

基于核相关滤波的长期目标跟踪算法

茅正冲, 陈海东*

江南大学轻工过程先进控制教育部重点实验室, 江苏 无锡 214122

摘要 针对传统核相关滤波器(KCF)无法处理严重遮挡及光照变化等问题,提出一种结合快速角点检测与双向光流法的长期 KCF 跟踪算法。首先利用 KCF 跟踪器在目标位置上提取融合方向梯度直方图特征、颜色属性特征和灰度特征的多通道特征,计算输出响应图并得到所跟踪目标的峰值旁瓣比(PSR),然后通过比较 PSR 与经验阈值来判断目标是否被遮挡;当目标出现遮挡时,在快速角点检测的角点基础上利用双向光流法重新检测下一帧目标位置,并采用一种新模板更新策略来应对严重遮挡。与其他算法进行对比实验,验证了本文算法对处理遮挡和光照变化具有高效性及稳健性。

关键词 傅里叶光学; 目标跟踪; 核相关滤波; 快速角点检测; 双向光流; 遮挡

中图分类号 TP391

文献标识码 A

doi: 10.3788/LOP56.010702

Long-Term Object Tracking Algorithm Based on Kernelized Correlation Filter

Mao Zhengchong, Chen Haidong*

Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education,
Jiangnan University, Wuxi, Jiangsu 214122, China

Abstract Focusing on the issue that the traditional kernelized correlation filter (KCF) has poor performance in handling heavy occlusion and illumination variations, a long-term KCF tracking algorithm is proposed combined with fast corner detection and bidirectional optical flow method. First, the KCF tracker is used to extract the multi-channel features of the histogram of gradient, color attributes, and gray features at the target location. The output response map is calculated and the peak sidelobe ratio (PSR) of the tracked target is obtained. The PSR and the empirical threshold determine whether the target is occluded by comparison. When the target is occluded, the bidirectional optical flow method is used to redetect the target position of the next frame based on the corner points detected by the fast corner detection, and a new template updating strategy is adopted to deal with the heavy occlusion. Compared with other algorithms, the proposed algorithm is effective and robust to the processing of occlusion and illumination variations.

Key words Fourier optics; target tracking; kernelized correlation filter; fast corner detection; bidirectional optical flow; occlusion

OCIS codes 070.4550; 150.1135; 100.4999

1 引言

目标跟踪一直是计算机视觉领域中的焦点,在室内视频监控、运动状态分析、临床医学成像等领域有着广泛应用^[1]。先确定第一帧图像中目标初始位置,然后跟踪器对后续的每一帧图像中目标的位置进行估计。尽管近年来的研究取得了显著的进展,

但仍然充满挑战,目前尚无单一的方法能够成功地处理所有场景^[2]。许多因素都会影响跟踪算法的性能,如目标尺度与光照变化、目标旋转与遮挡、快速移动、相机抖动和背景干扰等^[3]。如何从第一帧中提取有效的外观信息并建立模型是跟踪中需要考虑的首要关键因素^[4]。

借助于快速离散傅里叶变换在频域内的有效计

收稿日期: 2018-06-13; 修回日期: 2018-06-28; 录用日期: 2018-07-18

基金项目: 国家自然科学基金(60973095)、江苏省产学研联合创新资金—前瞻性联合研究项目(BY2015019-29)

* E-mail: 714778302@qq.com

算,近年来很多研究将相关滤波应用到目标跟踪。视觉目标追踪(VOT)2014年挑战结果^[5]显示,基于相关滤波器的跟踪算法已成为近年来最成功的算法之一。例如,Bolme等^[6]提出了对目标外观有较强的适应性的最小输出平方误差和(MOSSE)跟踪算法;Henriques等^[7]提出建立多通道特征模型,利用循环采样结构来训练正则化最小二乘(RLS)分类器的检测跟踪(CSK)算法,取得了较好的跟踪效果;Danelljan等^[8]将颜色属性(CN)信息进行主成分分析(PCA)降维与灰度特征叠加,得到了更加精确、稳健的目标表现模型。然而对于遮挡问题,均存在目标易丢失、难恢复等问题。朱明敏等^[9]在核相关滤波(KCF)^[10]的基础上增加光流金字塔作为检测器来处理遮挡;王艳川等^[11]利用局部稀疏表示模型的重构残差进行遮挡判断,并在遮挡时利用在线支持向量机进行检测,取得了较好的效果。

本文在KCF的基础上扩展了颜色特征及灰度特征,利用峰值旁瓣比(PSR)值作为遮挡判断,使用快速角点检测与双向光流结合的方法重检测,严重遮挡时停止更新模板,具有良好的稳健性。

2 KCF跟踪算法的改进

2.1 KCF跟踪算法

KCF算法将目标区域进行循环位移采集训练图像,针对线性岭回归判别函数 $f(t) = \mathbf{w}^T t$,通过训练样本 g_i 找到与其对应的回归目标 e_i 最小平方误差的权值 w_i 。最小代价函数表示为

$$\min_{\mathbf{w}} \sum_i [f(g_i) - e_i]^2 + \lambda \|\mathbf{w}\|^2, \quad (1)$$

式中 λ 是正则化参数,防止过拟合。

在岭回归方法中引入核函数 $k(\mathbf{g}, \mathbf{g}') = \langle \phi(\mathbf{g}), \phi(\mathbf{g}') \rangle$,将 \mathbf{w} 映射到高维空间后表示为 $\mathbf{w} = \sum_i a_i \phi(g_i)$,因此(1)式需要优化的变量由 \mathbf{w} 转变为 \mathbf{a} ,表示为

$$\mathbf{a} = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}(\mathbf{e})}{\mathcal{F}[k(\mathbf{g}, \mathbf{g}') + \lambda]} \right\}, \quad (2)$$

式中 \mathcal{F}^{-1} 和 \mathcal{F} 代表离散傅里叶逆变换及变换。

根据分类器训练结果,对于下一帧新输入图像块 \mathbf{z} 求得分类器的响应为

$$f(\mathbf{z}) = \mathbf{w}^T \mathbf{z} = \sum_i a_i k(g_i, z_i), \quad (3)$$

响应矩阵中的最大值所在位置即为下一帧中的目标中心。

2.2 多特征融合

KCF通过高斯核函数计算输入的多通道特征,高斯核 k 计算公式为

$$k(\mathbf{g}, \mathbf{g}') = \exp \left\{ -\frac{1}{\sigma^2} [\|\mathbf{g}\|^2 + \|\mathbf{g}'\|^2 - 2\mathcal{F}^{-1}(\mathbf{F})] \right\}, \quad (4)$$

式中 $\mathbf{F} = \mathcal{F}(\mathbf{g}) \cdot \mathcal{F}(\mathbf{g}')$, $\mathbf{g} = [g_1, g_2, \dots, g_i]$ 为 i 维多通道特征, \cdot 代表元素点乘。

根据(4)式,分别提取灰度特征、改进梯度方向直方图(FHOG)特征和颜色名称空间(CN)特征,最终级联成特征向量 \mathbf{g} ,特征维度 $i = 42$ 。

1) 灰度特征。通过将图像转换成灰度图像,以每个像素点的灰度值作为图像的特征。

2) FHOG特征。方向梯度直方图(HOG)特征将图像划分为若干个细胞单元(cell),每个cell为 $2 \times 22 \times 2$ 个像素大小,取 2×2 个邻近的cell组成一个块(block),在 $[0, \pi)$ 的方向上划分9个无符号梯度方向,每个block就会得到 $4 \times 9 = 36$ 维特征。FHOG在此基础上对 $[0, 2\pi)$ 的方向上划分18个有符号的梯度方向,得到 $4 \times (18 + 9) = 108$ 维特征,显然特征维度过大且复杂,故对每个方向的4个cell的通道值进行累加得到27维特征,并对每个cell中提取的9个无符号梯度方向通道值进行累加得到4维特征,最终组合成31维特征通道,在描述外观特征上效果更好。

3) CN特征。将颜色划分为11种:黑、蓝、棕、灰、绿、橙、粉、紫、红、白、黄。Weijer等^[12]利用概率浅语义分析(PLSA),从Google带标签商品数据集中学习,将RGB空间以每8明度划分为32个区间,共获得 $32^3 = 32768$ 种颜色,建立了RGB空间映射到颜色命名空间的字典(32768×11)。文献[8]中通过类似PCA降维的方法找到一个标准正交基下的投影矩阵,并重新计算新的外观特征映射(32786×10)。

2.3 目标再检测

使用PSR来判断目标跟踪状态,通过检测器不断纠正模型,从而增强了稳健性。PSR定义为

$$R_{\text{PSR}} = \frac{p_{\text{max}} - \mu}{\sigma}, \quad (5)$$

式中 p_{max} 为响应矩阵中的最大值, μ 为所有响应值的平均值, σ 为响应值的标准差。当 R_{PSR} 小于阈值 T_1 时,判断为出现遮挡,对下一帧位置进行预估。

传统光流跟踪以每5个像素点为间距进行均匀采样,并将所有采样点作为跟踪点,不仅计算量大而

且可能忽略大量关键点。研究表明,角点在光流中位移相对稳定,为提高运算速度,本文采用快速角点检测的方法,主要步骤如下:

1) 如图1以像素 p 为中心,在半径为3的圆上获取16个像素点定义为 p_1, p_2, \dots, p_{16} 。

2) 计算 p_1, p_5, p_9, p_{13} 与中心 p 之间的像素

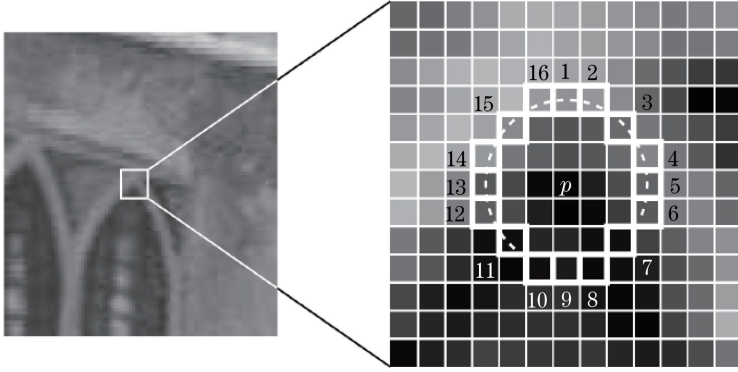


图1 角点检测图

Fig. 1 Corner detection

光流法主要基于三个假设:1) 相邻两帧目标相对位移所对应位置范围内亮度不变;2) 时间连续或者运动为“小运动”;3) 同一表面上邻近的点保持相邻且具有相似运动。

由假设1)可得:

$$I(x, y, t) = I(x + dx, y + dy, t + dt), \quad (6)$$

式中 x, y 为横纵坐标, $I(x, y, t)$ 为对应像素点亮度, $I(x + dx, y + dy, t + dt)$ 为下一帧对应像素点亮度。将(6)式进行泰勒级数展开得:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + O(\partial^2), \quad (7)$$

由(6)式和(7)式联立方程组计算可得:

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0. \quad (8)$$

由假设2)可将(8)式进行转换得:

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t, \quad (9)$$

式中 $I_x = \partial I / \partial x, I_y = \partial I / \partial y, I_t = \partial I / \partial t$ 分别表示亮度在 x, y 方向上的梯度及前后帧之间像素值变化率, $u = \partial x / \partial t, v = \partial y / \partial t$ 分别表示像素点在 x, y 方向上光流位移量。

对于(9)式,求解两个未知数显然是不可能的,利用假设3),将(9)式转换为

$$AD = b, \quad (10)$$

由最小二乘法得求解(10)式为

差,若至少有3个值大于给定阈值(默认灰度差为0.2),则作为候选角点;否则,舍弃该点。

3) 若该点为候选角点,则计算 p_1 到 p_{16} 与中心 p 的像素差,若至少有连续9个值大于阈值,则为角点;否则,舍弃该点。

$$D = (A^T A)^{-1} A^T b, \quad (11)$$

$$\text{式中 } A = \begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix}, d = \begin{bmatrix} u \\ v \end{bmatrix}, b = \begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{tm} \end{bmatrix}.$$

首先通过快速角点检测得到关键点,记为 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$,先利用前向光流计算下一帧中的跟踪点,再进行反向光流找到前一帧中对应的跟踪点,记为 $(x'_1, y'_1), (x'_2, y'_2), \dots, (x'_n, y'_n)$,计算两组跟踪点之间的欧氏距离,保留误差小于中位值的跟踪点作为关键点,记为 $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$,再次通过前向光流计算角点光流偏移值,记为 $(u_1, v_1), (u_2, v_2), \dots, (u_m, v_m)$ 。对所有角点分配权重,第 m 个角点的权值为

$$r_{(x,y)}^m = \exp \left[-\frac{(x_m - x_0)^2 + (y_m - y_0)^2}{2\sigma^2} \right], \quad (12)$$

式中 x_0, y_0 为上一帧目标中心横纵坐标, $\sigma = \sqrt{S}$, S 为目标面积大小。

对所有角点在 x, y 方向上的位移量进行加权平均得到:

$$\begin{cases} U = \frac{\sum_i^m r_{(x,y)}^i u_i}{\sum_i^m r_{(x,y)}^i} \\ V = \frac{\sum_i^m r_{(x,y)}^i v_i}{\sum_i^m r_{(x,y)}^i} \end{cases}, \quad (13)$$

最终得到目标在下一帧的粗略位置 (x, y) ，其中 $x = x_0 + U, y = y_0 + V$ 。

2.4 模板更新方案

传统 KCF 跟踪算法中的更新方法为

$$\begin{cases} \mathbf{a}_t = (1 - \eta)\mathbf{a}_{t-1} + \eta\mathbf{a} \\ \mathbf{g}_t = (1 - \eta)\mathbf{g}_{t-1} + \eta\mathbf{g} \end{cases}, \quad (14)$$

式中 η 为学习因子， \mathbf{a} 和 \mathbf{g} 分别表示滤波器模板以及目标外观模型。 η 在目标受到遮挡时保持不变是不合理的，应该停止对模板的更新，以增强跟踪算法的可靠性。所以在本文中对 η 的取值做以下处理，即：

$$\eta = \begin{cases} 0.02, & R_{\text{PSR}} \geq T_2 \\ 0, & R_{\text{PSR}} < T_2 \end{cases}, \quad (15)$$

式中 R_{PSR} 为当前峰值响应值， T_2 为经验阈值 5.7。

3 实验

3.1 实验方法

为了验证本文算法应对复杂环境的性能，在 OTB^[13] 公开数据集中选取具有代表性的 10 组出现目标遮挡(OCC)的视频序列进行测试，除此还包括包含光照变化(IV)、尺度变化(SV)、快速运动(FM)、出视野(OV)等。对 KCF 算法、检测学习跟踪(TLD)算法^[14]、CSK 算法、内核结构化输出跟踪(Struck)算法^[15]及本文算法进行对比实验。

采用一次通过评估(OPE)、时间稳定性评估(TRE)和空间稳定性评估(SRE)性能指标进行评估，实验结果如图 2 所示。其中成功率和精度分别是指覆盖率大于 0.5 和平均中心位置误差小于 20 pixel 时，算法成功跟踪的帧数与视频总帧数的比值。

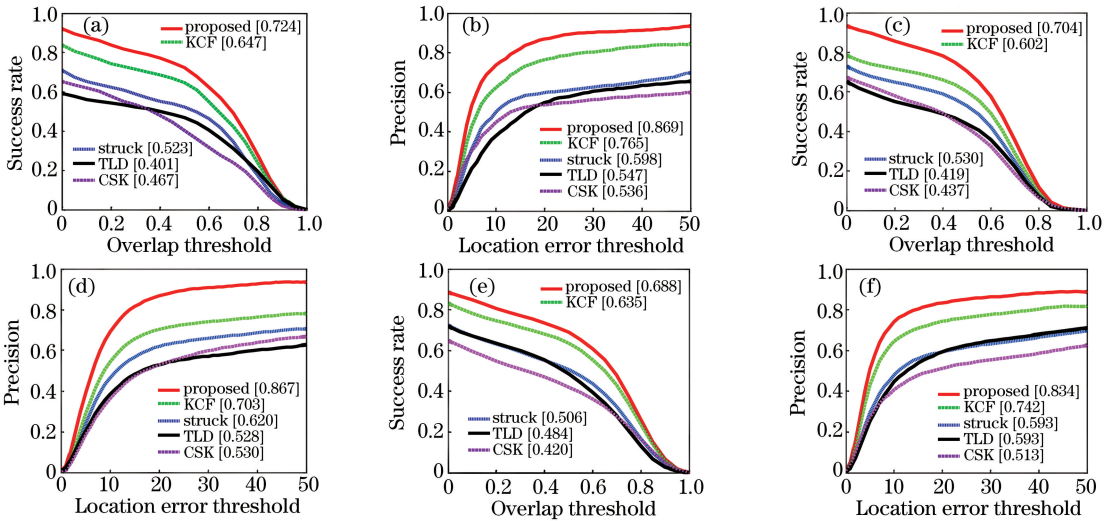


图 2 (a)(b) OPE、(c)(d) SRE 和 (e)(f) TRE 的成功率与精度图

Fig. 2 Success rate and precision of (a)(b) OPE, (c)(d) SRE, and (e)(f) TRE

3.2 实验环境和参数设定

仿真软件为 Matlab2014a，所有仿真均在 Intel i5-3230M CPU，主频为 2.6 GHz，内存为 16 G 的计算机上实现。采用 KCF 算法中的参数，正则化参数 $\lambda = 0.0001$ ，检测区域为目标大小的 2.5 倍等。文中遮挡阈值 $T_1 = 8$ ，严重遮挡阈值 $T_2 = 5.7$ 。

4 结果分析

4.1 定量分析

在 OPE 实验下，本文算法的距离精度为 0.869，比 KCF 提高了 10.4%，比 Struck 提高了 27.1%，比 CSK 提高了 33.3%，比 TLD 提高了 32.2%，本文算法的成功率为 0.724，比 KCF 提高了 7.7%，比 Struck 提高了 20.1%，比 CSK 提高了 25.7%，比

TLD 提高了 32.0%。

在 SRE 实验下，本文算法的距离精度为 0.867，比 KCF 提高了 16.4%，比 Struck 提高了 24.7%，比 CSK 提高了 33.7%，比 TLD 提高了 33.9%，本文算法的成功率为 0.704，比 KCF 提高了 10.2%，比 Struck 提高了 17.4%，比 CSK 提高了 26.7%，比 TLD 提高了 28.5%。

在 TRE 实验下，本文算法的距离精度为 0.834，比 KCF 提高了 9.2%，比 Struck 和 TLD 提高了 24.1%，比 CSK 提高了 32.1%，本文算法的成功率为 0.688，比 KCF 提高了 5.3%，比 Struck 提高了 18.2%，比 CSK 提高了 26.8%，比 TLD 提高了 20.4%。

在各项测试中，本文算法的距离精度和成功率均

有明显的提升,在 TRE 和 SRE 下结果并未出现明显下降,说明本文算法具有较好的抗漂移能力和稳健性。由表 1 可知,本文算法平均速度为 $23.2 \text{ frame}\cdot\text{s}^{-1}$,与

TLD 算法差异不大,比 Struck 算法速度快,而相比 KCF 算法和 CSK 算法速度有明显下降,但仍能达到实时性的标准。

表 1 5 种跟踪算法的平均速度

Table 1 Average speed of five tracking algorithms

Item	Algorithm				
	Proposed	KCF	Struck	TLD	CSK
Average speed / ($\text{frame}\cdot\text{s}^{-1}$)	23.2	110.3	9.56	28.6	217.1

4.2 定性分析

4.2.1 出视野性能分析

图 3 为 5 种算法在 Lemming 和 Suv 上的跟踪效果图。当 Lemming 数据集在第 332 帧目标还未消失时,所有跟踪算法都能够准确地跟踪;在第 380 帧时,目标消失一段时间后再次出现,CSK 和 KCF 算法均丢失跟踪目标;在第 900 帧时目标回归,CSK

和 KCF 算法丢失位置又恢复跟踪。在 Suv 数据集的第 37 帧,目标在视频中消失过半时 Struck 算法结果出现偏移;第 48 帧出现相似目标干扰时 TLD 算法错误跟踪;第 532 帧丛林的出现使 Struck 算法彻底丢失目标。通过实验发现,本文算法能够较好地应对目标出视野情况。

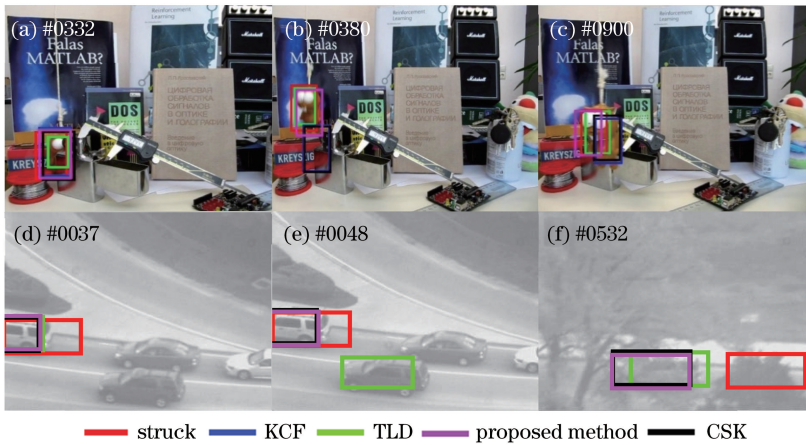


图 3 5 种算法的跟踪效果图。(a)~(c) Lemming;(d)~(f) Suv

Fig. 3 Comparison of tracking results of five algorithms. (a)-(c) Lemming; (d)-(f) Suv

4.2.2 尺度变化性能分析

图 4 为 5 种算法在 Freeman4 和 Woman 上的跟踪效果图。在 Freeman4 数据集中目标从教室后

走向教室前,本文算法和 KCF 算法均能实现对目标的跟踪。在 Woman 数据集中第 124 帧和第 152 帧,目标从出视野到恢复,CSK 和 TLD 算法均跟踪失败;



图 4 5 种算法的跟踪效果图。(a)~(c) Freeman4;(d)~(f) Woman

Fig. 4 Comparison of tracking results of five algorithms. (a)-(c) Freeman4; (d)-(f) Woman

在第 562 帧出现大尺度变换时, TLD 重新恢复对目标的跟踪。通过实验发现, 本文算法能较好地应对小尺度变换, 但在大尺度变换时会丢失跟踪目标。

4.2.3 遮挡性能分析

图 5 为 5 种算法在 Jogging 和 David3 上的跟踪效果图。在 Jogging 数据集中第 70 帧目标还未被

遮挡时, 5 种算法均能实现对目标的准确跟踪; 第 79 帧出现遮挡到 95 帧目标恢复, TLD 算法和本文算法均能重新捕捉到目标位置。在 David3 数据集中第 74 帧背景出现干扰时, TLD 算法丢失跟踪目标, 在第 93~169 帧的过程中, KCF 算法和本文算法标定的目标框均未出现漂移情况。

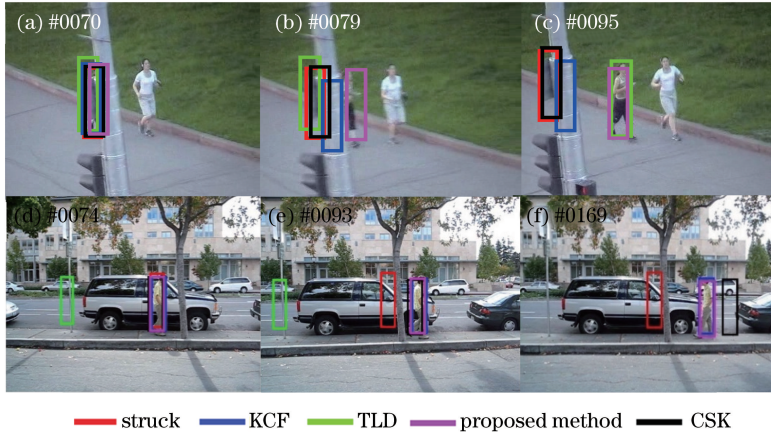


图 5 5 种算法的跟踪效果图。(a)~(c) Jogging; (d)~(f) David3

Fig. 5 Comparison of tracking results of five algorithms. (a)-(c) Jogging; (d)-(f) David3

4.2.4 快速移动性能分析

图 6 为 5 种算法在 Coke 和 Matrix 上的跟踪效果图。在 Coke 数据集中第 128 帧出现短暂遮挡, TLD 算法出现漂移但在第 258 帧恢复; 目标从 258

帧快速移动到 276 帧时, TLD、Struck 算法均能取得较好的跟踪效果。在 Matrix 数据集第 25 帧、39 帧和 96 帧都出现了快速移动, 本文算法均能有效处理, 但结果稍有偏差。

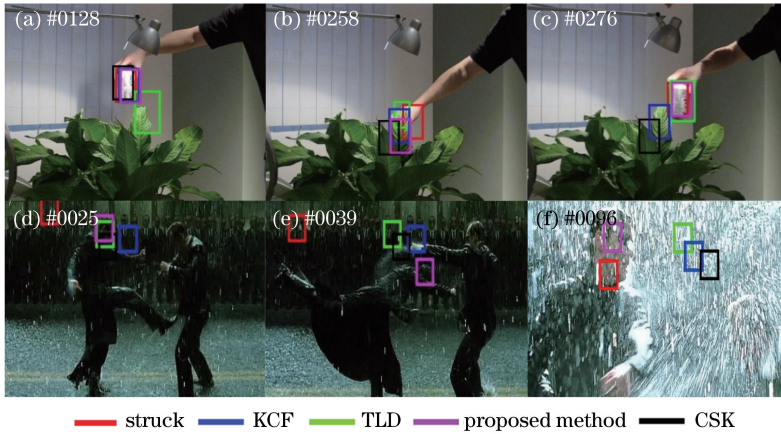


图 6 5 种算法的跟踪效果图。(a)~(c) Coke; (d)~(f) Matrix

Fig. 6 Comparison of tracking results of five algorithms. (a)-(c) Coke; (d)-(f) Matrix

4.2.5 光照变化及旋转变换性能分析

图 7 为 5 种算法在 Soccer 和 Basketball 上的跟踪效果图。在 Soccer 数据集中第 57 帧、69 帧和 116 帧, 目标所受光照不断增强, 只有本文算法能始终保持对目标的精确跟踪。在 Basketball 数据集中第 74 帧时目标快速移动, KCF 算法丢失跟踪目标; 在第 284 帧场景亮度明显提升, 以及第 500 帧目标

发生旋转变换时, 本文算法均能稳健地应对这些情况。

5 结 论

针对传统 KCF 在处理遮挡和光照变化等复杂环境下运行效果很差的问题, 提出利用多特征融合来适应颜色变化, 当目标出现遮挡时, 利用双向光流

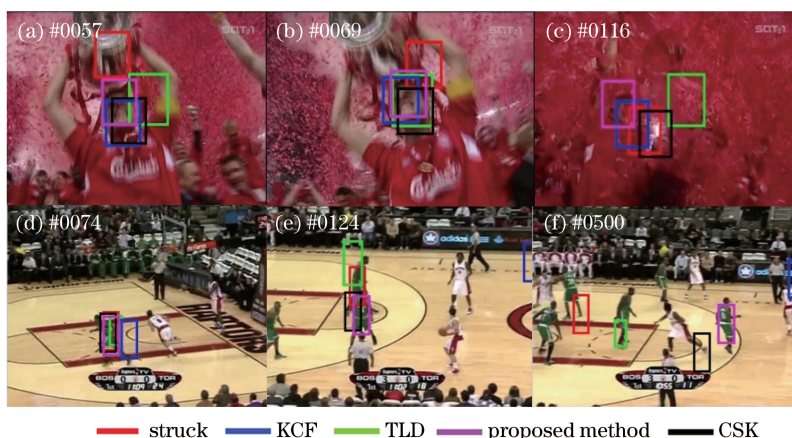


图 7 5 种算法的结果对比图。(a)~(c) Soccer;(d)~(f) Basketball

Fig. 7 Comparison of tracking results of five algorithms. (a)-(c) Soccer; (d)-(f) Basketball

做检测器来估计下一帧位置。通过与已有的先进跟踪算法进行对比实验,发现本文算法速度虽有下降但仍能保持跟踪的实时性,同时验证了本文算法在光照变换、部分遮挡、出视野等复杂环境下均有较强的稳健性。

参 考 文 献

- [1] Kristan M, Matas J, Leonardis A, *et al.* The visual object tracking VOT2015 challenge results [C] // IEEE International Conference on Computer Vision Workshop (ICCVW), 2015: 564-586.
- [2] Pan Z F, Zhu Y L. Kernelized correlation filters object tracking method with multi-scale estimation [J]. Laser & Optoelectronics Progress, 2016, 53(10): 101501.
潘振福, 朱永利. 多尺度估计的核相关滤波器目标跟踪方法[J]. 激光与光电子学进展, 2016, 53(10): 101501.
- [3] 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.
- [4] Zhang W, Kang B S. Recent advances in correlation filter-based object tracking: a review [J]. Journal of Image and Graphics, 2017, 22(8): 1017-1033.
张微, 康宝生. 相关滤波目标跟踪进展综述 [J]. 中国图象图形学报, 2017, 22(8): 1017-1033.
- [5] Kristan M, Pflugfelder R, Leonardis A, *et al.* The visual object tracking VOT2014 challenge results [C] // European Conference on Computer Vision, 2014: 191-127.
- [6] Bolme D S, Beveridge J R, Draper B A, *et al.* Visual

object tracking using adaptive correlation filters [C] // IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010: 2544-2550.

- [7] Henriques J F, Caseiro R, Martins P, *et al.* Exploiting the circulant structure of tracking-by-detection with kernels [C] // European Conference on Computer Vision, 2012: 702-715.
- [8] Danelljan M, Khan F S, Felsberg M, *et al.* Adaptive color attributes for real-time visual tracking [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2014: 1090-1097.
- [9] Zhu M M, Hu M H. Long-term visual object tracking algorithm based on correlation filter [J]. Journal of Computer Applications, 2017, 37(5): 1466-1470.
朱明敏, 胡茂海. 基于相关滤波器的长时视觉目标跟踪方法 [J]. 计算机应用, 2017, 37(5): 1466-1470.
- [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] Wang Y C, Huang H, Li S M, *et al.* Correlation filter tracking based on online detection and scale-adaptation [J]. Acta Optica Sinica, 2018, 38(2): 0215002.
王艳川, 黄海, 李邵梅, 等. 基于在线检测和尺度自适应的相关滤波跟踪 [J]. 光学学报, 2018, 38(2): 0215002.
- [12] van de Weijer J, Schmid C, Verbeek J, *et al.* Learning color names for real-world applications [J]. IEEE Transactions on Image Processing, 2009, 18(7): 1512-1523.
- [13] Wu Y, Lim J, Yang M H. Online object tracking: a benchmark [C] // IEEE Conference on Computer

- Vision and Pattern Recognition, 2013: 2411-2418.
- [14] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409-1422.
- [15] Hare S, Saffari A, Torr P H S. Struck: structured output tracking with kernels [C] // International Conference on Computer Vision, 2011: 263-270.