

## 复杂场景下的自适应相关滤波跟踪算法

卢明瑞<sup>1,2</sup>, 韩超<sup>1,2\*</sup>, 鲁帆<sup>1,2</sup>, 缪宝睿<sup>1,2</sup>, 杨基坤<sup>1,2</sup>, 查君君<sup>1,2</sup>, 沙文瀚<sup>3</sup><sup>1</sup>安徽工程大学电气工程学院, 安徽 芜湖 241000;<sup>2</sup>安徽工程大学高端装备先进感知与智能控制教育部重点实验室, 安徽 芜湖 241000;<sup>3</sup>奇瑞新能源汽车股份有限公司, 安徽 芜湖 241000

**摘要** 针对目标跟踪算法应对遮挡、模糊、尺度变换等挑战时, 容易导致漂移和跟踪失败的情况, 提出一种复杂场景下的自适应相关滤波跟踪算法。首先, 采用所提的多特征互补策略, 利用特征训练相应滤波器, 根据每个滤波器的响应值动态调整特征的融合权重, 完成对目标的位置估计; 然后, 以估计位置中心构建尺度滤波器, 完成目标最优尺度的估计; 最后, 融合多尺度搜索区域策略, 并根据跟踪置信度对跟踪模型选择性进行更新, 进一步提升了跟踪器的性能和抗遮挡能力。在 OTB2015 的 74 组彩色数据集上进行测试, 并对所提算法与近年来先进的相关滤波算法进行对比。所提算法的平均距离精度为 0.801, 平均重叠精度为 0.715, 实时跟踪速度为 39.24 frame/s。实验结果表明, 跟踪器在复杂环境下的表现良好, 整体性能优越。

**关键词** 机器视觉; 目标跟踪; 相关滤波; 多特征融合; 遮挡判别

中图分类号 TP391 文献标志码 A

DOI: 10.3788/LOP202259.2415004

## Adaptive Correlation Filtering Tracking Algorithm for Complex Scenes

Lu Mingrui<sup>1,2</sup>, Han Chao<sup>1,2\*</sup>, Lu Fan<sup>1,2</sup>, Miao Baorui<sup>1,2</sup>, Yang Jikun<sup>1,2</sup>, Zha Junjun<sup>1,2</sup>, Sha Wenhan<sup>3</sup><sup>1</sup>School of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, Anhui, China;<sup>2</sup>Key Laboratory of Advanced Perception and Intelligent Control of High-End Equipment, Ministry of Education, Anhui Polytechnic University, Wuhu 241000, Anhui, China;<sup>3</sup>Chery New Energy Automobile Co., Ltd., Wuhu 241000, Anhui, China

**Abstract** When the target tracking algorithm deals with occlusion, blur, scale transformation, and other challenges, it can easily result in drift and tracking failure; thus, an adaptive correlation filtering tracking algorithm for complex scenes is proposed. First, the proposed multi-feature complementarity method is employed to train the corresponding filters with features, and the fusion weight of features is dynamically adjusted according to the response value of each filter to complete the position estimation for the target. Then, a scale filter is constructed with the center position to estimate the optimal scale of the target. Finally, the multi-scale search region method is integrated, and the tracking model is selectively updated according to the tracking confidence degree, which further enhances the performance and anti-occlusion ability of the tracker. Tests were performed on 74 color datasets of OTB2015 and the proposed algorithm was compared with the advanced correlation filtering algorithms recently. The average distance accuracy of the proposed algorithm is 0.801, the average overlap accuracy is 0.715, and the real-time tracking speed is 39.24 frame/s. Experimental results show that the tracker performs well in a complex environment and has an excellent overall performance.

**Key words** machine vision; target tracking; correlation filtering; multi-feature fusion; occlusion discrimination

## 1 引言

目标跟踪作为计算机视觉的重要分支, 广泛应用

于智能监控、无人驾驶、智能交通等领域。复杂环境下对动态物体的实时跟踪是一个较为困难的问题, 遮挡、光照变换、尺度变换、快速形变等因素都会对跟踪器产

收稿日期: 2021-09-06; 修回日期: 2021-10-12; 录用日期: 2021-10-27

基金项目: 安徽工程大学检测技术与节能装置安徽省重点实验室开放基金(DTESD2020A06)、2021年度安徽高校研究生科学研究项目(YJS20210447)、芜湖市科技计划项目(2021cg21)

通信作者: hanchaozh@126.com

生较大的影响<sup>[1-4]</sup>。设计一款鲁棒性强、实时性高的跟踪器是当下关注和研究的热点<sup>[5-6]</sup>。

近年来,相关滤波算法在目标跟踪领域的各个基准数据集上取得良好的表现<sup>[7-8]</sup>。MOOSE(误差最小平方和滤波器)算法<sup>[9]</sup>首先引入了相关滤波算法的框架,在特征上选取灰度特征,并将相关度的计算通过傅里叶变换引入到频域中,运行速度可达数百帧每秒。CSK算法在MOOSE算法的基础上引入岭回归的概念及核相关函数(循环矩阵),提升了算法的鲁棒性。Henriques等<sup>[10]</sup>使用CSK算法的框架,在特征上使用多通道的方向梯度直方图(HOG)特征,并采用Gauss核函数,提出了核相关滤波(KCF)算法,进一步提升了跟踪器的性能。但在复杂环境下KCF算法的鲁棒性较差,无法应对尺度变换的挑战。Danelljan等<sup>[11]</sup>在CSK算法的基础上使用二维的颜色特征,提出了CN算法,为特征的选择提供了新的思路。SAMF算法<sup>[12]</sup>对HOG特征、灰度特征、颜色特征进行线性融合,并在尺度变换中引入尺度池的概念,提升了算法应对复杂环境的能力,但其遍历的思想限制了算法的速度。Danelljan等<sup>[13-14]</sup>通过建立单独的尺度滤波器,提出DSST算法来处理尺度变换;通过减少滤波器的更新维数,提出fDSST算法,加快了尺度搜索的效率。为减少相关滤波算法的核相关矩阵引发边界效应,造成分类器性能降低的现象,Danelljan等<sup>[15]</sup>提出了空间正则化相关滤波跟踪算法(SRDCF),并在后续提出了拥有深度特征的deepSRDCF算法<sup>[16]</sup>和样本净化的SRDCFdecon算法<sup>[17]</sup>。然而,这些方法都会带来巨大的计算成本,实时性较差<sup>[18-19]</sup>。在复杂环境下使目标跟踪算法鲁棒性强的同时保持良好的运行速度依旧是个困难的问题<sup>[20-21]</sup>。

针对上述问题,本文在KCF算法的基础上,基于HOG特征、颜色特征和灰度特征构建滤波器,根据各个滤波器的响应值对三种特征进行自适应的动态加权,实现位置滤波器的位置估计;构建尺度滤波器,用于估计跟踪目标的尺度;同时根据跟踪置信度引入遮挡判断和模型更新策略,解决遮挡导致跟踪漂移的问题。根据跟踪目标的形状与移动速度,选择不同尺度的搜索区域,进一步提升算法的鲁棒性和运行速度。在目标跟踪数据集OTB2015的74组彩色视频集下进行测试,所提算法的平均距离精度和平均重叠精度分别达0.801和0.715,相较于KCF算法,分别提高了0.152和0.199。所提算法在速度上保持了相关滤波算法的优越性,仅在CPU的环境下达到了39.24 frame/s,满足了实时性的要求。

## 2 算法基础及改进

### 2.1 算法基础

KCF算法在分类器的计算中引入了循环矩阵,避免了矩阵的逆运算,减少了计算量。在应对非线性问

题时通过高斯核函数将其转换为线性问题,使得算法能够更好地处理一般性问题。通过训练样本可以得到线性回归方程,当线性回归与真实样本间误差最小时,寻找一个与 $\mathbf{x}$ 大小相同的分类器 $\omega$ :

$$\min \sum_i^n [f(\mathbf{x}_i) - y_i] + \lambda \|\omega\|^2, \quad (1)$$

式中:输出函数 $f(\mathbf{x}) = \omega^T \mathbf{x}$ ;  $\lambda$ 为正则化系数,防止过拟合。根据式(1)得到:

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

式中: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$ 为一个向量; $\mathbf{I}$ 为单位矩阵; $\mathbf{X}^H$ 为复共轭转置矩阵。循环矩阵 $\mathbf{X}$ 可以在傅里叶域中对角化,在处理非线性问题时,为了使目标与背景分离,需要将其转换到高维的空间中,设置权重向量:

$$\omega = \sum_i^n \alpha_i \varphi(\mathbf{x}_i). \quad (3)$$

同时,将核函数定义为 $\mathbf{k}(\mathbf{x}, \mathbf{x}') = \varphi^T(\mathbf{x}) \varphi(\mathbf{x}')$ ,利用离散傅里叶变换和循环矩阵的性质,可以求出岭回归的闭式解为

$$\alpha = (\mathbf{k} + \lambda \mathbf{I})^{-1} \mathbf{y}, \quad (4)$$

式中: $\mathbf{k}$ 为核函数。采用高斯核作为核函数,定义为

$$\mathbf{k}^{\mathbf{x}\mathbf{x}'} = \exp \left[ -\frac{1}{\delta^2} (\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2) - 2F^{-1}(\hat{\mathbf{x}} \odot \hat{\mathbf{x}}'^*) \right], \quad (5)$$

式中: $\delta$ 表示高斯核带宽;\*表示复共轭。在新的一帧图片中,通过检测搜索区域 $\mathbf{z}$ 中的响应最大位置,完成对目标的位置估计:

$$\hat{f}(\mathbf{z}) = \hat{\mathbf{a}} \odot \hat{\mathbf{k}}^{\mathbf{z}\mathbf{z}}, \quad (6)$$

同时为了应对跟踪目标的变化,KCF算法实时地对模板进行更新,更新方式为

$$\hat{\mathbf{a}}_t = (1 - \eta \times \sigma) \hat{\mathbf{a}}_{t-1} + \eta \times \sigma \times \hat{\mathbf{a}}_t, \quad (7)$$

式中: $\eta$ 为固定的更新参数。

### 2.2 多特征自适应融合

有效的目标特征选择和提取可以提高目标跟踪的精度和效率,单一的特征无法应对复杂多变的环境。HOG特征可以更好地适应背景相似性的干扰,颜色特征对运动模糊和形变较敏感,灰度特征具有更快的运行速度。为此,通过计算这三种特征滤波器的响应值并进行加权,充分发挥出各个特征的优势,更好地适应复杂的环境。对多特征进行提取时,式(5)可表示为

$$\mathbf{k}^{\mathbf{x}\mathbf{x}'} = \exp \left[ -\frac{1}{\delta^2} (\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2) - 2F^{-1}(\mathbf{D}) \right], \quad (8)$$

式中: $\mathbf{D} = \hat{\mathbf{x}} \odot \hat{\mathbf{x}}'^*$ ;  $\mathbf{x} = \alpha \mathbf{x}_{\text{HOG}} + \beta \mathbf{x}_{\text{CN}} + \gamma \mathbf{x}_{\text{gray}}$ ;参数 $\alpha, \beta, \gamma$ 均为加权系数。根据三种特征构建三个滤波器,分别计算出 $\hat{\mathbf{k}}_{\text{HOG}}^{\mathbf{z}\mathbf{z}}, \hat{\mathbf{k}}_{\text{CN}}^{\mathbf{z}\mathbf{z}}, \hat{\mathbf{k}}_{\text{gray}}^{\mathbf{z}\mathbf{z}}$ ,根据式(6),并进行傅里叶逆变换 $F^{-1}$ ,求出三种特征的响应值:

$$f_{\text{CN}}(\mathbf{z}) = F^{-1}(\hat{\mathbf{a}}_{\text{CN}} \odot \hat{\mathbf{k}}_{\text{CN}}^{\mathbf{z}\mathbf{z}}), \quad (9)$$

$$f_{\text{gray}}(\mathbf{z}) = F^{-1}(\hat{\mathbf{a}}_{\text{gray}} \odot \hat{\mathbf{k}}_{\text{gray}}^{\mathbf{z}\mathbf{z}}), \quad (10)$$

$$f_{\text{HOG}}(\mathbf{z}) = F^{-1}(\hat{\mathbf{a}}_{\text{HOG}} \odot \hat{\mathbf{k}}_{\text{HOG}}^{\mathbf{z}\mathbf{z}}). \quad (11)$$

对于三种特征的加权值,给予三种模式下的权重比。对于权重比,如果差距不大,则跟踪器不能很好地适应快速变换的环境,甚至带来相反的作用;而权重比值设置得近似,则体现不出特征动态加权的优势。因此采用三种不同的权重组合,如表 1 所示。

表 1 多特征融合三种加权比值

Table 1 Three weighted ratios of multi-feature fusion

Parameter	$\alpha$	$\beta$	$\gamma$
$\lambda_1$	0.5	0.4	0.1
$\lambda_2$	0.3	0.6	0.1
$\lambda_3$	1/3	1/3	1/3

如果  $f_{\text{HOG}}(\mathbf{z}) - f_{\text{CN}}(\mathbf{z}) > \zeta$ , 采用  $\lambda_1$  的加权模式, 否则采用  $\lambda_2$  的加权模式。最后判断灰度特征的响应值, 若  $f_{\text{gray}}(\mathbf{z})$  大于其余两种响应分数, 且差值大于  $\zeta$ , 采用  $\lambda_3$  的加权模式, 以此完成对多特征的实时动态加权。

### 2.3 尺度估计

在目标跟踪算法中, 位置估计和尺度估计都对算法性能有重要的影响, 受文献[13]的启发, 在位置估计的中心, 构建一维的尺度滤波器与目标物体的尺度相匹配。首先对目标物体检测出  $s=33$  组不同尺度的图

像块, 其中  $a=1.02$ , 形式为

$$a^n |n = \left\{ \left[ -\frac{s-1}{2}, \left[ -\frac{s-3}{2}, \dots, \left[ -\frac{s-31}{2}, \right. \right. \right. \right. \\ \left. \left. \left. \left[ -\frac{s-33}{2}, \left[ \frac{s-31}{2}, \dots, \left[ \frac{s-1}{2} \right] \right] \right] \right] \right\}. \quad (12)$$

对图像块进行调整, 并构建尺度金字塔; 再提取特征, 并利用图像块, 通过利用位置滤波器训练循环样本图像块的方式得到正则化最小二乘(RLS)分类器, 求解得到一维的尺度滤波器, 寻找最大置信度来完成尺度检测。对尺度滤波器模板进行更新, 得到的目标尺度为

$$(w_t, h_t) = \mu(w_t^*, h_t^*) + (1 - \mu)(w_{t-1}, h_{t-1}), \quad (13)$$

式中:  $(w_t^*, h_t^*)$  为最大置信度区域块的长和宽;  $(w_{t-1}, h_{t-1})$  为前一时刻目标物体的长和宽;  $\mu$  为阻尼因子, 使尺度平滑变化, 跟踪稳定。

### 2.4 遮挡判别及模型更新策略

为提升目标被遮挡时算法的鲁棒性, 受文献[22]的启发, 引入一种新的判断机制。从图 1 的 Joggle-1 序列的跟踪可视化响应图中可以看到, 遇到遮挡情况时, 响应图出现剧烈振荡、最大响应峰的位置不够明确等特点。

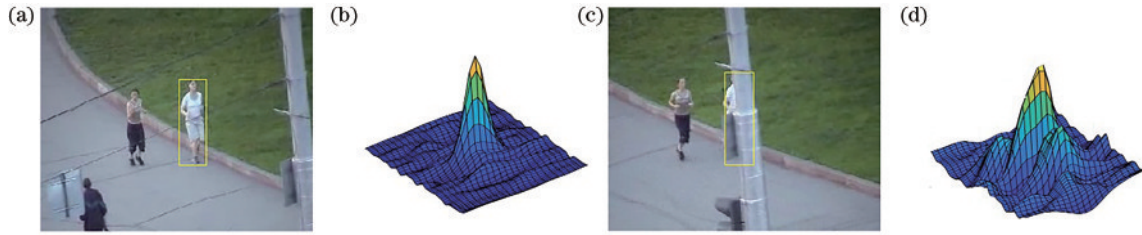


图 1 Joggle-1 序列下的响应图。(a) 第 4 帧无遮挡图; (b) 第 4 帧响应图; (c) 第 49 帧遮挡图; (d) 第 49 帧响应图

Fig. 1 Response diagrams corresponding to Joggle-1 sequence. (a) No occlusion image in frame 4; (b) response diagram of frame 4; (c) occlusion image in frame 49; (d) response diagram of frame 49

根据多特征的融合响应, 可以计算连续帧之间的  $\Delta E_{\text{APC}}$  值, 当其发生剧烈变化且大于阈值  $\epsilon$  时, 判断目标物体受到遮挡。

$$E_{\text{APC}} = \frac{|F_{\text{max}} - F_{\text{min}}|^2}{\text{mean} \left[ \sum_{w,h} (F_{w,h} - F_{\text{min}})^2 \right]}, \quad (14)$$

$$\Delta E_{\text{APC}} = E_{\text{APC}_t} - E_{\text{APC}_{t-1}}, \quad (15)$$

$$|\Delta E_{\text{APC}_t} - \Delta E_{\text{APC}_{t-1}}| > \epsilon, \quad (16)$$

式中:  $F_{\text{max}}$  为目标融合响应的最大值;  $F_{\text{min}}$  为目标融合响应的最小值;  $\Delta E_{\text{APC}}$  为连续帧之间的平均峰值相关能量(APCE)的变化值;  $\epsilon$  为遮挡阈值。当  $\Delta E_{\text{APC}}$  的连续差值大于  $\epsilon$  时, 判定目标被遮挡。传统的模型更新方式为固定参数的更新, 式(7)中  $\sigma=1$ , 容易造成误差累计, 降低跟踪精度。当判断目标被遮挡时, 若按照固定的学习率进行更新, 则模型污染的概率会增加, 此时  $\sigma=0$ , 模型不更新。

### 2.5 多尺度搜索区域

在当下的目标跟踪算法中, 跟踪目标的搜索区域

基本是固定的, 没有考虑到目标本身的特点。若搜索区域过大, 循环位移后得到的滤波器会使正样本的数量增多, 产生漂移现象, 同时背景信息增多, 误判的概率加大, 降低了搜索效率。当目标物体移动速度较快时, 小区域的搜索容易导致跟踪目标的丢失。针对复杂的环境, 定义三种不同的搜索区域, 分别为目标大小的 2 倍、2.5 倍、3 倍。当目标物体的移动速度较快, 大于阈值  $\tau$  时, 应当采取较大的区域进行搜索, 减少跟踪目标丢失的可能, 因此在对目标进行跟踪的过程中, 对目标的移动速度进行计算, 速度模型的计算公式为

$$v = \frac{(L_t - L_{t-1}) + (L_{t-1} - L_{t-2}) + \dots + (L_2 - L_1)}{N_{\text{frame}} - 1}, \quad (17)$$

式中:  $L_t - L_{t-1}$  为两帧之间的欧氏距离;  $N_{\text{frame}}$  为当前帧数。因为将放大的目标物体作为搜索区域, 对于不同形状的物体, 放大后形成的区域也存在着差异。引入目标物体的纵横比  $\Omega$ , 当  $\Omega$  大于一定的阈值时, 放大后的搜索区域较为狭长, 则更适合较大的放大倍数; 反



之,则可采取相对较小的搜索区域,提升算法的运行速度。结合  $\Omega$ , 决定搜索区域的范围:

$$A_{\text{area}} = \begin{cases} 3, & v > \tau \text{ or } \Omega > \eta_1 \\ 2.5, & \eta_1 < \Omega < \eta_2 \\ 2, & \Omega < \eta_1 \end{cases} \quad (18)$$

### 2.6 算法流程

所提算法首先提取跟踪目标的 HOG 特征、颜色

特征、灰度特征,根据滤波器的响应分数决定加权系数,通过位置滤波器得到融合响应,完成对目标的位置估计;根据置信度,判断是否对跟踪模型进行更新,并通过速度模型计算出目标的移动速度,结合目标特点  $\Omega$  选择适合的搜索区域;最后通过尺度滤波器得到的最优尺度,完成对跟踪目标的位置估计和尺度估计。所提算法流程如图 2 所示。

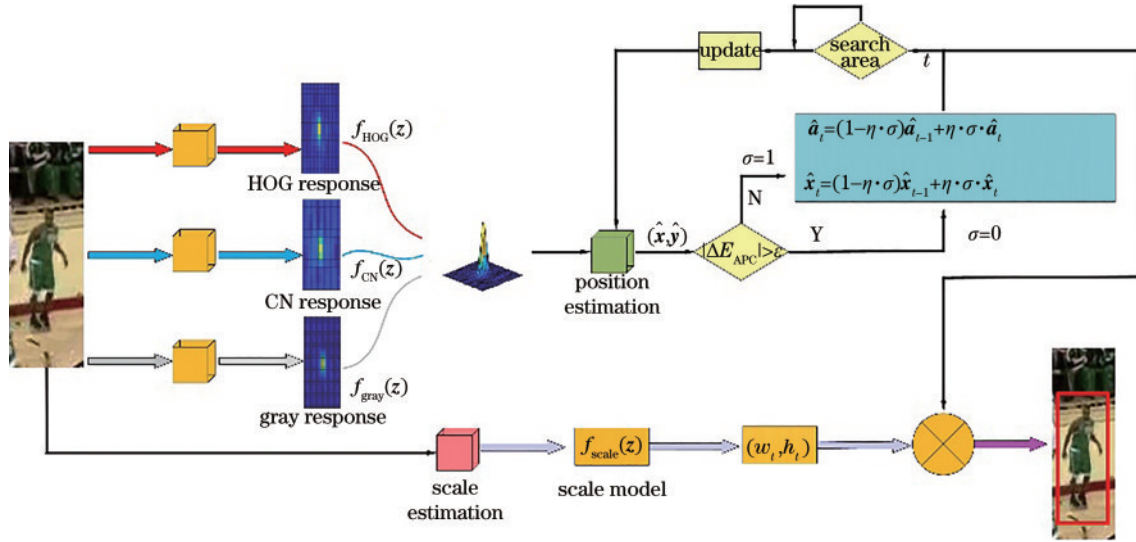


图 2 所提算法的流程

Fig. 2 Flowchart of the proposed algorithm

## 3 分析与讨论

### 3.1 实验环境及参数

实验是在 Intel Core i5-8400 CPU @ 2.80 GHz (主频为 2.8 GHz)、内存为 8 GB 的计算机, MATLAB R2019a 平台下完成的。基于区域选择的速度模型训练

的  $t=4$ , 能够降低复杂环境对跟踪器的影响, 体现构建速度模型的优势。在此前提下选取 OTB 数据集的视频序列, 对参数进行范围性选择, 检测的平均跟踪精度如图 3 所示。根据平均跟踪精度的表现, 选取参数  $\zeta=0.3, \epsilon=0.005, \tau=3, \eta_1=1.43, \eta_2=2.8, \mu=0.025, \eta=0.02$ 。

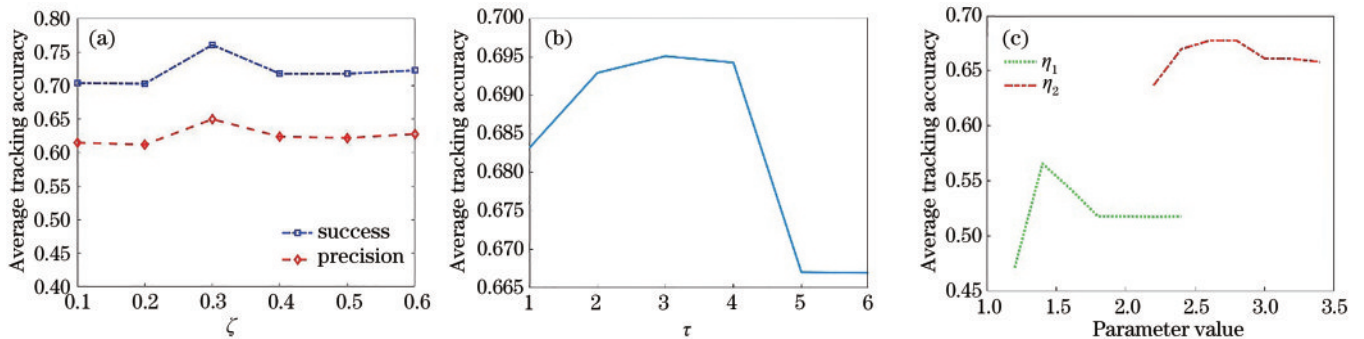


图 3 不同参数对跟踪器性能的影响。(a) 参数  $\zeta$ ; (b) 参数  $\tau$ ; (c) 参数  $\eta_1$  和  $\eta_2$

Fig. 3 Influence of different parameters on tracker performance. (a) Parameter  $\zeta$ ; (b) parameter  $\tau$ ; (c) parameters  $\eta_1$  and  $\eta_2$

为验证算法在各个挑战下的整体效果,同时防止特征选取上非彩色图片对整体的影响,选取 OTB50 的 41 组彩色数据集和 OTB2015 的 74 组彩色数据集进行测试。为了验证跟踪器的性能,采取平均距离精度 (precision)、平均重叠精度 (success rate) 和中心位置误差 (CLE) 对跟踪器的性能进行评价。平均距离精度为

跟踪的中心像素  $(x_t, y_t)$  与真实的物体中心  $(x_p, y_p)$  的欧氏距离小于 20 pixel 的帧数占总帧数的百分比。中心位置误差的计算公式为

$$E_{CL} = \sqrt{(x_p - x_t)^2 + (y_p - y_t)^2} \quad (19)$$

平均重叠精度是指真实的区域  $R_p$  与跟踪的区域

$R_i$  之间的重叠率大于 0.5 的帧数占总帧数的百分比, 表达式为

$$S = \frac{|R_p \cap R_i|}{|R_p \cup R_i|} \quad (20)$$

### 3.2 动态权重特征融合算法对比实验

为了验证动态融合特征的效果, 在其余参数和方式不变的情况下, 更改算法对特征的融合方式。OURS 为所提动态特征加权算法, OURS1 为模式  $\lambda_1$  下

的特征加权方式, OURS2 为模式  $\lambda_2$  下的特征加权方式, OURS3 为模式  $\lambda_3$  下的特征加权方式。在 OTB50 的 41 组彩色数据集下, 4 种不同的加权方式的结果如图 4 所示, 特征动态加权融合的方式在平均距离精度和平均重叠精度上要优于三种固定的特征加权方式, 分别达 0.777 和 0.674。因此所提特征动态加权方式可以在一定程度上提高跟踪器的性能。

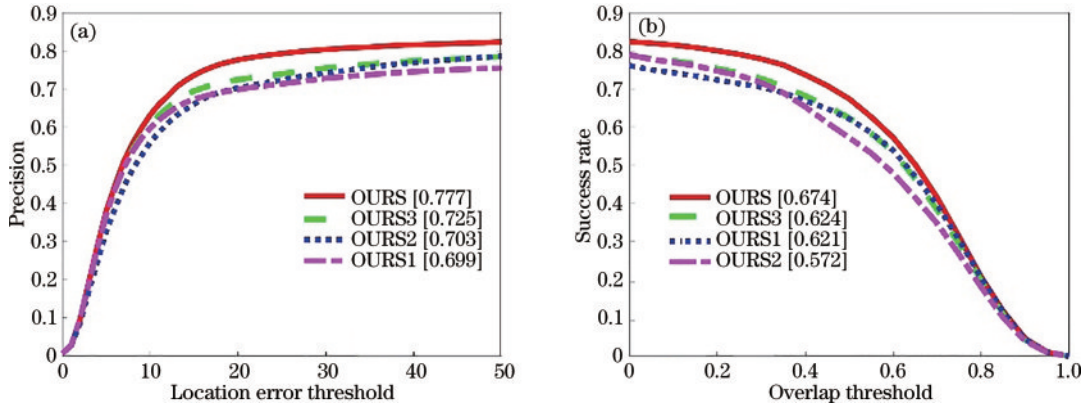


图 4 不同的特征加权方式的平均距离精度和平均重叠精度。(a) 平均距离精度; (b) 平均重叠精度  
Fig. 4 Precision and success rate of different feature weighted methods. (a) Precision; (b) success rate

### 3.3 尺度变换性能对比实验

为了验证所提算法在尺度上的表现, 选取了数据集中集中在尺度变换上具有代表性的 Blurcar2、Doll、Carscale 三组序列进行测试。先提取跟踪器在跟踪过

程的每一帧的尺度, 然后与算法 SAMF<sup>[12]</sup>、SRDCF<sup>[15]</sup> 的跟踪结果及物体的实际尺度 ground\_truth 进行对比, 如图 5 所示。

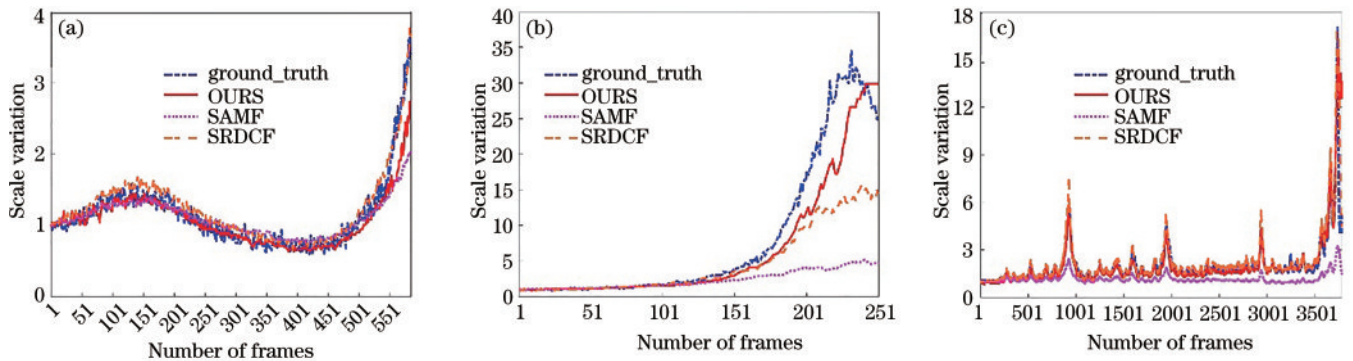


图 5 不同算法在 3 组不同视频序列下的尺度变化。(a) Blurcar2; (b) Doll; (c) Carscale

Fig. 5 Scale changes of different algorithms under three groups of different video sequences. (a) Blurcar2; (b) Doll; (c) Carscale

Blurcar2 尺度变换的范围相对平缓, 在 0.5 到 4 之间, 所提算法对尺度的表现良好。在 Doll 这个长时间的数据集下有多次的尺度变换, 所提算法也能够较好地适应, SAMF 算法应对尺度变换的表现不佳。Carscale 的尺度变换范围较大, 最大尺度可以达原始帧的 30 多倍, 在 200 帧后, 所提算法对尺度表现最佳, SAMF 和 SRDCF 算法则不能很好适应。实验结果表明, 所提算法在应对尺度变化时表现更好。表 2 为所提算法在三组数据集上的跟踪结果。

### 3.4 遮挡判别验证实验

对遮挡判别的条件进行验证, 结果如图 6 所示。

表 2 所提算法在 3 组尺度序列下的指标

Table 2 Index of the proposed algorithm under three groups of scale sequences

Sequence	Precision	Success rate	Speed / (frame · s <sup>-1</sup> )
Blurcar2	0.999	1.000	44.06
Carscale	0.897	0.980	60.42
Doll	0.961	0.966	57.55

对于 Juggle-1 和 Juggle-2 这两个数据集, 在跟踪的过程中出现明显的遮挡。Juggle-1 数据集在 42 帧开始逐渐出现明显的遮挡情况, 到 65 帧结束遮挡, 而此时  $\Delta E_{APC}$

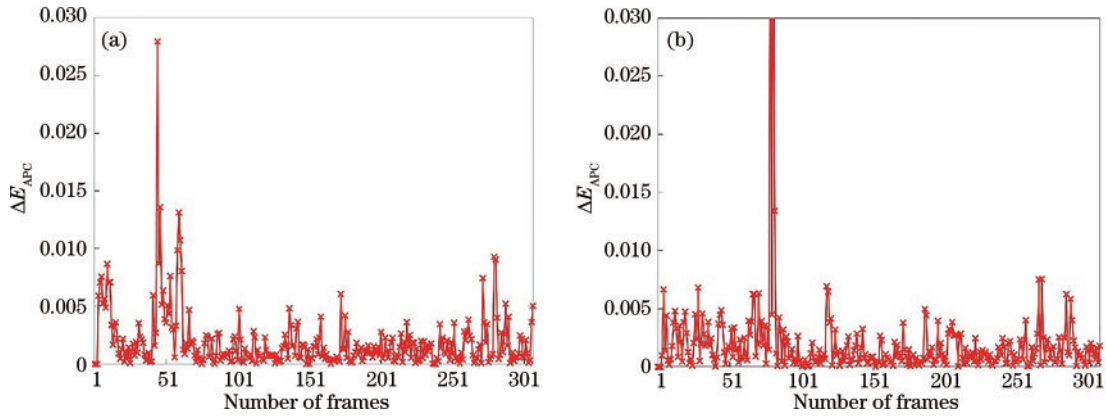


图 6 Joggle序列下每帧图像的 $\Delta E_{APC}$ 值和变化。(a)Joggle-1; (b)Joggle-2

Fig. 6  $\Delta E_{APC}$  value and changes of each frame under the Joggle sequences. (a) Joggle-1; (b) Joggle-2

出现明显的变化,第45帧达到最大。根据事先设定的阈值 $\epsilon=0.005$ ,可以避免模型的严重污染。在短时间的遮挡结束后,没有跟踪漂移和失败情况。若选取更小的阈值,则会多次暂停模型更新,导致模型退化,跟踪效果不佳;若选取更大的判断阈值,会造成模型对遮挡的判定不敏感,导致模型污染,降低跟踪器的性能。在Joggle-2数据集中,障碍物的遮挡出现在第65帧,结束于第81帧,如图6(b)所示,所提算法的遮挡判断有效。所提算法在应对30帧左右的遮挡挑战时能够很好地避免模型污染和退化,遮挡挑战后依旧可以保持良好的跟踪效果。

### 3.5 多尺度搜索区域对比

为了验证搜索区域对跟踪效果的影响,比较构建的速度模型和形状特点的优势,选择多尺度搜索区域的算法 OURS 和固定搜索区域的算法 OURS4、OURS5、OURS6 进行对比。对比算法的搜索区域尺寸分别对应着目标的2倍、2.5倍、3倍。同时,选取 SRDCF、SRDCFdecon、SAMF 算法进行纵向比较。在 OTB50 彩色数据集下,不同算法的平均重叠精度和运行速度如表3所示。多尺度搜索区域的算法的平均重叠精度达0.674,优于固定倍数的搜索区域算法和对比算法,验证了多尺度搜索区域算法的有效性。

表3 搜索区域算法的结果对比

Table 3 Result comparison of search area algorithms

Parameter	OURS	OURS4	OURS5	OURS6	SRDCF	SRDCFdecon	SAMF
Success rate	0.674	0.564	0.626	0.656	0.634	0.677	0.576
Speed / (frame·s <sup>-1</sup> )	39.07	52.10	44.24	35.52	2.05	5.18	14.54

### 3.6 中心误差分析

在Basketball和Faceoccl两个平均距离精度表现相近的视频序列下,对所提算法与KCF<sup>[10]</sup>、SAMF<sup>[12]</sup>、SRDCF<sup>[15]</sup>、SRDCFdecon<sup>[17]</sup>算法的CLE进行对比。

CLE越小,算法的精度和稳定性就越高。如图7(a)的Basketball序列所示,所提算法的CLE没有超过20,且变化的幅度最小。在该视频集的光照变换(IV)、遮挡(OCC)、形变(DEF)、平面外旋转(OPR)、背景杂

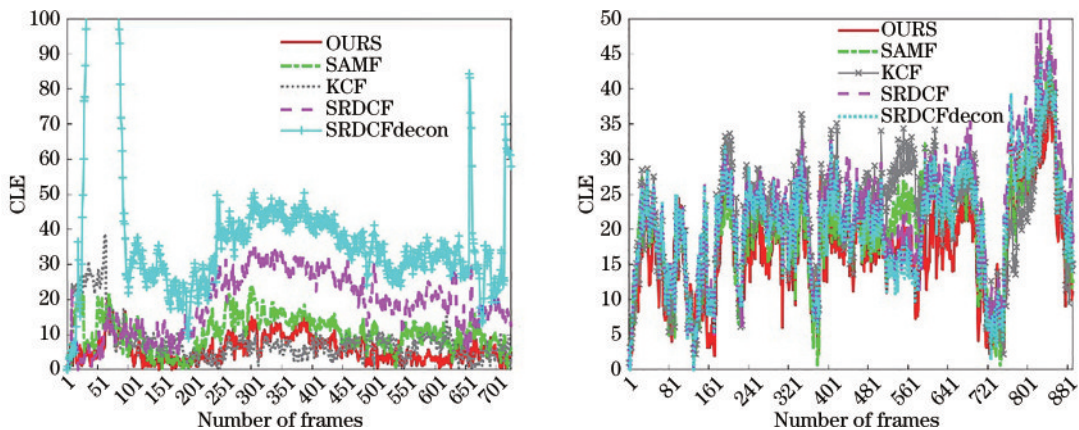


图 7 两组视频下CLE的变化。(a)Basketball; (b)Faceoccl

Fig. 7 CLE changing between two sets of videos. (a) Basketball; (b) Faceoccl



乱(BC)挑战下,所提算法表现最好。在 Faceoccl 视频序列下有着 OCC、IV 等挑战,如图 7(b)所示,所提算法的平均 CLE 为 17.4,小于相近 SAMF 和 SRDCF 算法的 19.8 和 22.3,进一步验证所提算法的稳定性。

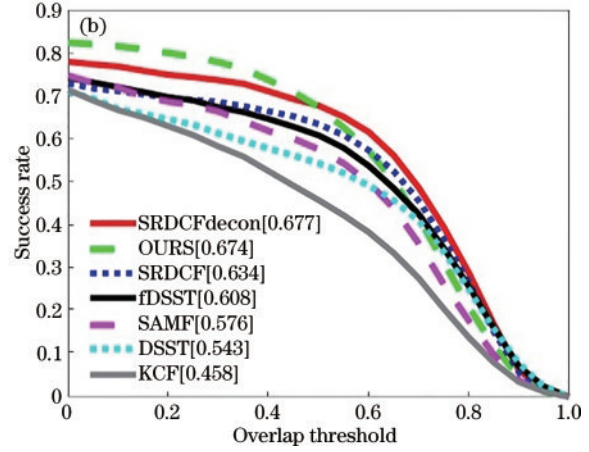
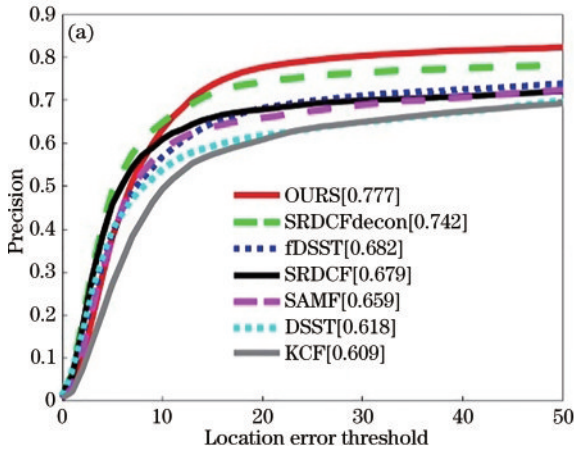


图 8 7 种算法在 OTB50 数据集中的距离误差和重叠精度。(a) 平均距离精度; (b) 平均重叠精度  
Fig. 8 Precision and success rate of seven algorithms on OTB50 dataset. (a) Precision; (b) success rate

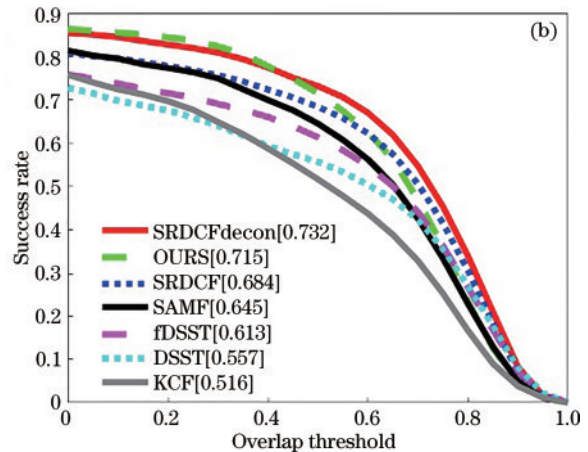
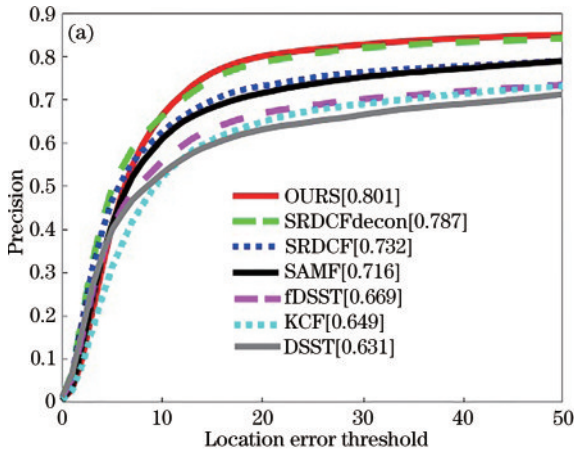


图 9 7 种算法在 OTB2015 中的距离误差和重叠精度曲线。(a) 平均距离精度; (b) 平均重叠精度  
Fig. 9 Precision and success rate of seven algorithms on OTB2015 dataset. (a) Precision; (b) success rate

在 OTB50 数据集上,所提跟踪算法 Ours 的平均距离精度为 0.777,平均重叠精度为 0.674,较 KCF 算法在平均距离精度上提升了 0.168,平均重叠精度提升了 0.216;较 SRDCFdecon 算法在平均距离精度上提升了 3.5 个百分点,平均重叠精度相近。在 OTB2015 数据集上,所提算法 Ours 的平均距离精度为 0.801,平均重叠精度为 0.715,较 SRDCFdecon 算法在平均距离精度上提升了 1.4 个百分点。

同时,为了检测算法在 OTB2015 数据集下对 11 种不同挑战的表现,表 4 和表 5 列出了 KCF<sup>[10]</sup>、SAMF<sup>[12]</sup>、DSST<sup>[13]</sup>、fDSST<sup>[14]</sup>、SRDCF<sup>[15]</sup>、SRDCFdecon<sup>[17]</sup> 算法在 11 种挑战下的平均距离精度和平均重叠精度。每种属性下的第一名用双下划线标注,第二用单下划线标注。从表 4 和表 5 可以看出:所提算法在目标形变

### 3.7 定量分析

对所提算法与 KCF<sup>[10]</sup>、SAMF<sup>[12]</sup>、DSST<sup>[13]</sup>、fDSST<sup>[14]</sup>、SRDCF<sup>[15]</sup>、SRDCFdecon<sup>[17]</sup> 6 种主流的性能较好的算法在 OTB50 和 OTB2015 彩色测试集上进行测试,结果如图 8 和图 9 所示。

(DEF)、遮挡(OCC)、平面旋转(PR)、超出视野(OV)、低分辨率(LR)的挑战下表现最好;在遮挡、超出视野和低分辨率的挑战上,所提算法的平均距离精度分别超越 SRDCFdecon 算法 3.0 个百分点、6.9 个百分点、10.7 个百分点;在平均重叠精度上,所提算法对目标形变、超出平面旋转和平面内旋转的挑战上表现最好,分别超越 SRDCFdecon 算法 2.6 个百分点、0.7 个百分点、3.9 个百分点。

为了检测算法的跟踪速度,给出了各算法在不同数据集上的运行速度,如表 6 所示。所提算法在 OTB50 和 OTB2015 两组数据集的帧率分别达 39.07 frame/s 和 39.24 frame/s,满足了跟踪算法的实时性要求。在平均距离精度和平均重叠精度上表现较好的 SRDCFdecon 和 SRDCF 两个算法在 OTB50、OTB2015

表 4 7 种目标跟踪算法在 11 不同挑战属性下的平均距离精度

Table 4 Precision of seven target tracking algorithms at 11 different challenge attributes

Algorithm	IV	DEF	SV	OCC	MB	FM	IPR	OPR	OV	BC	LR
OURS	<u>0.760</u>	<u>0.741</u>	<u>0.755</u>	<u>0.764</u>	0.787	0.732	<u>0.805</u>	<u>0.790</u>	<u>0.630</u>	<u>0.808</u>	<u>0.707</u>
SRDCFdecon	<u>0.803</u>	0.730	<u>0.771</u>	<u>0.734</u>	<u>0.808</u>	<u>0.763</u>	<u>0.715</u>	<u>0.759</u>	0.561	<u>0.814</u>	0.600
SRDCF	0.715	0.693	0.689	0.673	0.734	<u>0.744</u>	0.631	0.675	0.520	0.676	0.594
SAMF	0.648	0.650	0.663	0.706	0.674	0.657	0.665	0.701	<u>0.605</u>	0.630	<u>0.645</u>
fDSST	0.715	0.589	0.628	0.612	0.658	0.660	0.680	0.617	0.476	0.720	0.609
DSST	0.677	0.509	0.578	0.552	0.585	0.548	0.644	0.598	0.357	0.648	0.510
KCF	0.676	0.573	0.591	0.603	0.622	0.614	0.651	0.624	0.428	0.677	0.538

表 5 7 种目标跟踪算法在 11 不同挑战属性下的平均重叠精度

Table 5 Success rate of seven target tracking algorithms at 11 different challenge attributes

Algorithm	IV	DEF	SV	OCC	MB	FM	IPR	OPR	OV	BC	LR
OURS	<u>0.692</u>	<u>0.667</u>	<u>0.633</u>	<u>0.687</u>	<u>0.742</u>	0.683	<u>0.688</u>	<u>0.693</u>	<u>0.531</u>	<u>0.720</u>	0.501
SRDCFdecon	<u>0.748</u>	<u>0.641</u>	<u>0.712</u>	<u>0.694</u>	<u>0.795</u>	<u>0.729</u>	<u>0.649</u>	<u>0.686</u>	<u>0.561</u>	<u>0.740</u>	<u>0.571</u>
SRDCF	0.675	0.626	0.631	0.640	0.719	<u>0.712</u>	0.574	0.607	0.492	0.634	<u>0.581</u>
SAMF	0.584	0.555	0.562	0.641	0.660	0.598	0.602	0.622	0.490	0.596	0.539
fDSST	0.645	0.525	0.547	0.553	0.633	0.641	0.610	0.545	0.438	0.645	0.547
DSST	0.601	0.437	0.489	0.491	0.568	0.515	0.552	0.506	0.305	0.535	0.423
KCF	0.503	0.434	0.396	0.475	0.565	0.534	0.515	0.477	0.376	0.576	0.325

表 6 7 种目标跟踪算法的运行速度

Table 6 Running speed of seven target tracking algorithms

Algorithm	Speed (frame·s <sup>-1</sup> )	
	OTB50	OTB2015
OURS	39.0778	39.2448
SRDCFdecon	2.0497	1.9694
SRDCF	5.1833	4.871
SAMF	14.5473	13.933
fDSST	37.4683	31.6959
DSST	8.4053	7.0361
KCF	198.4457	173.5403

两组数据集上的运行速度仅仅为 2.0497 frame/s、5.1833 frame/s 和 1.9694 frame/s、4.871 frame/s。

### 3.8 定性分析

从 OTB2015 中选取 7 组数据集 (Box、Dragonbaby、Bird2、Panda、Carscale、Soccer、Tiger2) 进行对比分析。这 7 组数据集分别代表着短时间遮挡、运动模糊、快速移动、低分辨率、尺度变换、背景混乱、光照变换的挑战。

如图 10(a) 所示, 目标在第 460 帧时遇到了横杠的

短时遮挡, 对跟踪的效果产生影响, 并造成跟踪模型的污染, 导致后续无法找回目标, 从后续的跟踪过程可以看到, 仅所提算法和 SAMF 算法准确跟踪到目标。从图 10(b) 可以看到, 面对 Dragonbooby 序列的运动模糊、快速移动和光照的多种挑战下, 所提算法的跟踪效果最能让人满意。如图 10(c) 所示, 在 Bird2 序列中的第 57 帧, 目标物体快速改变运动方向, 且在快速移动和相似目标的遮挡和干扰影响下, 所提算法的跟踪效果最好。从图 10(d) 可以看到, 在低分辨率情况的 Panda 序列下, 在第 191 帧时, SRDCFdecon 丢失了目标物, 在随后的跟踪过程后目标的移动速度加快, 到第 535 帧时, 大部分算法跟踪失败, 614 帧后, 仅所提算法依旧保持较好的跟踪效果。如图 10(e) 所示, 在快速的尺度变化挑战下, 第 106 帧到第 223 帧目标放大了几十倍, 总体上看所提算法对物体尺度变化有较好的适应性。在图 10(f) 中, Soccer 序列有较强的背景干扰和相似目标物, 且在第 76 帧开始出现大量的红色背景干扰, 对特征的提取有较强的挑战, 在第 109 帧后还有目标相似物的影响, 仅所提算法有较好的跟踪效果。如图 10(g) 所示, 在 Tiger2 数据集光照变换和快速移动的多种因素影响下, 所提算法的表现相对于其余算法表现良好。





图 10 7 种算法在不同视频序列上的对比。(a) Box; (b) Dragonbaby; (c) Bird2; (d) Panda; (e) Carscale; (f) Soccer; (g) Tiger2  
 Fig. 10 Comparison of seven algorithms on different video sequences. (a) Box; (b) Dragonbaby; (c) Bird2; (d) Panda; (e) Carscale;  
 (f) Soccer; (g) Tiger2

## 4 结 论

提出了一种改进的相关滤波目标跟踪算法。首先,对 HOG 特征、颜色特征、灰度特征进行动态自适应的加权融合,完成位置滤波器的位置估计;然后,在目标跟踪的中心位置构建尺度滤波器,提升了算法在尺度上的表现;同时,提出一种新的遮挡判别,对跟踪

模型进行选择性的更新,并针对不同物体的特性,给予相适应的搜索区域。通过实验表明,所提算法在应对复杂环境时表现更佳,尤其在应对短时的遮挡、尺度变换、低分辨率、目标形变等挑战时表现更好。在跟踪速度上,所提算法远远优于对比的其他算法,满足了实时性的要求。然而,在面对 30 帧以上的长时间遮挡时如何保持算法良好的跟踪效果,还须进一步研究。

## 参 考 文 献

- [1] 孟磊, 杨旭. 目标跟踪算法综述[J]. 自动化学报, 2019, 45(7): 1244-1260.  
Meng L, Yang X. A survey of object tracking algorithms [J]. Acta Automatica Sinica, 2019, 45(7): 1244-1260.
- [2] Shu Q Z, Lai H C, Wang L J, et al. Multi-feature fusion target re-location tracking based on correlation filters[J]. IEEE Access, 2021, 9: 28954-28964.
- [3] 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.
- [4] 刘宗达, 董立泉, 赵跃进, 等. 视频中快速运动目标的自适应模型跟踪算法[J]. 光学学报, 2021, 41(18): 1815001.  
Liu Z D, Dong L Q, Zhao Y J, et al. Adaptive model tracking algorithm for fast-moving targets in video[J]. Acta Optica Sinica, 2021, 41(18): 1815001.
- [5] Zhao F, Hui K D, Wang T T, et al. A KCF-based incremental target tracking method with constant update speed[J]. IEEE Access, 2021, 9: 73544-73560.
- [6] Yang Y X, Xing W W, Zhang S L, et al. Visual tracking with long-short term based correlation filter[J]. IEEE Access, 2020, 8: 20257-20269.
- [7] Zhang S L, Lu W, Xing W W, et al. Learning scale-adaptive tight correlation filter for object tracking[J]. IEEE Transactions on Cybernetics, 2020, 50(1): 270-283.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] Danelljan M, Khan F S, Felsberg M, et al. Adaptive color attributes for real-time visual tracking[C]//2014 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH, USA. New York: IEEE Press, 2014: 1090-1097.
- [12] Li Y, Zhu J K. A scale adaptive kernel correlation filter tracker with feature integration[M]//Agapito L, Bronstein M M, Rother C. Computer vision-ECCV 2014 workshops. Lecture notes in computer science. Cham: Springer, 2015, 8926: 254-265.
- [13] 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. London: British Machine Vision Association, 2014: 1-11.
- [14] Danelljan M, Hager G, Khan F S, et al. Discriminative scale space tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(8): 1561-1575.
- [15] Danelljan M, Häger G, Khan F S, et al. Learning spatially regularized correlation filters for visual tracking [C]//2015 IEEE International Conference on Computer Vision, December 7-13, 2015, Santiago, Chile. New York: IEEE Press, 2015: 4310-4318.
- [16] Danelljan M, Häger G, Khan F S, et al. Convolutional features for correlation filter based visual tracking[C]//2015 IEEE International Conference on Computer Vision Workshop, December 7-13, 2015, Santiago, Chile. New York: IEEE Press, 2015: 621-629.
- [17] Danelljan M, Häger G, Khan F S, et al. Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 1430-1438.
- [18] Wu L Y, Xu T F, Zhang Y S, et al. Multi-channel feature dimension adaption for correlation tracking[J]. IEEE Access, 2021, 9: 63814-63824.
- [19] Yin X L, Liu G X, Ma X H. Fast scale estimation method in object tracking[J]. IEEE Access, 2020, 8: 31057-31068.
- [20] Li S M, Chu J, Zhong G C, et al. Robust visual tracking with occlusion judgment and re-detection[J]. IEEE Access, 2020, 8: 122772-122781.
- [21] Sun J P, Ding E J, Sun B, et al. Adaptive kernel correlation filter tracking algorithm in complex scenes[J]. IEEE Access, 2020, 8: 208179-208194.
- [22] 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.