

行动预测及尺度自适应目标跟踪算法

汤学猛, 陈志国^{1*}, 傅毅^{1,2}

¹江南大学物联网工程学院, 江苏 无锡 214122;

²无锡环境科学与工程研究中心, 江苏 无锡 214153

摘要 核相关滤波算法的出现将传统目标跟踪算法推向了一个新的高度, 以其高速度、高精度、高鲁棒性的特点迅速获得了广泛认可与关注。针对核相关滤波算法在尺度变化和快速移动方面的不足, 基于目标在发生尺度变化时其主要特征的相对位置可以反映其尺度的大小, 本文采用分块检测的方式来进行目标的尺度计算; 在特征模板更新方面, 针对核相关滤波算法处理高速变化目标容易丢失的问题, 采用卡尔曼滤波预测的方式提前更新目标模板, 以解决模板更新滞后的问题。实验数据表明, 本文算法能够稳定地提高跟踪精度。

关键词 图像处理; 尺度变化; 分块跟踪; 卡尔曼预测

中图分类号 TP391.4

文献标志码 A

doi: 10.3788/LOP57.081014

Action Prediction and Scale Adaptive Target Tracking Algorithm

Tang Xuemeng¹, Chen Zhiguo^{1*}, Fu Yi^{1,2}

¹ School of Internet of Things Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China;

² Wu Xi Research Center of Environmental Science and Engineering, Wuxi, Jiangsu 214153, China

Abstract The emergence of kernelized correlation filter has pushed the traditional target tracking algorithm to a new height. With its high speed, high precision, and high robustness, it has quickly gained wide recognition and attention from the society. In view of the shortcomings of the kernelized correlation filter in scale change, based on the relative position of the main features of the target when the scale changes and fast moving, the size of the target can be reflected. The block detection method is used to calculate the scale of the target. In respect, the kernelized correlation filter loses the target when dealing with the target with high-speed changes of the feature, and the target template is updated in advance by Kalman filter prediction to solve the problem of template update lag. Experimental results show that the proposed algorithm can improve the tracking accuracy steadily.

Key words image processing; scale change; block tracking; Kalman prediction

OCIS codes 100.4999; 100.2960

1 引言

使用目标跟踪定位技术确定目标在视频序列中的位置, 是计算机界最具有代表性的问题之一。目标跟踪在许多应用中有着至关重要的作用。特别是在监控及人机交互、横跨监控、运动分析、行为识别、医疗图像等方面。在目标跟踪过程中, 如环境变化、目标尺度变化等影响因素, 会导致跟踪的精确度和速度下降, 使得整个跟踪进程变得异常复杂^[1]。

目标跟踪算法大概分为生成式算法^[2]和判别式算法, 目标跟踪早期主要以生成式算法为主, 代表算法有: 卡尔曼算法^[3]、均值漂移算法^[4]等。生成式算法模型构造太过简单, 诸多因素使得其在跟踪目标的准确性上无法令人满意, 比如, 在环境变得更为复杂的时候, 模型无法准确抓取目标特征, 导致目标丢失大概率发生。经典判别式算法有 Struck^[5]、TLD (Tracking-Learning-Detection)^[6]等, Struck 和 TLD 在实时性方面效果还不错, 并且 Struck 是

收稿日期: 2019-07-17; 修回日期: 2019-08-12; 录用日期: 2019-09-16

基金项目: 国家自然科学基金(61502203)、江苏省自然科学基金(BK20150122)、江苏省高等学校自然科学研究面上项目(17KJB520039)、江苏省“333 高层次人才培养工程科研项目”(BRA2018147)

* E-mail: 427533@qq.com

2012年之前最好的方法。21世纪以来,在用经典机器学习方法去训练分类器的探索中,判别式算法的性能优势得到了验证,此后判别式算法迎来了快速发展^[7]。

尽管判别式算法具有较高的算法复杂度,依然不能满足目标实时性的要求。寻找一种速度更快、精确度更高,计算更快的算法成为新的研究方向。2010年,学者首次把通信领域的相关滤波方法引入目标跟踪中。在这个基础上,Bolme等^[8]提出了最小化输出均方误差和(MOSSE)算法。由于MOSSE算法出现,相关滤波器得到了飞速发展。2015年,Henriques等^[9]提出了KCF(kernelized correlation filters)算法,把相关滤波推到了发展的巅峰。针对相关滤波跟踪算法基本上分为两种:单模型和多模型。单模型主要采用单个特征训练滤波器,例如:CSK(circular struct and kernel)^[10]、判别相关滤波(DCF)^[11]。多模型是采用多个特征融合来训练滤波器,例如:增加尺度自适应跟踪器DSST(discriminative scale space tracking)^[12]、串接CN和FHOG(fast histogram of oriented gradients)特征的尺度自适应滤波器SAMF(scale adaptive with multiple features tracker)^[13]、CN和FHOG特征贡献进行加权组合尺度滤波器Staple^[14]。然而,传统的相关滤波器依然存在重大的缺陷,国内外的专家一直尝试在快速移动和尺度变化方面取得突破。何雪东等^[15]提出了快速尺度自适应核相关滤波目标跟踪算法。高美凤等^[16]在尺度方面做了改进,提出了尺度自适应核相关滤波目标跟踪。成悦等^[17]在相关滤波的基础上,提出了基于模型与尺度更新的相关滤波跟踪算法。单倩文等^[18]基于改进的多尺度特征提出快速检测和识别算法。Li等^[19]为了解决跟踪过程中目标尺度变化问题,在基于传统相关滤波器上提出了一种新的鲁棒的相似变换估计大位移(LDES)算法。随着传统方法遇到瓶颈,人工智能以及深度学习领域的兴起,人们尝试着把深度学习用于目标跟踪。2017年,Choi等^[20]在相关滤波器上增加了注意网络机制,提出了ACFN(Attentional Correlation Filter Network for Adaptive Visual Tracking)网络结构算法,成功解决了目标移动时尺度变化问题。Danelljan等^[21]发表了ECO(efficient convolution operators)算法。2018年,Zhang等^[22]提出通过空间对齐相关滤波器网络进行视觉跟踪(SACF)。为了解决目标尺度问题,Wang等^[23]提出了无监督深度跟踪(UDT)算

法。Danelljan等^[24]通过重叠最大化实现对目标精确定位和尺度变化。

然而这些算法着重于算法的精确度或者尺度变化,对目标快速移动和尺度变化效果一般。为了解决目标因尺度变化和快速移动而导致不能准确定位的问题,本文首先采用分块跟踪使得跟踪过程中的目标尺度自适应,再用卡尔曼预测根据目标前几帧运动状态进行预定位,使得在目标周围的特征更大化,更新目标模板,然后使用模板对待测帧进行检测,从而提高跟踪的性能。

2 KCF基本跟踪原理

KCF跟踪算法^[1]是通过循环矩阵对滤波器训练,并且通过岭回归来求解参数。KCF滤波器的关键是找到回归问题的最优解:

$$\min_{\mathbf{w}} \sum_i [f(x_i) - y_i]^2 + \lambda \|\mathbf{w}\|^2. \quad (1)$$

其目的是为了找到一组 \mathbf{w} ,最后需要在傅里叶域计算更新权值,解得

$$\mathbf{w} = (\mathbf{X}^H \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^H \mathbf{y}, \quad (2)$$

式中: \mathbf{I} 是单位矩阵, \mathbf{y} 为 y_i 所组成的向量,训练样本集 (x_i, y_i) 。 $\mathbf{X} = (x_1, x_2, x_3, \dots, x_n)^T$, λ 为正则化系数。这里的 $\mathbf{X}^H = (\mathbf{X}^*)^T$,并且 \mathbf{X}^* 是 \mathbf{X} 的共轭。为了解决矩阵求逆运算量大的问题,将循环矩阵进行傅里叶对角化,将矩阵求逆转化为向量点积运算。这样就将解 \mathbf{w} 最优的问题变为解最优 α 的问题,由文献^[25]可知快速求解方法如下

$$\bar{f}(\alpha) = \frac{\bar{f}(y)}{\bar{f}(K) + \lambda}. \quad (3)$$

训练阶段结束之后,对于下一帧,在以上一帧目标中心点为测试中点、大小为 $N \times M$ 的图像块上进行检测。相关响应图可表示为

$$f(z) = \mathcal{F}^{-1} [\bar{f}(K) \odot \bar{f}(\alpha)]. \quad (4)$$

$f(z)$ 的最大值就是目标中心位置。为了更好地跟踪目标,采用双线性插值方法对滤波器的系数和目标模板进行更新,

$$\hat{\alpha}^j = (1 - \gamma) \hat{\alpha}^{j-1} + \gamma \hat{\alpha}, \quad (5)$$

$$\hat{x}^j = (1 - \gamma) \hat{x}^{j-1} + \gamma \hat{x}, \quad (6)$$

式中: γ 表示学习率; j 表示帧数。

3 改进的KCF算法

KCF跟踪算法采用固定大小窗口来训练滤波器,跟踪目标过程中尺度不会发生变化,致使模型对

误差累积变大,最终导致目标跟踪失败。其次,当目标快速运动时,目标可能出现在检测框之外,导致模型检测不到目标相应的位置,也会出现跟踪失败。针对上述问题,本文在 KCF 的基础上提出了行动预测的尺度自适应算法。在 KCF 算法上,增加了卡尔曼算法预测机制和分块跟踪。

3.1 基于卡尔曼算法运动预测

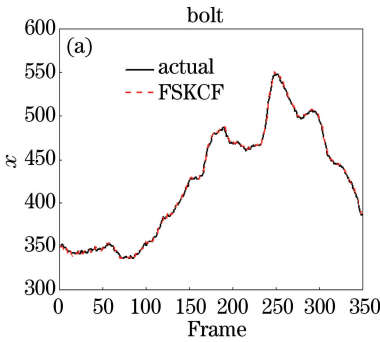
在跟踪过程中,为了使模板对目标更好地响应,本文使用卡尔曼滤波对目标运动提前一步进行判断。卡尔曼滤波器是线性观测方程,主要的核心公式有 5 个。时间更新如下

$$\hat{x}_k^- = \mathbf{A}\hat{x}_{k-1} + \mathbf{B}u_{k-1}, \quad (7)$$

$$P_k^- = \mathbf{A}P_{k-1}\mathbf{A}^T + \mathbf{Q}, \quad (8)$$

式中; \hat{x}_k^- 是 k 时刻的状态值; P_k^- 为新的误差协方差; \mathbf{Q} 为噪声, \mathbf{A} 、 \mathbf{B} 为单位矩阵。 \hat{x}_k^- 为上一帧目标所在位置目标, u_{k-1} 为目标最近几帧目标运动的速度,通过(7)式粗略估算目标当前位置。状态更新如下

$$K_k = P_k^- \mathbf{H}^T (\mathbf{H}P_k^- \mathbf{H}^T + \mathbf{R})^{-1}, \quad (9)$$



$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - \mathbf{H}\hat{x}_k^-), \quad (10)$$

$$P_k = (\mathbf{I} - K_k \mathbf{H})P_k^-, \quad (11)$$

式中; K_k 为卡尔曼增益; \hat{x}_k 校正更新 k 时刻的状态值; P_k 为增益矩阵方差误差, \mathbf{H} 为单位矩阵。根据(9)式,卡尔曼预测使用核矩阵对粗略的目标重新估算,来预测目标在当前的位置。卡尔曼预测小车下一刻出现的位置,如图 1 所示。



图 1 预测小车运动

Fig. 1 Predicting car movement

以测试视频 bolt 为例,在预测过程中,采用帧数作为自变量,分别对位置的横坐标和纵坐标进行预测,预测结果如图 2 所示。

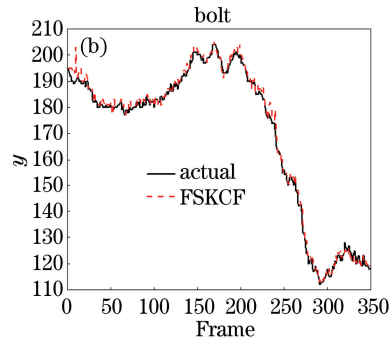


图 2 预测结果。(a) x 坐标预测;(b) y 坐标预测

Fig. 2 Prediction results. (a) x coordinate prediction; (b) y coordinate prediction

3.2 尺度估计

本文使用分块跟踪算法估算目标整体尺度。分块跟踪算法是把样本块均分成 9 份,由于中心特征

块重合,为了减少计算量,只选择其他 8 个特征块进行分块跟踪,中间位置用整体的目标跟踪,这样就会得到 9 个目标位置,如图 3 所示。

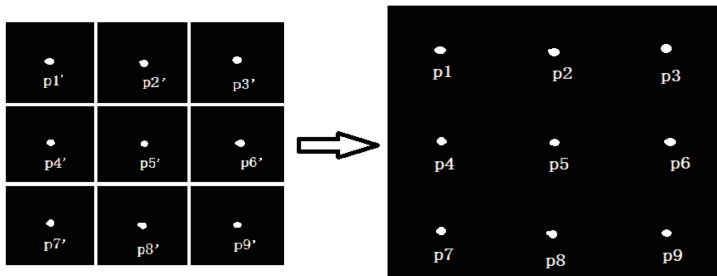


图 3 分块检测结果

Fig. 3 Block detection results

图中 $p5'$ 为卡尔曼预测当前帧目标位置, $p1', \dots, p4', p6', \dots, p9'$ 分别是根据 $p5'$ 算出的 8

个特征块目标中心, $p5$ 为 KCF 得到的目标位置, $p1, \dots, p4, p6, \dots, p9$ 为 8 个特征块得到的位置。

特征点定位之后通过计算特征点之间的相对位置就可以算出前后帧的尺度变化。特征块分成 9 块,因此一个特征块的特征比原来少了,在跟踪过程中,8 个分块可能会出现跟踪误差。为了降低误

差带来的影响,通过 9 个点的坐标计算出当前目标的尺度。 x 和 y 分别为当前帧目标框的长和高; C_{target} 为上一帧目标尺度, C_{scal} 为当前尺度因子。计算公式为

$$\begin{cases} x = \frac{\Delta_{p^2_x - p^1_x} + \Delta_{p^3_x - p^2_x} + \Delta_{p^5_x - p^4_x} + \Delta_{p^6_x - p^5_x} + \Delta_{p^8_x - p^7_x} + \Delta_{p^9_x - p^8_x}}{2} \\ y = \frac{\Delta_{p^4_y - p^1_y} + \Delta_{p^7_y - p^4_y} + \Delta_{p^5_y - p^2_y} + \Delta_{p^8_y - p^5_y} + \Delta_{p^6_y - p^3_y} + \Delta_{p^9_y - p^6_y}}{2} \end{cases}, \quad (12)$$

$$C_{scal} = \frac{xy}{C_{target}} \quad (13)$$

3.3 流程图

KCF 用上一帧计算得到分块模板与目标模板分别对当前帧的目标和分块的特征位置点进行定位,得到当前帧目标以及各个分块中心点的位置,结

合中心点的计算获得当前的目标尺度信息,然后输出,再将分块位置根据中心点重新定位。虽然分块在数量上增加了,但是每个检测块的特征减少了,满足了实时要求,如图 4 所示。

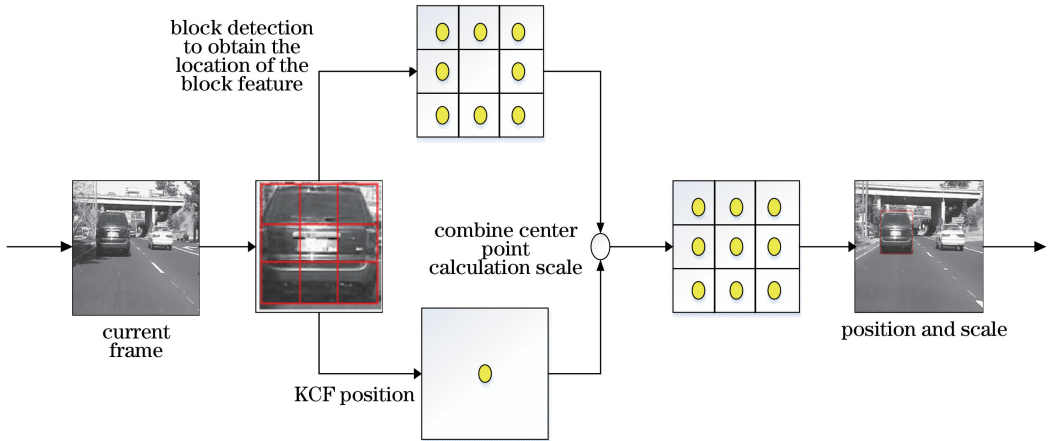


图 4 算法流程

Fig. 4 Flow chart of algorithm

图 5 为改进算法的流程图。

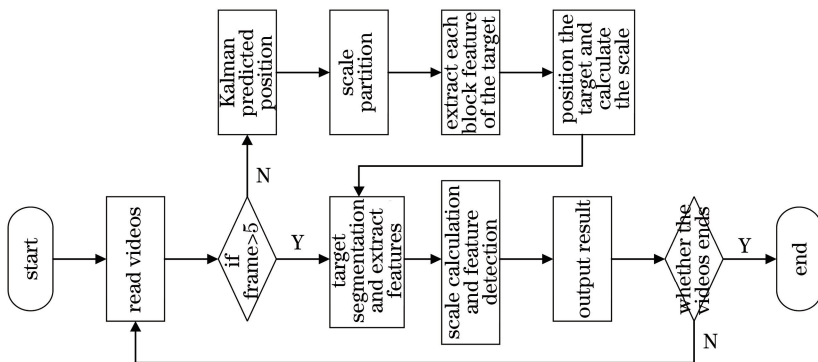


图 5 改进算法的流程图

Fig. 5 Flow chart of improved algorithm

4 实验结果分析

4.1 实验环境与参数

为了验证本文 FSKCF 算法的有效性,本文在 17 个视频上进行综合测评,并与目前比较先进的跟踪算法进行对比,如 DSST、ECO、KCF、LDES、LADCF^[25] 和 SAMF。测试平台为 Matlab2018b,所有实验均在 Intel(R) Core(TM) i3-7100 CPU,主频 3.9 GHz,8 G 内存配置的 windows10 64 bit PC 系统上完成。本文算法的基本参数仍采用原 KCF 参数,且在所有视频序列上均保持一致,高斯核的标准差 $\sigma=0.5$,正则化系数 $\lambda=0.0001$,空间带宽参数为 0.1。实验中方向梯度直方图特征(HOG)的 cell 大小为 4,梯度方向个数为 9,学习参数 $\eta=0.02$ 。本文从 Visual Tracker Benchmark 上选取 17 组有关尺度变化和快速移动的视频进行测试,并与其他 6 种算法进行对比。

表 1 实验中测试视频

Table 1 Test video in the experiment

| Video | Number of frames | Main factor of influence |
|----------|------------------|---------------------------------------|
| BlurCar2 | 576 | SV, MB, FM |
| Bolt | 350 | SV, DEF, BC, OCC, IPR, OPR |
| Bolt2 | 293 | SV, DEF, BC |
| Box | 400 | SV, IV, OCC, MB, IPR, OPR, OV, BC, LR |
| Boy | 602 | SV, MB, FM, IPR, OPR |
| Car2 | 913 | SV, MB, FM, BC, IV |
| Car4 | 659 | SV, IV, BC |
| Dancer | 225 | SV, DEF, IPR, OPR |
| Doll | 1000 | IV, SV, OCC, IPR, OPR |
| Human6 | 792 | SV, OCC, DEF, FM, OPR, OV |
| Human8 | 128 | IV, SV, DEF |
| Singer1 | 319 | IV, SV, OCC, OPR |
| Soccer | 380 | IV, SV, OCC, MB, FM, IPR, OPR, BC |
| Surfer | 370 | SV, FM, IPR, OPR, LR |
| Walking | 412 | SV, OCC, DEF |
| Walking2 | 500 | SV, OCC, LR |
| Woman | 597 | SV, OCC, IV, DEF, MB, FM, OPR |

4.2 定性分析

为了更好的对比实验结果,本文显示了 7 种算法在部分视频集上的结果,如图 6 所示。视频 BlurCar2 上中目标“汽车”一直在不停地变化尺度,从图 6(a)中可以看出,第 100 帧、第 200 帧和第 300 帧中,目标的尺度一直在变小,7 种算法能准确地跟踪,但是 FSKCF 相较其他对比算法在尺度变化上

效果更佳。从图 6(b)中可以看出,在视频 Box 上,第 50 帧、第 100 帧中,ECO、LDES 和 KCF 在尺度变化上出现了漂移。从图 6(c)中可以看出,视频 Boy 中,目标出现上下左右移动,在第 250 帧,其余 6 种算法对尺度估计都出现了问题,只有 FSKCF 能准确地估计目标尺度变化和定位。在图 6(d)中,目标快速移动,并且受到的光照不均匀,使得 SAMF 和 LDES 出现了尺度估计错误。在图 6(e)中,视频 Singer1 跟踪目标受到灯光照射,背景复杂,还有目标尺度一直变小,导致了 SAMF 和 KCF 一开始目标尺度变化就失败了,DSST 在第 319 帧时,跟踪目标位置出现了误差,导致尺度变化受到了影响。在图 6(f)中,视频 Surfer 跟踪目标发生了快速移动和目标变化,但是由于背景变化比较单一,受其他影响较小,因此除了 DSST,其他算法均表现不错。在图 6(g)中,视频 Walking2 上,目标受到了相似遮挡物干扰,使得在第 250 帧,KCF 跟踪出现漂移,在第 500 帧时,KCF 完全丢失跟踪目标。SAMF 在跟踪过程中,也受到遮挡物的影响,导致在尺度变化上效果一般。而 FSKCF 是对跟踪目标的提前预测,减少了相似遮挡物的干扰,降低了目标模板被污染的概率,能准确地保留目标模板信息。其次,通过分块跟踪得到目标的尺度,从而能准确地估计目标尺度变化和定位。在图 6(h)中,KCF 尺度一直没有变化,SAMF 在第 200 帧时尺度估计变大,其他几种算法跟踪结果良好。

4.3 定量分析

为了更好地分析本文算法的性能,采用距离精确度(DP) $P_{DP} = m/N$,中心误差(CLE) $E_{CLE} = \sqrt{(x_t - x_{t_0})^2 + (y_t - y_{t_0})^2}$ 、速度 $V = N/t_{oc}$ 、成功率(SR) $P_{SR} = S_n/N$ 作为综合指标。其中, m 为 E_{CLE} 小于等于固定阈值(实验中取值 20 pixel)的次数, N 为总数。 (x_t, y_t) 为第 t 帧滤波器得到的目标中心位置, (x_{t_0}, y_{t_0}) 为目标在第 t 帧的真实中心位置。 t_{oc} 为跟踪算法跑完这个视频所需要的时间。 $P_{score} = S_{area}(R_t \cap R_{gt}) / S_{area}(R_t \cup R_{gt})$ 为跟踪目标重叠率, R_t 为跟踪得到的目标窗口, R_{gt} 为真实目标窗口, S_{area} 是区域面积。 S_n 为成功的次数,当 $P_{score} > 0.4$ 时, S_n 增加 1。

首先分析在移动速度比较快的视频上的结果。Bolt 等 6 个测试视频中的目标移动速度比较快,表 2 所示为算法在视频上的准确度。DSST、KCF、LDES、LADCF 和 SAMF 在 6 个视频上的准确度都低于 90%,分别为 69.48%,69.23%,89.63%,

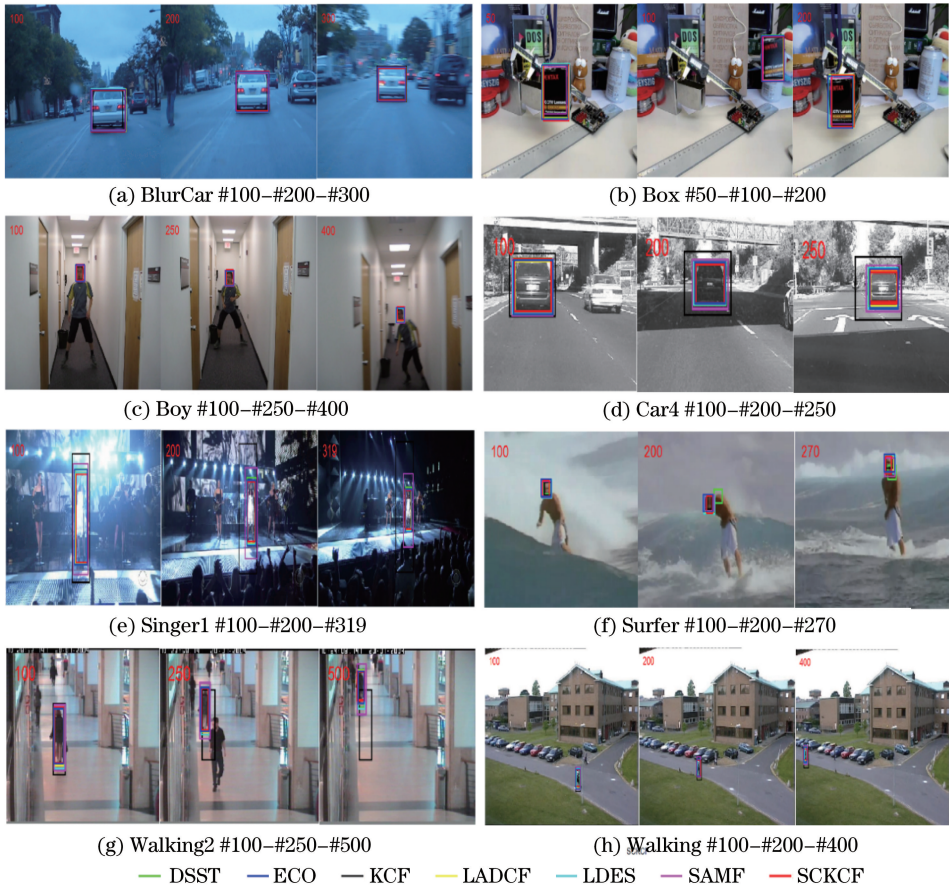


图 6 部分实验结果。(a) BlurCar; (b) Box; (c) Boy; (d) Car4; (e) Singer1; (f) Surfer; (g) Walking2; (h) Walking
 Fig. 6 Partial experimental results. (a) BlurCar; (b) Box; (c) Boy; (d) Car4; (e) Singer1; (f) Surfer; (g) Walking2; (h) Walking

表 2 算法在快速运动视频上的跟踪准确率

Table 2 Tracking precision of the algorithm on fast motion video

| Algorithm | DSST | ECO | KCF | LADCF | LDES | SAMF | FSKCF | % |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bolt | 100.00 | 100.00 | 98.90 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| Bolt2 | 2.10 | 83.30 | 1.70 | 39.90 | 01.70 | 1.70 | 100.00 | 100.00 |
| Boy | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| Car4 | 100.00 | 100.00 | 95.00 | 100.00 | 99.80 | 100.00 | 100.00 | 100.00 |
| Human6 | 44.80 | 98.90 | 29.00 | 97.90 | 98.90 | 92.30 | 99.70 | 99.70 |
| Surfer | 70.00 | 100.00 | 90.80 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| Average | 69.48 | 97.03 | 69.23 | 89.63 | 83.40 | 82.33 | 99.95 | 99.95 |

83.40% 和 82.33%。ECO 的准确度为 97.03%，FSKCF 的准确度为 99.95%，远远优于其他算法的效果。从表 2 数据可以看出，本文方法在运动预测方面具有很好的效果。

表 3 所示为算法在所有测试视频上的准确度。DSST 在视频 Bolt2、Human6、Soccer 和 Surfer 上的准确率都低于 90%，分别为 2.1%，44.8%，70.5% 和 70.0%，出现部分丢失情况。ECO 在视频 Soccer 上的准确率只有 24%。KCF 在视频 Bolt2、

Human6, Soccer 和 Walking2 上的准确率都低于 90%，分别为 1.7%，29%，81.3% 和 44%。LADCF 在视频 Bolt2、Soccer 上的准确率分别为 21.1%，39.9%。LDES 在视频 Bolt2、Human6 和 Soccer 上的准确率都低于 90%，分别为 1.7%，32.8% 和 24.7%。SAMF 在视频 Bolt2 和 Soccer 上得准确率分别为 1.7% 和 20.8%。准确率以 90% 为标准，对比算法在部分视频上的跟踪过程中会出现丢失，导致跟踪效果不好，最终算法的准确率低于 90%。而

表3 算法在视频上的跟踪准确率

Table 3 Tracking precision of the algorithm on the video

%

| Algorithm | DSST | ECO | KCF | LADCF | LDES | SAMF | FSKCF |
|-----------|-------|-------|-------|-------|-------|-------|-------|
| BlurCar2 | 100.0 | 100.0 | 97.9 | 100.0 | 100.0 | 100.0 | 100.0 |
| Bolt | 100.0 | 100.0 | 98.9 | 100.0 | 100.0 | 100.0 | 100.0 |
| Bolt2 | 2.1 | 83.3 | 1.7 | 39.9 | 1.7 | 1.7 | 100.0 |
| Box | 100.0 | 99.8 | 100.0 | 100.0 | 98.8 | 100.0 | 100.0 |
| Boy | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Car2 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Car4 | 100.0 | 100.0 | 95.0 | 100.0 | 99.8 | 100.0 | 100.0 |
| Dancer | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Doll | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Human6 | 44.8 | 98.9 | 29.0 | 97.9 | 98.9 | 92.3 | 99.7 |
| Human8 | 100.0 | 100.0 | 100.0 | 100.0 | 32.8 | 100.0 | 100.0 |
| Singer1 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Soccer | 70.5 | 24.0 | 81.3 | 21.1 | 24.7 | 20.8 | 97.4 |
| Surfer | 70.0 | 100.0 | 90.8 | 100.0 | 100.0 | 100.0 | 100.0 |
| Walking | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Walking2 | 100.0 | 100.0 | 44.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Woman | 93.8 | 100.0 | 93.8 | 100.0 | 98.5 | 93.8 | 99.2 |

FSKCF 在测试的视频上准确率都超过 90%，跟踪效果良好。

表 4 所示为算法在所有测试视频上的重叠率。在重叠方面,本文 FSKCF 与原 KCF 相比,KCF 在测试视频上的重叠情况明显低于 FSKCF。例如,在视频 Walking 上,KCF 的重叠率只有 0.73,而 FSKCF 为 1。ECO 和 LADCF 在测试视频上效果不错。因此,ECO、LADCF 和 FSKCF 在测试视频上的重叠效果比较好。

将 7 种算法对 17 组视频进行实验,获得在 17 组视频平均的 P_{DP} 、 E_{CLE} 、 OP 、 V 、 P_{SR} ,如表 5 所示。表 5 所示为几种算法在 17 个视频上指标的均值。与原始 KCF 算法相比,FSKCF 的精确度由 89.4% 提高到 99.8%,平均中心位置误差均从 33 降到 2.7,速度从 224.6 降到 66.7。在 DP 和 CLE 上都提升很多,有着很大的优势。虽然在速度方面下降了,但是仍然没有影响到视觉的流畅性,人眼视觉效果的速度为 24 frame/s,只要帧率 $V > 24$ 就能达到实时的要求了。

表4 算法在视频上的重叠率

Table 4 Overlap rate of the algorithm tracking on the video

| Algorithm | DSST | ECO | KCF | LADCF | LDES | SAMF | FSKCF |
|-----------|------|------|------|-------|------|------|-------|
| BlurCar2 | 1.00 | 1.00 | 0.98 | 1.00 | 1.00 | 1.00 | 1.00 |
| Bolt | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 0.99 |
| Bolt2 | 0.01 | 0.78 | 0.01 | 0.30 | 0.01 | 0.01 | 0.23 |
| Box | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Boy | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Car2 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Car4 | 1.00 | 1.00 | 0.51 | 1.00 | 1.00 | 1.00 | 1.00 |
| Dancer | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.96 |
| Doll | 1.00 | 1.00 | 0.90 | 1.00 | 1.00 | 1.00 | 0.86 |
| Human6 | 0.46 | 0.99 | 0.24 | 0.99 | 1.00 | 0.47 | 0.28 |
| Human8 | 1.00 | 1.00 | 0.73 | 1.00 | 0.29 | 0.78 | 1.00 |
| Singer1 | 1.00 | 1.00 | 0.44 | 1.00 | 1.00 | 0.75 | 1.00 |
| Soccer | 0.57 | 0.23 | 0.44 | 0.24 | 0.23 | 0.21 | 0.58 |
| Surfer | 0.36 | 0.97 | 0.74 | 1.00 | 1.00 | 1.00 | 0.74 |
| Walking | 1.00 | 1.00 | 0.73 | 1.00 | 1.00 | 1.00 | 1.00 |
| Walking2 | 1.00 | 1.00 | 0.43 | 1.00 | 1.00 | 0.99 | 1.00 |
| Woman | 0.94 | 0.93 | 0.94 | 0.94 | 0.96 | 0.94 | 0.98 |

表5 不同算法在17个视频上指标的均值

Table 5 Mean values of indicators on 17 videos by different algorithms

| Parameter | DSST | ECO | KCF | LADCF | LDES | SAMF | FSKCF |
|---------------|------|------|-------|-------|------|------|-------|
| DP/% | 92.4 | 95.2 | 89.4 | 94.9 | 90.8 | 94.2 | 99.8 |
| CLE | 21.7 | 11.5 | 33.0 | 12.5 | 36.4 | 18.5 | 2.7 |
| OP | 0.87 | 0.93 | 0.74 | 0.91 | 0.84 | 0.85 | 0.88 |
| $V/(frame/s)$ | 32.7 | 44.3 | 224.6 | 12.8 | 11.0 | 15.0 | 66.7 |

与其他算法相比,DSST、KCF和LDES算法在部分视频集上出现跟踪丢失的情况,导致整体在DP和CLE评价上表现不佳。ECO在DP上达到95.2%,CLE上11.5。LADCF在DP上达到94.9%,在CLE是12.5。在重叠率方面,本文FSKCF把平均重叠率从KCF的0.74提升到0.88。FSKCF稍微比ECO和LADCF低一点,但是仍然比DSST、KCF、LDES和SAMF效果要好。在速度方面,LADCF、LDES和SAMF都不能达到实时性的要求。DSST

和ECO分别达到了32.7和44.3。ECO在速度上也比FSKCF差。

因此,本文算法在DP和CLE上比其他6种算法效果要好,在跟踪性能上和尺度变换方面有很大的提升。实验中绘制了部分测试视频的DP和CLE评价曲线,分别见图7和图8。从图7中可以看出,FSKCF的准确度高于其他算法。从图8中可以看出,FSKCF在测试视频上的平均误差明显低于其他算法。

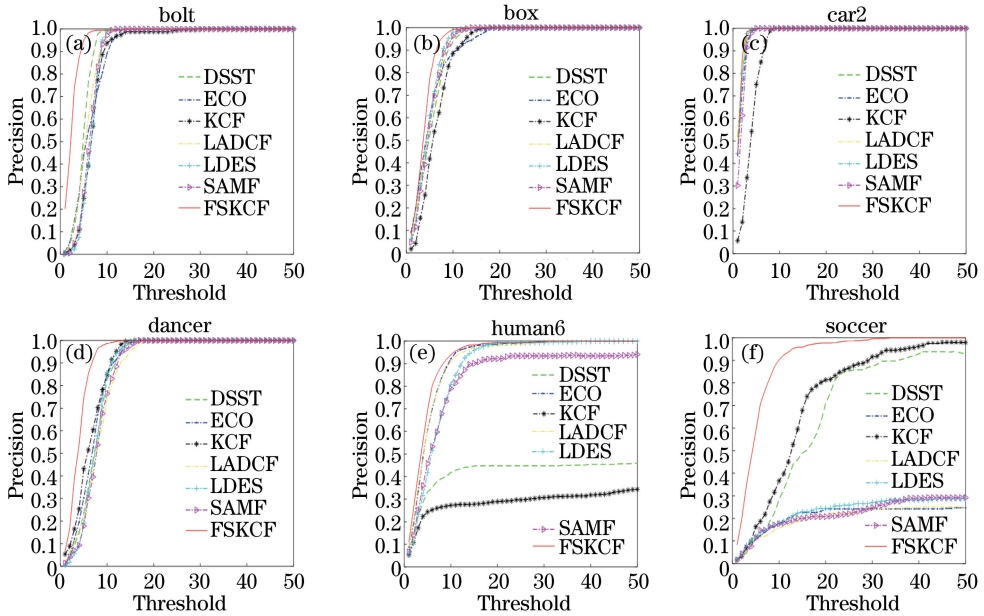


图7 部分实验结果的准确度曲线图。(a) Bolt;(b) Box;(c) Car2;(d) Dancer;(e) Human6;(f) Soccer

Fig. 7 Precision curves of partial experimental result. (a) Bolt; (b) Box; (c) Car2; (d) Dancer; (e) Human6; (f) Soccer

将本文算法与其他算法进行对比,由于每个视频帧数不一样,本文选取测试视频中15个视频的前250帧,绘制CLE和DP评价曲线。DP是整个测试视频的平均精确度。从图9可以看出,FSKCF在CLE和DP上都有明显优势,对KCF算法在快速移动与尺度变化方面有着很好的改进。

5 结论

针对KCF算法在目标快速移动和尺度变化的缺陷,本文采用分块检测的方法解决目标的尺度变化的问题,此方法在处理尺度变化目标上有着优异

的性能,可以准确地跟踪目标的尺度信息从而提高算法的跟踪精度,并在此基础上加入了卡尔曼滤波器来对目标的运动进行预测,以期可以提前获取当前目标的大致特征模板来应对目标接下来的特征变化,提高跟踪精度。实验表明,这两项改进可以极大地提高KCF的跟踪精度和效果。然而改进的算法也存在着某些不足,例如虽然算法在性能方面得到了优化,但是增加了分块跟踪和预测机制,速度方面受到了影响。因此,上述存在的问题,将是下一步研究的出发点与方向。

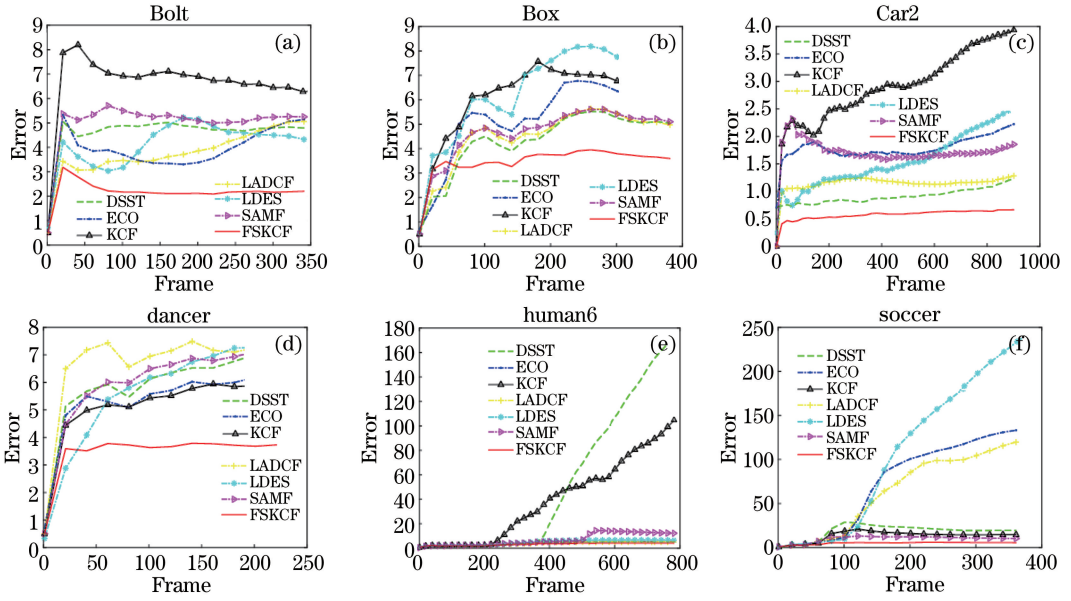


图 8 部分实验结果的误差曲线图。(a) Bolt; (b) Box; (c) Car2; (d) Dancer; (e) Human6; (f) Soccer

Fig. 8 Error curves of partial experimental result. (a) Bolt; (b) Box; (c) Car2; (d) Dancer; (e) Human6; (f) Soccer

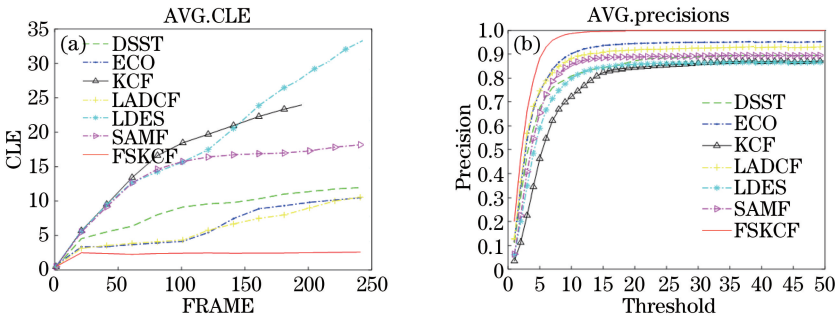


图 9 测试结果。(a)平均 CLE(前 250 帧);(b)DP 曲线图

Fig. 9 Test results. (a) Average CLE (front 250 frame); (b) DP graph

参 考 文 献

[1] Wu X J, Xu T Y, Xu W B. Review of target tracking algorithms in video based on correlation filter [J]. Command Information System and Technology, 2017, 8(3): 1-5.
 吴小俊, 徐天阳, 须文波. 基于相关滤波的视频目标跟踪算法综述[J]. 指挥信息系统与技术, 2017, 8(3): 1-5.

[2] Zhu W Q, Liu Y, Bian L, et al. Survey on object tracking method base on generative model [J]. Microprocessors, 2017, 38(1): 41-47.
 朱文青, 刘艳, 卞乐, 等. 基于生成式模型的目标跟踪方法综述[J]. 微处理机, 2017, 38(1): 41-47.

[3] Nguyen H T, Smeulders A W M. Fast occluded object tracking by a robust appearance filter [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(8): 1099-1104.

[4] Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25 (5): 564-577.

[5] Hare S, Golodetz S, Saffari A, et al. Struck: structured output tracking with kernels [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 38(10): 2096-2109.

[6] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409-1422.

[7] Mei X, Ling H B. Robust visual tracking using ℓ_1 minimization [C] // 2009 IEEE 12th International Conference on Computer Vision, September 29-October 2, 2009, Kyoto, Japan. New York: IEEE, 2009: 1436-1443.

[8] Bolme D, 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, 2010: 2544-2550.
- [9] 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.
- [10] Henriques J F, Caseiro R, Martins P, et al. Exploiting the circulant structure of tracking-by-detection with kernels[M] // Fitzgibbon A, Lazebnik S, Perona P, et al. Lecture notes in computer science. Computer vision-ECCV 2012. Berlin, Heidelberg: Springer, 2012, 7575: 702-715.
- [11] Lukezic A, Vojir T, Cehovin Zajc L, et al. Discriminative correlation filter with channel and spatial reliability [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, Hawaii, USA. New York: IEEE, 2017: 6309-6318.
- [12] Danelljan M, Häger G, Khan F, et al. Accurate scale estimation for robust visual tracking [C] // British Machine Vision Conference, September 1-5, 2014, Northumbria University, Newcastle, UK. UK: BMVA Press, 2014.
- [13] Li Y, Zhu JK. A scale adaptive kernel correlation filter tracker with feature integration [M] // Agapito L, Bronstein M, Rother C. Computer vision-ECCV 2014 Workshops. Lecture notes in computer science. Cham: Springer, 2015, 8926: 254-265.
- [14] 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 (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 1401-1409.
- [15] He X D, Zhou S Z. Fast scale adaptive kernel correlation filtering algorithm for target tracking [J]. Laser & Optoelectronics Progress, 2018, 55(12): 121501.
何雪东, 周盛宗. 快速尺度自适应核相关滤波目标跟踪算法[J]. 激光与光电子学进展, 2018, 55(12): 121501.
- [16] Gao M F, Zhang X X. Scale adaptive kernel correlation filtering for target tracking [J]. Laser & Optoelectronics Progress, 2018, 55(4): 041501.
高美凤, 张晓玄. 尺度自适应核相关滤波目标跟踪[J]. 激光与光电子学进展, 2018, 55(4): 041501.
- [17] Cheng Y, Li J Z, Zhu L N, et al. Correlation filter tracking algorithm based on model and scale updating [J]. Laser & Optoelectronics Progress, 2018, 55(12): 121015.
成悦, 李建增, 褚丽娜, 等. 基于模型与尺度更新的相关滤波跟踪算法[J]. 激光与光电子学进展, 2018, 55(12): 121015.
- [18] Shan Q W, Zheng X B, He X H, et al. Fast object detection and recognition algorithm based on improved multi-scale feature maps [J]. Laser & Optoelectronics Progress, 2019, 56(2): 021002.
单倩文, 郑新波, 何小海, 等. 基于改进多尺度特征图的目标快速检测与识别算法[J]. 激光与光电子学进展, 2019, 56(2): 021002.
- [19] Li Y, Zhu J K, Hoi S C H, et al. Robust estimation of similarity transformation for visual object tracking [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33: 8666-8673.
- [20] Choi J, Chang H J, Yun S, et al. Attentional correlation filter network for adaptive visual tracking [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 4828-4837.
- [21] 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 (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 6931-6939.
- [22] Zhang M D, Wang Q, Xing J L, et al. Visual tracking via spatially aligned correlation filters network [M] // Ferrari V, Hebert M, Sminchisescu C, et al. Computer vision-ECCV 2018. Lecture notes in computer science. Cham: Springer, 2018, 11207: 484-500.
- [23] Wang N, Song Y, Ma C, et al. Unsupervised deep tracking [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 16-20, 2019, Long Beach, CA, USA. New York: IEEE, 2019: 1308-1317.
- [24] Danelljan M, Bhat G, Khan F S. ATOM: accurate tracking by overlap maximization [J/OL]. [2019-07-16]. https://www.researchgate.net/publication/329057490_ATOM_Accurate_Tracking_by_Overlap_Maximization.
- [25] Xu T Y, Feng Z H, Wu X J, et al. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking [J/OL]. (2019-06-19) [2019-07-16]. <https://arxiv.xileisou.top/abs/1807.11348>.