

高置信度的自适应特征融合跟踪研究

刘万军¹, 李怡彤^{2*}, 姜文涛¹

¹辽宁工程技术大学软件学院, 辽宁 葫芦岛 125105;

²辽宁工程技术大学研究生院, 辽宁 葫芦岛 125105

摘要 针对互补学习跟踪算法遮挡、背景复杂等场景下出现跟踪漂移的问题,提出了高置信度的自适应特征融合目标跟踪算法。首先,使用巴氏系数实时计算每一帧前景与背景颜色直方图的相似度并利用对数损失函数求得最终融合因子,实现每一帧更优的特征融合;其次,将平均峰值相关能量和响应峰值与其对应历史均值的比值作为置信度判定参数,根据判定结果决定是否更新和纠正目标位置进行跟踪。在 OTB100 和 LaSOT 数据集上进行实验,提出算法与 Staple 算法相比精确率和成功率分别提高了 17.5%、15.4% 和 27.3%、18.0%,该结果表明了该算法的有效性和鲁棒性。

关键词 图像处理; 目标跟踪; 相关滤波; 特征融合; 巴氏系数; 置信度

中图分类号 TP391 文献标志码 A

DOI: 10.3788/LOP202259.2210003

Research on High-Confidence Adaptive Feature Fusion Tracking

Liu Wanjun¹, Li Yitong^{2*}, Jiang Wentao¹

¹College of Software, Liaoning Technical University, Huludao 125105, Liaoning, China;

²Graduate School, Liaoning Technical University, Huludao 125105, Liaoning, China

Abstract A high-confidence adaptive feature fusion target tracking algorithm is proposed to address the problem of tracking drift in the complex scenes, such as occlusion and complex backgrounds, using a complementary learning tracking algorithm. First, we use the Bhattacharyya coefficient to calculate the similarity between the foreground and background color histograms of each frame in real time, and adopt the log loss function to obtain the final fusion factor to achieve better feature fusion of each frame. The average peak correlation energy and the ratio of the response peak value to its corresponding historical average value are then used to determine confidence determination parameter, and the target position is updated and corrected for tracking based on the determination result. Experiments on the OTB100 and LaSOT datasets show that this algorithm improves the precision rate by 17.5% and 15.4%, and success rate by 27.3% and 18.0%, respectively when compared with the Staple algorithm. The results demonstrate the effectiveness and robustness of the algorithm.

Key words image processing; target tracking; correlation filtering; feature fusion; Bhattacharyya coefficient; confidence

1 引言

视觉目标跟踪技术作为计算机视觉领域一个重要的研究课题,在群体智能、安全监视和现代军事等领域有着广泛的应用^[1-3]。目标跟踪算法指通过第一帧给定目标的位置和大小等初始信息,在后续的视频图像序列中根据算法实时分析目标的位置等信息,实现对目标的准确跟踪。目标跟踪算法经过不断的研究与改进,已取得了一定进展。由于在跟踪的过程中会出现

多种多样的干扰因素,包括遮挡、光照变化、平面内/外旋转等,使得目标跟踪技术存在许多难题需要解决^[4-6]。

相关滤波首次被应用在视觉跟踪领域是 Bolme 等^[7]提出的最小输出平方和误差(MOSSE)跟踪算法,使用快速傅里叶变换将计算从时域转换到频域,减少了运算量,其运行速度可以达到 615 frame/s 的高速,体现了相关滤波的潜力,但该算法采用单通道的灰度特征导致跟踪效果不够理想。Henriques 等^[8]提出了

收稿日期: 2021-07-20; 修回日期: 2021-08-30; 录用日期: 2021-09-28

基金项目: 国家自然科学基金(61172144)、辽宁省自然科学基金(20170540426)、辽宁省教育厅基金(LJYL049, LJ2017QL034, LJ2017ZL003)

通信作者: *362685037@qq.com

核相关滤波(KCF)算法,该算法将MOSSE算法使用的单通道灰度特征替换为多通道方向梯度直方图^[9](HOG)特征,并通过引入核函数将非线性问题转化为高维空间中的线性问题,使算法得到优化。Bertinetto等^[10]将颜色直方图模型和相关滤波器模型相结合提出了Staple算法,该算法将HOG特征与颜色直方图特征相融合,提高了算法跟踪精度的同时也满足了实时跟踪的需求。

但Staple算法的特征融合因子是固定不变的,当两种模型出现模型污染时仍以该固定不变的融合因子进行融合可能会导致跟踪漂移,逐帧累积最终可能会丢失目标。另外,模型更新为逐帧更新过于单一,算法容易出现跟踪漂移。

针对以上问题,本文提出高置信度的自适应特征(HCAF)融合目标跟踪算法。首先,使用巴氏系数和对数损失函数控制颜色直方图模型的融合因子,实现相关滤波器模型和颜色直方图模型的自适应融合,缓解了颜色直方图模型污染导致的跟踪漂移问题。其次,使用双重置信度判定实现高置信度形态以增强更新的有效性和鲁棒性,实现了更佳模型更新策略。

2 HCAF融合跟踪算法原理

2.1 互补学习跟踪算法

相关滤波器模型由平移滤波器 h_t 和尺度滤波器 h_s 组成。在训练阶段,滤波器 h_t 学习目标的HOG特征,其优化求解可转化为 d 通道特征图像 m 下的岭回归问题,可表示为

$$h_t = \arg \min_h \left\| \sum_{i=1}^d h_i * m^i - g \right\|^2 + \lambda \sum_{i=1}^d \|h_i\|^2, \quad (1)$$

式中:“*”表示卷积算子; $l \in \{0, 1, \dots, d\}$ 为特征通道索引; g 为高斯理想输出; $\lambda \geq 0$ 为正则项参数,正则项起到防止过度拟合的作用。

该模型以固定学习率 η_{templ} 更新相关滤波器模型,可表示为

$$F_t = (1 - \eta_{\text{templ}})F_{t-1} + \eta_{\text{templ}}F_t, \quad (2)$$

式中, F_t 和 F_{t-1} 分别表示第 t 帧和第 $t-1$ 帧得到的相关滤波模型。

颜色直方图模型每帧确定跟踪目标位置后,采用最小化损失函数的方式更新模型。该模型的损失函数可表示为

$$L(\beta, x) = \frac{1}{|O|} \sum_{x \in O} [\beta^\top \Phi(x) - 1]^2 + \frac{1}{|B|} \sum_{x \in B} [\beta^\top \Phi(x)]^2, \quad (3)$$

式中: O 和 B 分别为目标区域和背景区域; β 为模型参数; $\Phi(x)$ 为第 x 帧中像素网格内统计颜色特征的函数。对于参数 β ,转化为解决相关岭回归问题,可表示为

$$\beta_t = \arg \min_{\beta} \left\{ L(\beta, x) + \frac{1}{2} \lambda \|\beta\|^2 \right\}. \quad (4)$$

该模型在线更新函数可表示为

$$\begin{cases} \rho_t(O) = (1 - \eta_{\text{hist}})\rho_{t-1}(O) + \eta_{\text{hist}}\rho'_t(O) \\ \rho_t(B) = (1 - \eta_{\text{hist}})\rho_{t-1}(B) + \eta_{\text{hist}}\rho'_t(B) \end{cases}, \quad (5)$$

式中: $\rho'_t(O)$ 表示第 t 帧得到的目标区域直方图得分; $\rho'_t(B)$ 表示第 t 帧得到的背景区域直方图得分; η_{hist} 为固定颜色直方图模型学习率。

将相关滤波器模型和颜色直方图模型进行融合,可表示为

$$f(x) = \gamma_{\text{templ}}f_{\text{templ}}(x) + \gamma_{\text{hist}}f_{\text{hist}}(x), \quad (6)$$

式中: $f_{\text{templ}}(x)$ 和 $f_{\text{hist}}(x)$ 分别为使用两种模型对各自候选样本检测得到的响应图; γ_{templ} 和 γ_{hist} 分别为两种模型的融合系数。

2.2 自适应特征融合

颜色直方图模型主要用于将目标的前景与背景分开,通过统计前一帧前景和背景的颜色直方图,使用贝叶斯分类器将目标从背景中分离出来。表示为

$$P(x \in O|O, B) = \frac{P(b_x|x \in O)P(x \in O)}{\sum_{\Omega \in \{O, B\}} P(b_x|x \in \Omega)P(x \in \Omega)} = \frac{P(b_x|x \in O)P(x \in O)}{P(b_x|x \in O)P(x \in O) + P(b_x|x \in B)P(x \in B)}, \quad (7)$$

式中: b_x 为颜色区间; $H_O(b_x) \approx P(b_x|x \in O)/|O|$ 和 $H_B(b_x) \approx P(b_x|x \in B)/|B|$ 分别为像素 x 属于前景区域和背景区域的先验概率; $P(b_x|x \in O)$ 和 $P(b_x|x \in B)$ 分别为前景直方图和背景直方图中像素 x 所属直方柱的像素总数。在某一帧中,直方图上颜色区间为 b_x 的像素属于前景区域 O 的概率表示为

$$P(x \in O|O, B) = \frac{H_O(b_x)}{H_O(b_x) + H_B(b_x)}. \quad (8)$$

使用积分图的方法计算像素级颜色概率图中概率之和,并赋值给矩形框的中心位置,从而得到颜色直方图模型的响应值,再与相关滤波模型的响应值进行融合。

不同状态下的跟踪图像及对应的前景和背景颜色概率如图1所示。图1(a)为正常跟踪时,可以看到对应的前景像素级颜色概率图中高亮点区域集中;图1(b)为受到相似颜色物体干扰时,对应的前景像素级颜色概率图中高亮点区域增多,这时颜色直方图模型会受到该相似颜色物体的影响,本应属于背景区域的像素会被识别到前景区域中导致先验概率出现偏差,求得的颜色直方图模型的响应值也会出错,若仍按原本的融合参数与相关滤波模型的响应值进行融合,就会产生模型污染并导致跟踪漂移。

本文使用巴氏系数算法求得前景颜色直方图与背景颜色直方图的相似度。图像分为彩色图像和灰色图

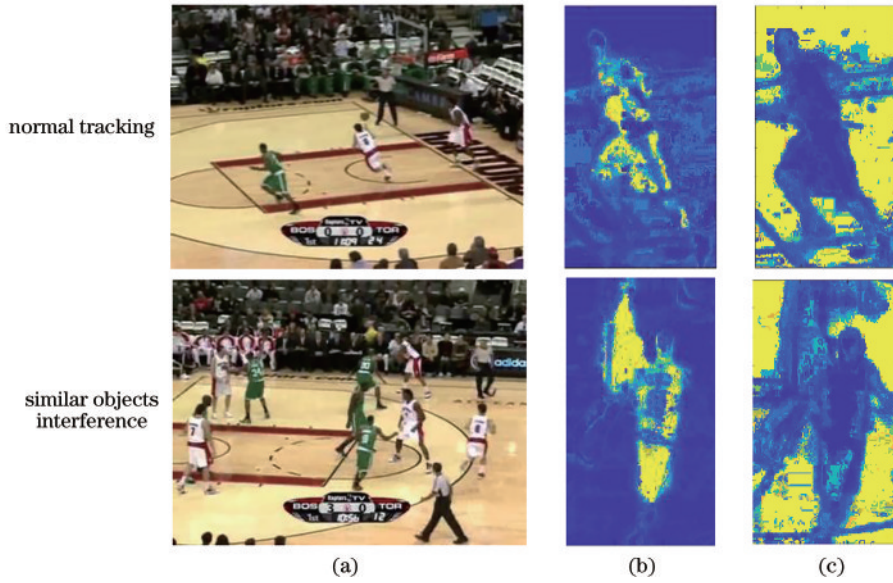


图 1 不同状态下的跟踪图像及对应的前景和背景颜色概率图。(a)原始图像;(b)前景颜色概率图;(c)背景颜色概率图
Fig. 1 Tracking images in different states and corresponding foreground and background color probability maps. (a) Original image; (b) foreground color probability map; (c) background color probability map

像,前景颜色直方图和背景颜色直方图的相似度分别表示为

$$\begin{cases} B_{C_gray} = \sum \sqrt{b_{g_hist(i,1)} * f_{g_hist(i,1)}} \\ B_{C_rgb} = \sum \sqrt{b_{g_hist(x,y,z)} * f_{g_hist(x,y,z)}} \end{cases}, \quad (9)$$

式中, $b_{g_hist(i,1)}$ 、 $f_{g_hist(i,1)}$ 为灰色图像中的概率分布样本; $b_{g_hist(x,y,z)}$ 、 $f_{g_hist(x,y,z)}$ 为彩色图像中的概率分布样本。将 B_{C_gray} 、 B_{C_rgb} 统称为 $B_{C,s}$, 即

$$B_{C,s} = \begin{cases} B_{C_gray}, \text{ gray image} \\ B_{C_rgb}, \text{ color image} \end{cases}. \quad (10)$$

首先对相同位置的概率相乘,其次进行开方,然后再对开方后结果进行累加。如果前景或背景在该处概率分布为 0,则乘积结果也为 0,表示在该处完全不同。如果前景和背景完全相同,那么在直方图上概率分布完全相同,累加结果必定为 1。在目标跟踪过程中,当 $B_{C,s}$ 值很小时说明前景与背景相差很多,这时颜色直方图特征适用性较高,当 $B_{C,s}$ 值很大时说明背景区域出现与目标相似的物体,这时颜色直方图特征适用性较低。本文使用对数损失函数得到最优的 $B_{C,s}$ 值,该函数式可表示为

$$Y = a - \log(B_{C,s}), \quad (11)$$

式中, \log 函数为单调递增,当 $B_{C,s}$ 为大值时 $\log B_{C,s}$ 也会得到大值,加上负号之后大值会变成小值。故当 $B_{C,s}$ 值很大时,通过该函数将得到对应的小值。相对地,当 $B_{C,s}$ 值小时会得到大值。

通过设定不同的 a 值,使融合出的特征具有不同的特征偏向性,以对应变化的目标状态及背景信息。在 OTB100 数据集进行大量实验得到当 a 取值比在 0.05509 上下浮动超过 0.00003 时,会使得颜色直方图特征的融合系数变化过于强烈,融合效果不佳导致跟

踪效果较差。因此本文选择实验效果最优的 $a = 0.05509$ 来实时改变 $B_{C,s}$ 值,对应的对数损失函数 Y 可表示为

$$Y = a - \log(B_{C,s}) = 0.05509 - \log(B_{C,s}). \quad (12)$$

对数损失函数 Y 如图 2 所示。

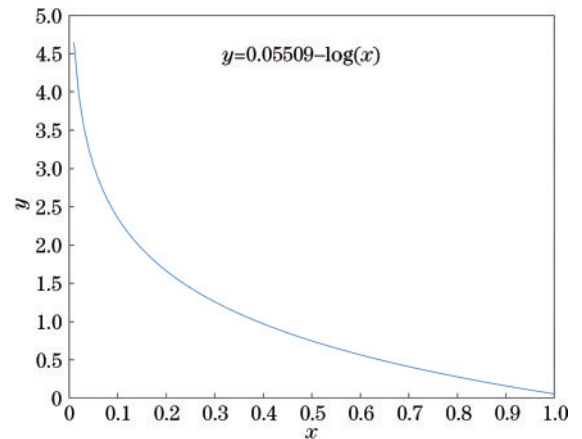


图 2 对数损失函数图

Fig. 2 Logarithmic loss function graph

将颜色直方图模型的融合系数 γ_{hist} 记为融合因子 α ,则相关滤波器模型融合系数 $\gamma_{tmpl} = 1 - \alpha$,并将 $B_{C,s}$ 赋值给 α 。最终的算法融合可表示为

$$f(x) = (1 - \alpha)f_{tmpl}(x) + \alpha f_{hist}(x). \quad (13)$$

部分框架图如图 3 所示。

由图 3 可以看到, α 随着每一帧前景颜色直方图和背景颜色直方图的相似度得分变化而变化,当出现相似物体干扰时函数式 Y 将 α 值减小,缓解了颜色直方图模型污染导致的算法跟踪漂移。

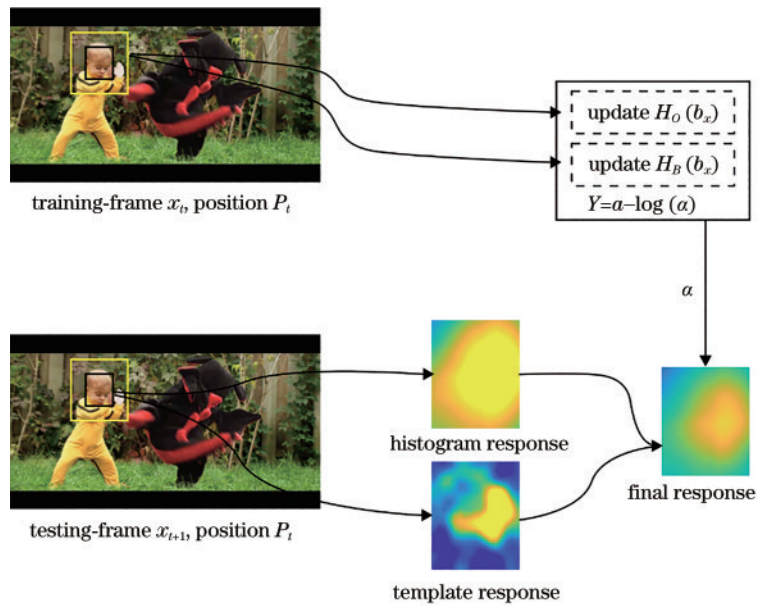


图 3 部分跟踪框架图

Fig. 3 Partial tracking framework

2.3 高置信度相关滤波器模型更新策略

相关滤波器模型在检测阶段,用新一帧的待测特征图像块 z 和 h_t 进行相关运算得到响应值,响应值越大越可能是目标的位置,响应图可表示为

$$R = h_t * z. \quad (14)$$

跟踪状态及响应结果如图 4 所示。图 4(a)、4(b) 为正常跟踪时,响应图的峰值就是目标的位置。但当跟踪目标被遮挡或出现背景复杂等干扰时,按原本模型更新策略更新会逐渐偏离正在跟踪的目标,导致跟踪失败。图 4(c)、4(d) 为出现干扰因素时,响应图剧

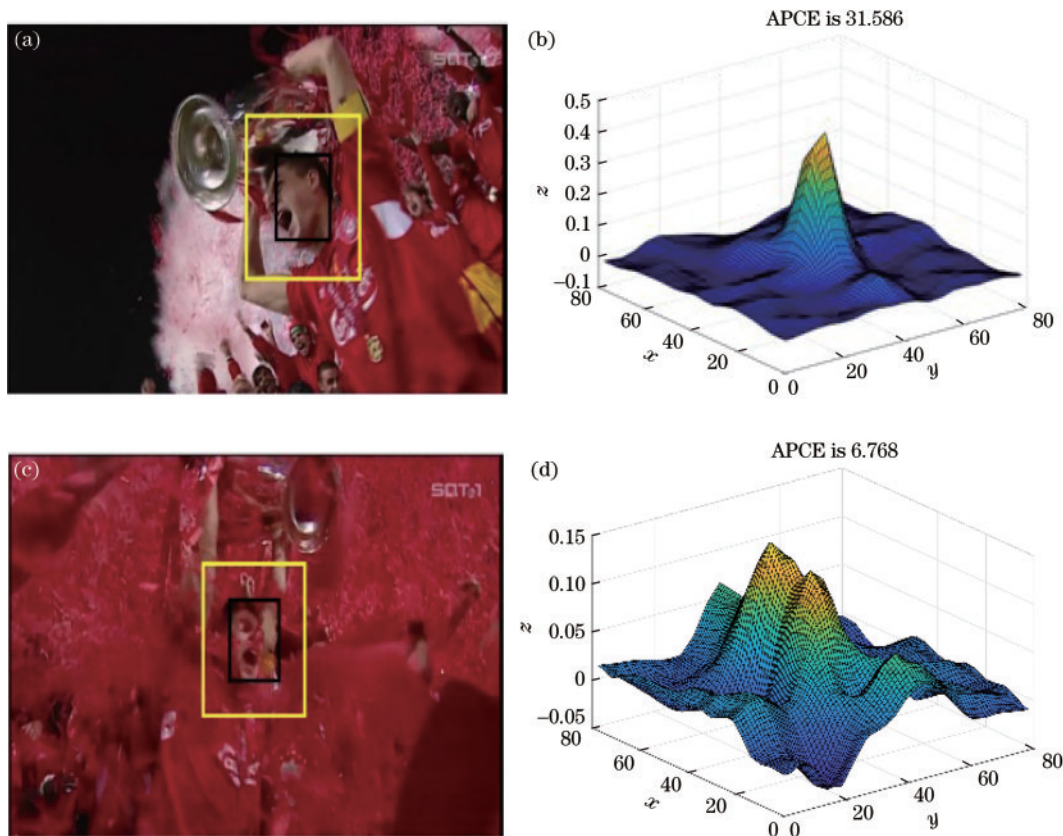


图 4 跟踪状态及响应结果图。(a)正常跟踪;(b)正常跟踪响应图;(c)背景复杂;(d)背景复杂响应图

Fig. 4 Target and response result graph. (a) Normal tracking; (b) response map under normal tracking; (c) background clutter; (d) response map under background clutter

烈震荡出现多个峰值,如果继续将响应值最高处作为目标所在位置,很可能出现跟踪错误,特别是原算法中的颜色特征很容易受到复杂环境因素的影响加重漂移。

在第 t 帧时的响应结果图 R 对应的平均峰值相关能量(APCE)^[11]表示为

$$A_{\text{PCE}t} = \frac{|R_t^{\max} - R_t^{\min}|^2}{\text{mean}\left[\sum (R_t - R_t^{\min})^2\right]}, \quad (15)$$

式中: R_t^{\max} 表示第 t 帧响应结果图中的响应最大值; R_t^{\min} 表示第 t 帧响应结果图中的响应最小值; R_t 表示第 t 帧目标的响应值;mean()表示求平均值。

从图 4 可以看到当出现跟踪干扰或目标丢失时,当前帧的 $A_{\text{PCE}t}$ 和响应峰值 R_{\max} 的值相比于上一帧减小很多,选择此时停止对模型进行更新实现减小模型漂移的概率。

仅通过比较 $A_{\text{PCE}t}$ 、响应峰值 R_{\max} 与其分别对应的上一帧的值难以全面分析目标状态情况,具有一定的局限性,尤其是当目标被相似的物体遮挡时,峰值震荡程度较小容易造成错误判断。本算法根据目标运动具有连续渐变的特性,选择通过判断 4 个变量的对比结果对这一帧的响应结果进行判别。引入的两个变量分别表示为

$$R_{\text{mean}} = \text{mean}\left(\sum R_t^{\max} + R_t^{\min}\right), \quad (16)$$

$$A_{\text{PCEmean}} = \text{mean}\left(\sum A_{\text{PCE}t-1} + A_{\text{PCE}t}\right), \quad (17)$$

式中: R_t 为第 t 帧的响应峰值; R_{mean} 为响应峰值历史均值; A_{PCEmean} 为 APCE 的历史均值。将 APCE 和响应峰值与其对应的历史均值的比例作为置信度判定参数,表示为

$$B_1 = A_{\text{PCEmean}}/A_{\text{PCE}t}, \quad (18)$$

$$B_2 = R_{\text{mean}}/R_t, \quad (19)$$

式中, B_1 和 B_2 分别为当前帧的响应峰值和 APCE 与其对应历史均值的比值。当 B_1 、 B_2 达到一定值时,此时置信度判定为不可靠,模型将停止更新,缓解了因不考虑出现目标遮挡、形变等情况时继续按照固定的模型更新频率进行模型更新而导致的跟踪漂移问题。

2.4 Kalman 滤波预测目标位置

在目标跟踪中,由于目标在相邻帧的间隔通常为毫秒量级,在如此短的时间内可以认为目标的状态保持稳定,本文通过使用 Kalman 滤波^[12]对目标在短时间内的状态进行有效估计,用于在目标跟踪出现波动时修正跟踪目标位置,当 B_1 、 B_2 达到一定值时,将 Kalman 滤波得到的预测位置返回跟踪过程中进行跟踪,能够在目标出现严重遮挡、光照变化等情况导致跟踪丢失时,重新找到目标位置继续跟踪。

首先给出 $\mathbf{X} = [x, y, v_x, v_y]$ 目标的初始状态,其中 (x, y) 为目标位置, (v_x, v_y) 为目标运动速度,给状态转移矩阵 \mathbf{A} 、状态观测矩阵 \mathbf{H} 、预测噪声协方差 \mathbf{Q} 、观测

噪声协方差 \mathbf{R} 赋初值为

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

$$\mathbf{Q} = \begin{pmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{pmatrix}, \mathbf{R} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

之后每一帧都通过方程

$$\mathbf{Z} = \mathbf{H}\mathbf{X} + \mathbf{V}, \quad (20)$$

得到一个观测矩阵 \mathbf{Z} , 开始进行 Kalman 滤波 5 个方程的更新, 5 个方程分别为

$$\mathbf{X}_- = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}, \quad (21)$$

式中: \mathbf{X} 表示当前帧的状态; \mathbf{X}_- 表示下一帧的预测状态值; \mathbf{U} 表示这一帧的控制量,本文算法没有控制量,故 \mathbf{U} 设为 0;

$$\mathbf{P}_- = \mathbf{A}\mathbf{P}\mathbf{A}' + \mathbf{Q}, \quad (22)$$

式中: \mathbf{P} 表示协方差矩阵; \mathbf{P}_- 表示下一帧预测的协方差矩阵;

$$\mathbf{K} = \mathbf{P}_- * \mathbf{H}' / (\mathbf{H} * \mathbf{P}_- * \mathbf{H}' + \mathbf{R}), \quad (23)$$

式中, \mathbf{K} 表示 Kalman 增益;

$$\mathbf{X} = \mathbf{X}_- * \mathbf{K} (\mathbf{Z} - \mathbf{H} * \mathbf{X}), \quad (24)$$

式中, \mathbf{X} 表示当前时刻估算值;

$$\mathbf{P} = \mathbf{P}_- - \mathbf{K} * \mathbf{H} * \mathbf{P}_-. \quad (25)$$

通过 5 个方程的计算得到下一帧目标的预测位置,当达到一定阈值时说明按原本模型得到的位置很可能是错误的,这时就要使用 Kalman 滤波得到的目标预测位置进行下一步的跟踪。

2.5 算法步骤

HCAF 算法步骤如下:

1) 初始帧目标定位和建模。手动确定初始帧目标中心位置,确定相关滤波器模型、尺度滤波器模型和颜色直方图模型。

2) 模型训练。对相关滤波器和颜色直方图模型进行样本采集并训练和测试。

3) 自适应融合及目标定位。计算前景和背景的颜色直方图相似度,得到最优的融合因子,代入融合公式得到最终的融合特征得分,实现两个模型的自适应融合。

4) 置信度判别及模型更新。当判定为极不可靠时使用当前位置进行模型更新;当 B_1 、 B_2 与 b_1 、 b_2 相比不满足条件时进行模型更新并输出结果,继续执行步骤 2);当 B_1 、 B_2 与 b_1 、 b_2 相比判定为不可靠且 B_1 、 B_2 与 b_3 、 b_4 相比不满足条件时,执行步骤 5)。

5) 位置预测。使用 Kalman 滤波进行目标位置预测,以预测的位置为中心开始执行步骤 2)。

HCAF 算法框架如图 5 所示。

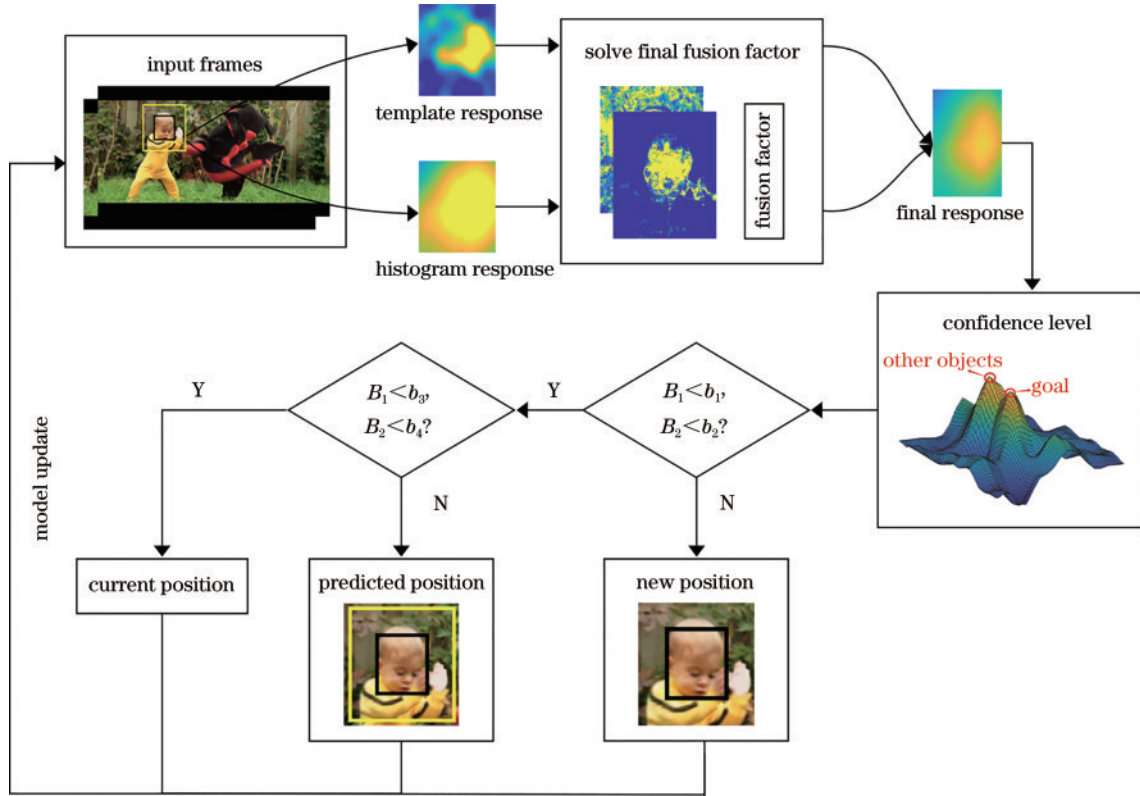


图 5 HCAF 算法框架示意图

Fig. 5 Schematic diagram of HCAF algorithm framework

3 实验结果分析

3.1 实验环境及参数配置

本文算法的实现均在 Intel® Core™ i7-9750H 2.60 GHZ CPU 和 16 GB 内存的设备上的 GNU Octave 进行的。算法参数配置见表 1。

表 1 参数配置

Table 1 Parameters configuration

Parameter	Parameter value
bg and fg color models learning rate	0.04
HOG model learning rate	0.01
Scale learning rate	0.025

3.2 评价指标

本文采用一次通过评估模式(OPE),从距离精度和成功率对跟踪性能进行评价。距离精度指目标中心位置误差小于 20 pixel 的帧数占视频序列总帧数的比例。精确率曲线表示为

$$A_{\text{precision}} = \frac{M \left[\sqrt{(x - x_t)^2 + (y - y_t)^2} \right]}{N}, \quad (26)$$

式中: (x, y) 表示算法预测的目标中心位置; (x_t, y_t) 表示真实的目标中心位置; M 表示满足条件的帧数; N 表示视频序列总帧数。成功率指的是算法的成功率与坐标轴所围成的面积, A_m 表示算法的标注框, A_n 表示人

工给出的标注框,成功率曲线可表示为

$$S = \frac{|A_m \cap A_n|}{|A_m \cup A_n|}. \quad (27)$$

3.3 本文算法参数分析

本文算法新增参数有函数式中的常数 a , 置信度判定阈值 b_1, b_2 和 b_3, b_4 。参数设置不同值的对比结果见表 2。

表 2 参数设置不同值的对比结果

Table 2 Comparison results of different parameter settings

a	b_1, b_2	b_3, b_4	Precision
0.05470	0.4795, 0.2794	0.21, 0.11	0.718
0.06833	0.4795, 0.2794	0.21, 0.11	0.706
0.05509	0.4710, 0.2710	0.21, 0.11	0.734
0.05509	0.4870, 0.2870	0.21, 0.11	0.721
0.05509	0.4795, 0.2794	0.15, 0.05	0.788
0.05509	0.4795, 0.2794	0.28, 0.18	0.792
0.05509	0.4795, 0.2794	0.21, 0.11	0.854

实验证明,当置信度参数 b_1, b_2, b_3, b_4 取值稍大时使置信度判断不够敏感,模型更新过于缓慢,导致目标跟踪出现漂移;当置信度参数 b_1, b_2, b_3, b_4 取值稍小时使置信度判断过于敏感,模型更新达不到想要的效果,导致跟踪漂移。当常数 a 取 0.05509, b_1, b_2 取值为 0.4795、0.2794 和 b_3, b_4 取值为 0.21、0.11 时跟踪效果最好。

3.4 定量分析

为了验证算法的优势与不足,将10种跟踪算法进行跟踪实验对比,10种算法包括本文HCAF算法、AutoTrack^[13]、ACFT^[14]、DeepSTRCF^[15]、ECO^[16]、BACF^[17]、Staple、Staple_CA^[18]、SRDCF^[19]、DSST^[20]。

在OTB100^[21]数据集中遮挡属性的精确率和成功率如图6所示。在OTB100数据集中背景复杂属性的精确率和成功率如图7所示。在OTB100数据集上的精确率和成功率如图8所示。

由图6可得,本文算法遮挡属性的精确率为0.808

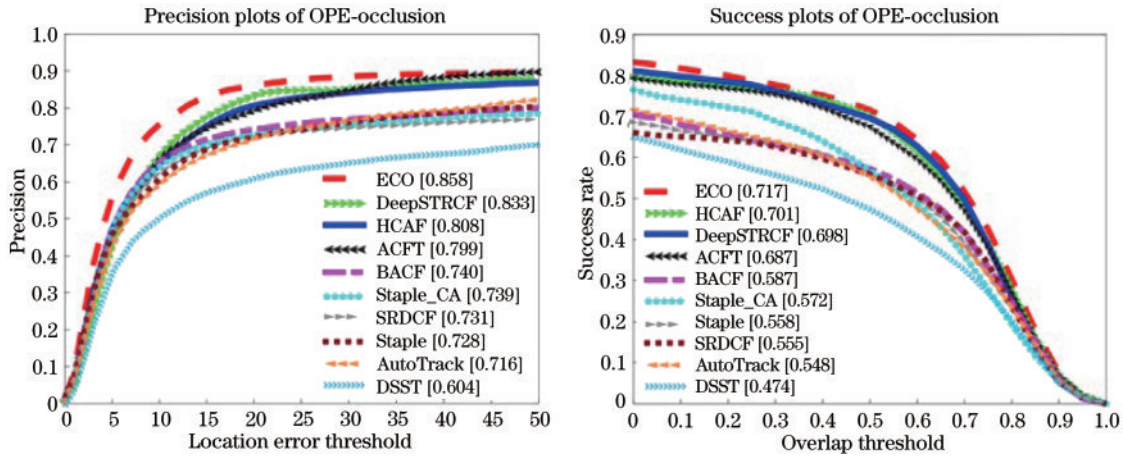


图6 在OTB100数据集中遮挡属性的精确率和成功率
Fig. 6 Precision and success rates of occlusion attributes on OTB100 dataset

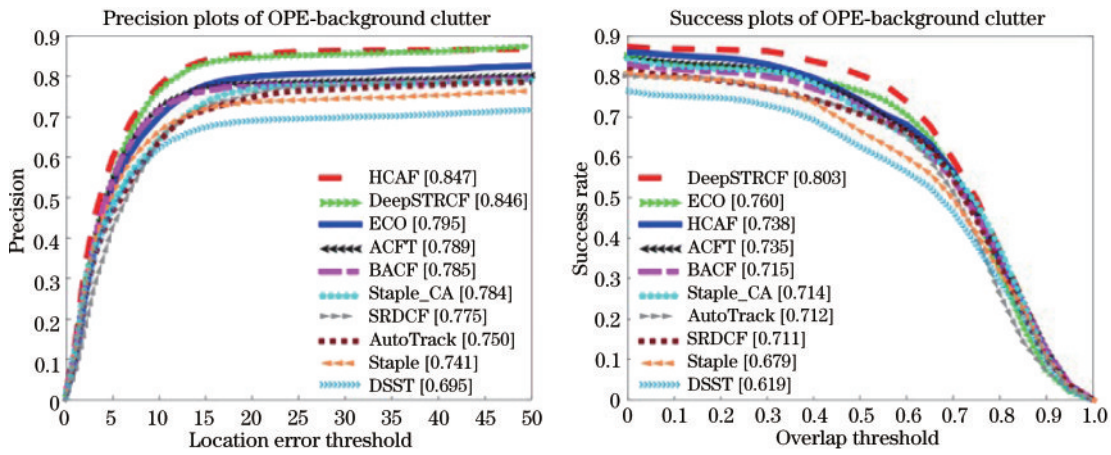


图7 在OTB100数据集中背景复杂属性的精确率和成功率
Fig. 7 Precision and success rates of background clutter attributes on OTB100 dataset

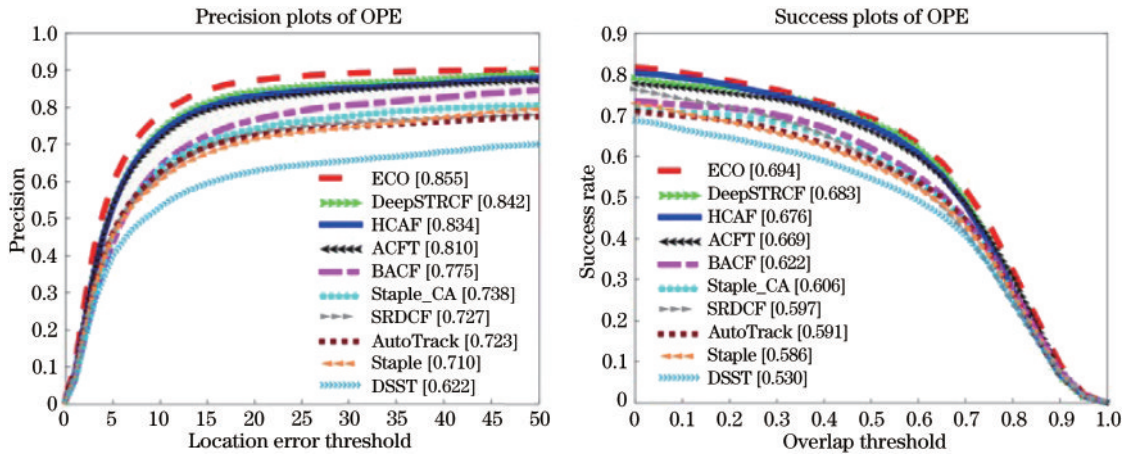


图8 在OTB100数据集上的精确率和成功率
Fig. 8 Precision and success rates on OTB100 dataset

和成功率为 0.701,与 Staple 算法遮挡属性的精确率 0.728 和成功率 0.558 相比,分别提高了 11.0% 和 25.6%。由图 7 可得,本文算法背景复杂属性的精确率为 0.847 和成功率为 0.738,与 Staple 算法背景复杂属性的精确率 0.741 和成功率 0.679 相比,分别提高了 14.3% 和 8.7%。由图 8 可得,本文算法的精确率为 0.834 和成功率为 0.676,与 Staple 算法的精确率

0.710 和成功率 0.586 相比,分别提高了 17.5% 和 15.4%。

由于数据集 OTB100 视频序列集中超过 1500 帧的视频序列较少,不能充分体现本文算法对于长时间目标跟踪的精确率,本文引入最新的数据集 LaSOT^[22],与当前主流跟踪算法进行对比测试。在 LaSOT 数据集上的精确率和成功率如图 9 所示。

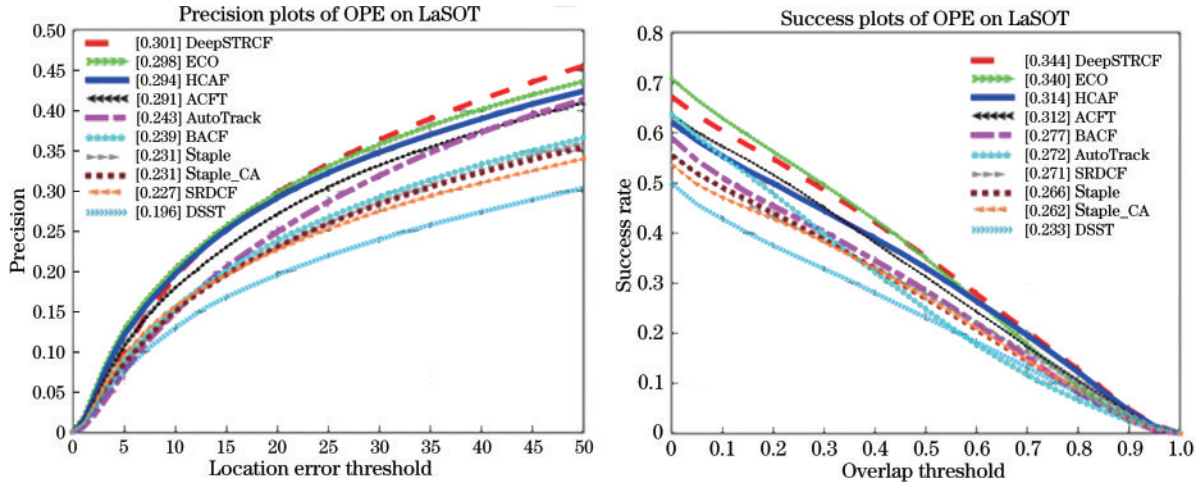


图 9 在 LaSOT 数据集上的精确率和成功率

Fig. 9 Precision and success rates on LaSOT dataset

与数据集 OTB100 相比,数据集 LaSOT 视频序列数量更多达到 1400 个,在视频序列中的图像帧数很多,平均帧长超过 2500 帧,种类划分更细致,挑战难度更大。从图 9 可以看到,本文算法 HCAF 的精确率达到 0.294,成功率达到 0.314,在基于 CF 的算法中具有

较好的跟踪结果。另外,Staple 算法的精确率和成功率分别为 0.231 和 0.266,与其相比本文算法分别提高了 27.3% 和 18.0%。

OTB100 和 LaSOT 数据集上的速率对比见表 3 (其中黑体为最佳,斜体为次佳)。

表 3 OTB100 和 LaSOT 数据集上的速率对比

Table 3 Speed comparison on OTB100 and LaSOT datasets

unit: frame/s

Dataset	HCAF	AutoTrack	ACFT	DeepSTRCF	ECO	BACF	Staple	Staple_CA	SRDCF	DSST
OTB100	77.8	20.2	74.1	29.8	17.9	26.7	76.6	72.4	5.8	24.47
LaSOT	67.2	14.2	66.5	20.4	9.3	17.6	65.1	63.7	2.8	15.47

从以上实验对比结果可以看出 HCAF 算法跟踪精度不如 ECO 和 DeepSTRCF 高,但是结合表 3 可以看出本文算法的跟踪速度在 OTB100 数据集上可以达到 77.8 frame/s、在 LaSOT 数据集上可以达到 67.2 frame/s,位居第一,而 ECO 算法和 DeepSTRCF 算法在两个数据集上速率分别为 17.9、9.3 frame/s 和 29.8、20.4 frame/s,说明了置信度控制模型更新策略有效提高了算法的跟踪速率。

另外,可以看出本文算法在遮挡和背景复杂属性上的跟踪精度均高于 Staple 算法,说明控制颜色直方图模型融合参数提高了算法处理背景复杂属性的能力以及置信度判定环节提高了算法处理遮挡属性的能力,实现了算法跟踪精度的提高。整体说明了本文算法既可以达到较好的跟踪精度又具有良好的跟踪速率,具有有效性和鲁棒性的同时也更加适用于实时跟踪。

3.5 定性分析

从 OTB100 数据集中选出 4 个具有挑战性的视频序列,包括 Basketball、Human3、Jogging-1、Soccer,将本文 HCAF 算法与 AutoTrack、ACFT、DeepSTRCF、ECO、BACF、Staple_CA、Staple、SRDCF、DSST 这 9 种算法进行定性比较实验。每个视频序列都有 5 种以上的跟踪干扰属性,对于验证算法性能极具说服力。10 种算法在部分视频序列上的跟踪结果如图 10 所示。

图 10(a)为视频序列 Basketball,该序列包含遮挡、形变等视频干扰属性。从 24 帧可得,跟踪目标被遮挡后部分算法跟踪漂移,本文算法跟踪效果较好。189 帧 ECO 算法因目标形变出现轻微漂移。500 帧时 ECO 因目标被遮挡导致跟踪失败,其余算法都可跟踪到目标,从 680 帧可以看到本文算法跟踪效果良好,而

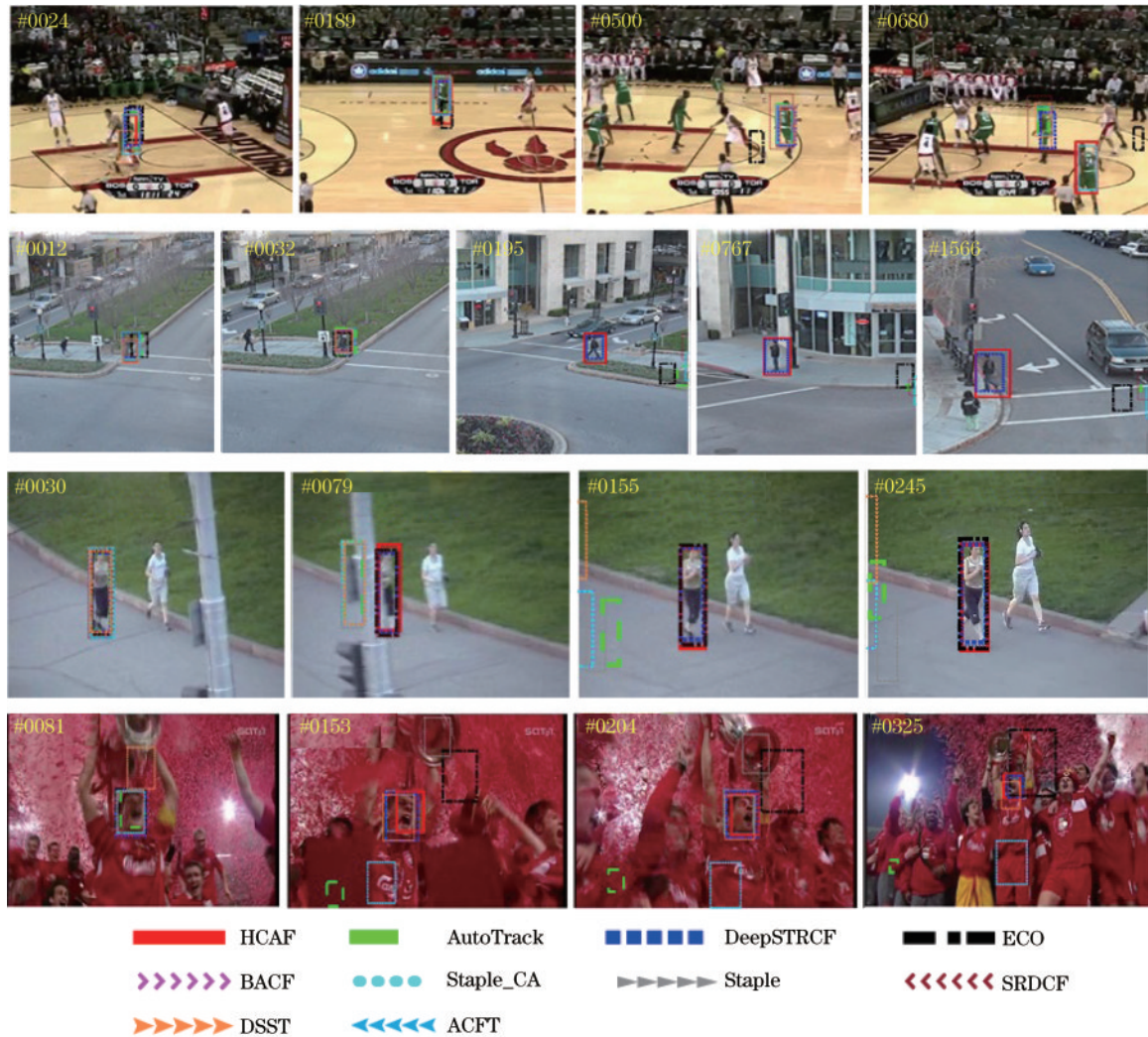


图 10 10 种算法在部分视频序列上的跟踪结果。(a) Basketball;(b) Human3;(c) Jogging-1;(d) Soccer

Fig. 10 Tracking results of 10 tracking algorithms in partial sequences. (a) Basketball; (b) Human3; (c) Jogging-1; (d) Soccer

Staple 算法因出现相似物体干扰而丢失跟踪目标。由此可见针对 Staple 算法中颜色直方图模型污染导致跟踪漂移问题而改进的自适应融合起到作用,本文算法在相似物体干扰的情形下跟踪更加准确。

图 10(b) 为视频序列 Human3, 该序列包含尺度变化、遮挡等视频干扰属性。从 12 帧可以看到背景开始杂乱时, Staple、ECO、DSST 算法出现跟踪漂移。从 195 帧可以看到在目标经过多次遮挡后本文算法、ACFT 和 DeepSTRCF 算法可以准确跟踪到目标。从 767 帧和 1566 帧可以看到, 目标出现遮挡、背景复杂等多种干扰情况, 本文算法和 DeepSTRCF 两个算法跟踪依然稳定。由此可见本文算法对于遮挡和背景复杂的处理更加稳定。

图 10(c) 为视频序列 Jogging-1, 该序列包含遮挡和形变等视频干扰属性。从 79 帧可以看到, 目标被完全遮挡后, AutoTrack、BACF、Staple_CA、Staple、DSST 均跟踪失败, 本文算法可以准确捕捉到目标继续进行跟踪。由此可见本文算法引入的遮挡判别和位置预测均起到有效作用。

图 10(d) 为视频序列 Soccer, 该序列包含众多干扰因素, 快速运动、运动模糊和背景复杂等。从 153 帧可以看到, 目标出现遮挡和背景复杂干扰, AutoTrack、Staple、ECO、Staple_CA 均出现跟踪丢失, 本文算法正常跟踪。从 204 帧可以看到, 除已经跟踪失败的 4 个算法外均能较好地处理快速运动和运动模糊问题。在 325 帧, 目标发生剧烈尺度变化, 本文算法能够很好地适应此变化做到准确跟踪, 与其他跟踪到目标的算法相比本文算法跟踪效果更好。在众多干扰因素下, 本文算法表现良好, 说明本文算法达到了更加稳定的跟踪目的。

从图 10 可以看出, 在 4 个视频序列 11 种视频属性都包含的情况下, 本文算法与当前主流算法相比表现较好, 在某些特定情况下比主流算法性能更优。

4 结 论

针对 Staple 算法在遮挡、相似物体干扰等复杂场景下跟踪漂移的问题提出了一种高置信度下的自适应特征融合的实时目标跟踪算法, 做了以下工作: 1) 利用

巴氏系数,求得每一帧前景颜色直方图与背景颜色直方图的相似度,通过对数损失函数求得最终融合因子,实现自适应特征融合的同时缓解了颜色直方图模型对整体算法造成的模型污染,提高了算法处理相似物体干扰的能力;2)加入置信度判别环节,将 APCE 和响应峰值与其对应历史均值的比值作为置信度判定参数,实现更优的模型更新策略,缓解模型退化,提高算法跟踪速率;3)使用 Kalman 滤波实时预测跟踪目标位置,进一步提高了算法处理复杂环境的能力。由 OTB100 数据集上的实验结果可得,提出算法与 Staple 算法相比精确率和成功率分别提高了 17.5% 和 15.4%。由 LaSOT 数据集上的实验结果可得,提出算法与 Staple 算法相比精确率和成功率分别提高了 27.3% 和 18.0%。由此可见该算法具有较高的稳定性和鲁棒性。下一步将考虑如何提高相关滤波器模型的跟踪能力以进一步提高整体算法的跟踪效果。

参 考 文 献

- [1] 沈玉玲, 伍忠东, 赵汝进, 等. 基于模型更新与快速重检测的长时目标跟踪[J]. 光学学报, 2020, 40(3): 0315002.
Shen Y L, Wu Z D, Zhao R J, et al. Long-term object tracking based on model updating and fast re-detection[J]. Acta Optica Sinica, 2020, 40(3): 0315002.
- [2] 陈法领, 丁庆海, 常铮, 等. 自适应特征融合的多尺度核相关滤波目标跟踪[J]. 光学学报, 2020, 40(3): 0315001.
Chen F L, Ding Q H, Chang Z, et al. Multi-scale kernel correlation filter algorithm for visual tracking based on the fusion of adaptive features[J]. Acta Optica Sinica, 2020, 40(3): 0315001.
- [3] 刘万军, 孙虎, 姜文涛. 自适应特征选择的相关滤波跟踪算法[J]. 光学学报, 2019, 39(6): 0615004.
Liu W J, Sun H, Jiang W T. Correlation filter tracking algorithm for adaptive feature selection[J]. Acta Optica Sinica, 2019, 39(6): 0615004.
- [4] 崔丽群, 贺情杰, 何牧泽. 基于双核模型上下文的流形正则相关滤波跟踪算法[J]. 激光与光电子学进展, 2021, 58(12): 1210011.
Cui L Q, He Q J, He M Z. Manifold regular correlation filter tracking algorithm based on dual-core model context [J]. Laser & Optoelectronics Progress, 2021, 58(12): 1210011.
- [5] 储珺, 危振, 缪君, 等. 基于遮挡检测和多块位置信息融合的分块目标跟踪算法[J]. 模式识别与人工智能, 2020, 33(1): 59-65.
Chu J, Wei Z, Miao J, et al. Block target tracking based on occlusion detection and multi-block position information fusion[J]. Pattern Recognition and Artificial Intelligence, 2020, 33(1): 59-65.
- [6] 许敬祥, 伍雪冬, 杨开运. 基于高斯输出约束的上下文相关滤波跟踪[J]. 激光与光电子学进展, 2020, 57(4): 041508.
Xu J X, Wu X D, Yang K Y. Context-aware correlation filter tracking based on Gaussian output constraint[J]. Laser & Optoelectronics Progress, 2020, 57(4): 041508.
- [7] Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters[C]//2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 13-18, 2010, San Francisco, CA, USA. New York: IEEE Press, 2010: 2544-2550.
- [8] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.
- [9] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 20-25, 2005, San Diego, CA, USA. New York: IEEE Press, 2005: 886-893.
- [10] Bertinetto L, Valmadre J, Golodetz S, et al. Staple: complementary learners for real-time tracking[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 1401-1409.
- [11] Wang M M, Liu Y, Huang Z Y. Large margin object tracking with circulant feature maps[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 4800-4808.
- [12] Prabhu U, Seshadri K, Savvides M. Automatic facial landmark tracking in video sequences using Kalman filter assisted active shape models[M]//Kutulakos K N. Trends and topics in computer vision. Lecture notes in computer science. Heidelberg: Springer, 2012, 6553: 86-99.
- [13] Li Y M, Fu C H, Ding F Q, et al. AutoTrack: towards high-performance visual tracking for UAV with automatic spatio-temporal regularization[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 11920-11929.
- [14] Du C J, Lan M Y, Gao M Y, et al. Real-time object tracking via adaptive correlation filters[J]. Sensors, 2020, 20(15): 4124.
- [15] Li F, Tian C, Zuo W M, et al. Learning spatial-temporal regularized correlation filters for visual tracking [C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 4904-4913.
- [16] Danelljan M, Bhat G, Khan F S, et al. ECO: efficient convolution operators for tracking[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 6931-6939.
- [17] Galoogahi H K, Fagg A, Lucey S. Learning background-aware correlation filters for visual tracking[C]//2017 IEEE International Conference on Computer Vision, October 22-29, 2017, Venice, Italy. New York: IEEE Press, 2017: 1144-1152.
- [18] Mueller M, Smith N, Ghanem B. Context-aware

- correlation filter tracking[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 1387-1395.
- [19] Danelljan M., Häger G., Khan F. S., et al. Learning spatially regularized correlation filters for visual tracking [C]//IEEE International Conference on Computer Vision (ICCV) 2015, December 7-13, 2015: 4310-4318, doi: [10.1109/ICCV.2015.490](https://doi.org/10.1109/ICCV.2015.490).
- [20] Danelljan M, Häger G, Khan F S, et al. Accurate scale estimation for robust visual tracking[C]//Proceedings of the British Machine Vision Conference 2014, September 1-5, 2014, Nottingham, UK. London: British Machine Vision Association, 2014.
- [21] Wu Y, Lim J, Yang M H. Object tracking benchmark[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834-1848.
- [22] Fan H, Lin L T, Yang F, et al. LaSOT: a high-quality benchmark for large-scale single object tracking[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 5369-5378.