Wireless Communications*, 2013, 12(11): 5454–5467. [doi: [10.1109/TWC.2013.100213.121227](https://doi.org/10.1109/TWC.2013.100213.121227)]
- 18 邱雪娜, 刘士荣, 吕强. 基于信息分享机制的粒子滤波算法及其在视觉跟踪中的应用. *控制理论与应用*, 2010, 27(12): 1724–1730.
- 19 田梦楚, 薄煜明, 陈志敏, 等. 萤火虫算法智能优化粒子滤波. *自动化学报*, 2016, 42(1): 89–97.
- 20 Yang XS. A new metaheuristic bat-inspired algorithm. González JR, Pelta DA, Cruz C, *et al.* *Nature Inspired Cooperative Strategies For Optimization*. Berlin, Heidelberg: Springer, 2010. 65–74.
- 21 陈志敏, 田梦楚, 吴盘龙, 等. 基于蝙蝠算法的粒子滤波法研究. *物理学报*, 2017, 66(5): 050502.
- 22 Neumin OS, Zhuk SY. Sequential detection of target trajectory tracking loss using the decision statistics of pips. *Radioelectronics and Communications Systems*, 2014, 57(8): 352–361. [doi: [10.3103/S0735272714080032](https://doi.org/10.3103/S0735272714080032)]
- 23 Son HS, Park JB, Joo YH. Segmentalized FCM-based tracking algorithm for zigzag maneuvering target. *International Journal of Control, Automation and Systems*, 2015, 13(1): 231–237. [doi: [10.1007/s12555-013-0406-0](https://doi.org/10.1007/s12555-013-0406-0)]
- 24 Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(4): 341–359. [doi: [10.1023/A:1008202821328](https://doi.org/10.1023/A:1008202821328)]