

## 基于分块的尺度自适应 CSK 跟踪算法

王 暉<sup>1</sup>, 王春平<sup>1</sup>, 付 强<sup>1,2</sup>, 徐 艳<sup>1</sup>, 刘 璞<sup>3</sup>

(1. 军械工程学院 2 系, 石家庄 050003; 2. 清华大学计算机科学与技术系, 北京 100084;

3. 北方电子仪器研究所, 北京 100191)

**摘要:** 基于循环矩阵结构(CSK)的跟踪算法只适用于跟踪尺度固定不变的目标,为此,提出了一种基于分块的尺度自适应 CSK 刚体目标跟踪算法。利用各分块的空间结构推导出分块位置与目标尺度之间的内在关系,然后对各分块的跟踪结果进行置信度评价,选取高置信度的结果进行综合,有效估计目标的尺度和位置,提高了算法在遮挡、背景干扰等情况下的鲁棒性。对典型视频序列的对比试验表明,所提算法不仅能够实时跟踪目标的尺度变化,跟踪的精度和鲁棒性也明显高于原始 CSK 算法。

**关键词:** 目标跟踪; 分块跟踪; 循环矩阵结构; 尺度跟踪

**中图分类号:** TP391 **文献标志码:** A **文章编号:** 1671-637X(2017)02-0025-05

## Patch-Based Scale Adaptive CSK Tracking Method

WANG Wei<sup>1</sup>, WANG Chun-ping<sup>1</sup>, FU Qiang<sup>1,2</sup>, XU Yan<sup>1</sup>, LIU Pu<sup>3</sup>

(1. The 2nd Department, Ordnance Engineering College, Shijiazhuang 050003, China; 2. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China; 3. Northern Electronic Instrument Institute, Beijing 100191, China)

**Abstract:** The object tracking method based on Circular Structure with Kernels (CSK) is only suitable for tracking objects with fixed scale. To solve the problem, a new patch-based scale adaptive CSK tracking method was proposed. First, the relationship between the position of patches and the scale of object was derived from spatial structure of each patch. Then, the tracking confidence level of each patch was evaluated, and those with high confidence levels were selected for integration. Therefore, effective estimation was made to the scale of the object through independent tracking of each patch and the integration. The robustness of the method under interference was improved. Experiment was made by using the typical vision sequences. The result showed that the algorithm can not only realize real-time tracking of the change of object scale, but also has higher accuracy and robustness compared with the original CSK method.

**Key words:** object tracking; patch-based tracking; CSK; scale tracking

### 0 引言

视觉跟踪是计算机视觉的一个基础研究方向,在人机交互、监控、增强现实、机器感知等场景中有着广泛的应用。视觉跟踪也称在线跟踪<sup>[1]</sup>,其任务是给定初始目标状态条件下,在后续帧中对目标状态进行实时估计,跟踪中面临的主要挑战包括光照变化、遮挡、背景干扰、目标形变、缩放等。CSK 算法是 Henriques 等人<sup>[2]</sup>于 2012 年提出的一种高速跟踪算法,在 CVPR2013 的跟

踪算法综合测评<sup>[1]</sup>中,该算法在实时性、抗遮挡和背景干扰方面表现出了非常优异的性能。Danelljan 等人<sup>[3]</sup>通过对样本的全局学习和特征降维技术将色彩属性(Color Name)引入 CSK 框架,大幅提高了 CSK 算法的性能。Henriques 等人在后续的研究中<sup>[4]</sup>,针对 CSK 算法只采用灰度进行跟踪导致效果不佳的问题,利用多通道 HOG 特征进行跟踪,提高了算法对目标和背景的判别能力。但由于 CSK 算法的分类器维数固定,致使其无法进行尺度跟踪,限制了 CSK 及其改进算法的跟踪性能和应用场景。

对目标分块的方法多用于目标识别和检测<sup>[5]</sup>,以及人体跟踪<sup>[6]</sup>等领域。文献[7]用分块的直方图描述目标,通过投票得出跟踪结果,并引入积分直方图方法保证跟踪的实时性,实现了抗遮挡的 MeanShift 跟踪算

收稿日期:2015-12-08

修回日期:2015-12-28

基金项目:国家自然科学基金(61141009)

作者简介:王 暉(1989—),男,甘肃靖远人,博士生,研究方向为计算机视觉、嵌入式系统设计。

法;文献[8]采用人工选取的方式对目标进行分块,使得分块能够有效表达目标形状;文献[9]考虑了分块在时间和空间上的相关性,通过解位置约束下的低秩稀疏问题实现了遮挡条件下的目标跟踪。上述算法大多关注分块方法的抗遮挡性能,忽略了分块的位置(尤其是相对位置)所携带的尺度信息。而事实上,当目标尺度发生变化时,各分块的相对位置也会发生变化。本文从这个基本的观测出发,推导出目标的位置和尺度与目标各分块位置的线性关系,并对分块的跟踪进行评价,将尺度的求解转换为对分块跟踪结果的综合问题。

## 1 CSK 算法

大部分判别式跟踪算法为满足跟踪的实时性要求,对背景和背景都采用稀疏采样策略,从而制约了跟踪性能<sup>[4]</sup>。CSK 算法通过样本的转移实现了一种特殊的稠密采样,并发现样本能通过核运算形成特殊的结构——循环矩阵结构,在此基础上引入经典信号处理中的卷积操作代替一般跟踪算法所采用的滑窗操作,实现了稠密采样条件下的高速跟踪。

训练阶段,CSK 算法从当前帧采样以目标为中心、大小为  $M \times N$  的图像块  $p$ ,对  $p$  在行和列方向上分别做循环移位,得到训练样本集  $P = \{p_{m,n}\}$ ,其中,  $(m,n) \in \{0,1,\dots,M-1\} \times \{0,1,\dots,N-1\}$  分别表示在行和列上做  $m$  次和  $n$  次循环移位。样本  $p_{m,n}$  的训练标记为高斯函数  $r(m,n)$ ,目标处对应的值为 1,距离目标越远值越接近 0。训练的目的在于在样本集  $P$  上学习一个分类器,用于在后续帧中检测目标。

CSK 采用岭回归方法(Ridge Regression,即 Regularized Least Squares,RLS)学习分类器,求使得如下损失函数最小的分类器参数  $\omega$ ,即

$$\min_{\omega} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |\langle \varphi(p_{m,n}), \omega \rangle - r(m,n)|^2 + \lambda \|\omega\| \quad (1)$$

式中,  $\varphi(p_{m,n})$  将  $p_{m,n}$  映射到高维的特征空间  $H$  中,  $H$  通过内积核  $\kappa(f_1, f_2) = \langle \varphi(f_1), \varphi(f_2) \rangle$  定义,该内积核可为点乘核或径向基函数核等;  $\lambda$  为正则项。上述最小化问题的解<sup>[10]</sup>为  $\omega = \sum \alpha_{m,n} \varphi(p_{m,n})$ ,其中,  $\alpha_{m,n}$  具有解析解。

$$\alpha = (K + \lambda I)^{-1} r \quad (2)$$

式中:  $\alpha$  和  $r$  均为  $M \times N$  的矩阵,元素分别为  $\alpha_{m,n}$  和  $r(m,n)$ ;  $K$  为训练样本生成的核矩阵;  $I$  为单位阵。CSK 算法的关键是证明了  $K$  为循环矩阵,进而利用循环矩阵的属性求解分类器参数  $\alpha$ ,即

$$\alpha = F^{-1} \left( \frac{F(r)}{F(k) + \lambda} \right) \quad (3)$$

式中:  $F$  和  $F^{-1}$  分别表示傅里叶变换及其逆变换;  $k$  是元素为  $\kappa(p_{m,n}, p)$  的矩阵。由于矩阵维度降低和采用了快速傅里叶变换(FFT),式(3)较式(2)的运算量大降低。

检测阶段,从新的一帧中采样图像块  $z$ ,利用学习到的目标表现  $p$  和分类器  $\alpha$ ,可得出  $z$  内所有位置处的检测响应

$$\hat{r} = F^{-1} (F(\bar{k}) F(\alpha)) \quad (4)$$

式中,  $\bar{k}$  是元素为  $\kappa(z_{m,n}, z)$  的矩阵。对响应  $\hat{r}$  归一化,则响应矩阵中的元素值表示目标位于该元素处的概率,所以,  $\hat{r}$  的最大值即为当前帧的目标位置。

CSK 算法采用高斯径向基函数作为核函数  $\kappa(\cdot)$ ,并利用 FFT 计算,采用线性插值对表现  $p$  和分类器  $\alpha$  进行更新,每帧跟踪的计算复杂度由  $O((MN)^2)$  降为  $O(MN \ln \sqrt{MN})$ 。具体细节见文献[2,4]。

## 2 本文算法

### 2.1 尺度计算原理

跟踪的任务是确定目标在当前帧  $I^t$  的状态  $(x^t, y^t, s_x^t, s_y^t)$ ,  $(x^t, y^t)$  和  $(s_x^t, s_y^t)$  分别为目标的中心坐标和行列方向上的尺度,  $t$  表示当前时刻。现有的基于分块的跟踪算法,仅利用分块的跟踪结果对目标进行定位,忽视了各分块的跟踪结果所包含的尺度信息。对于刚性目标,当目标尺度发生变化时,各分块的相对位置也会发生变化,所以通过分块间位置的变化可以反推出目标的尺度变化情况。假设目标分为等大小的  $U \times V$  块,只对分块位置进行跟踪,分块  $(u, v)$  的状态为  $(x_{u,v}^t, y_{u,v}^t)$ ,  $u = 1, 2, \dots, U, v = 1, 2, \dots, V$ ,如图 1 所示。

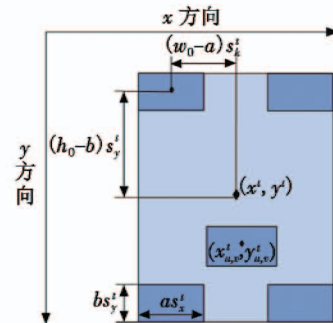


图1 目标分块方式

Fig.1 The schematic of our patches

从图 1 可得出如下结论,即

$$\begin{cases} x_{u,v}^t = x^t - (w_0 - a) s_x^t + (u - 1) (a s_x^t) = x^t + (ua - w_0) s_x^t \\ y_{u,v}^t = y^t - (h_0 - b) s_y^t + (v - 1) (b s_y^t) = y^t + (vb - h_0) s_y^t \end{cases} \quad (5)$$

式中:  $a s_x^t$  和  $b s_y^t$  为分块分别在  $x$  和  $y$  方向上的尺寸;  $(w_0 - a) s_x^t$  和  $(h_0 - b) s_y^t$  为目标中心与左上角分块中心

分别在  $x$  和  $y$  方向上的位置差;  $(w_0 - a)$  是由分块方式确定的一个常数, 在表达上减去  $a$  是为了与后一项的  $as_x^t$  进行抵消, 以简化等式,  $(h_0 - b)$  同理。显然, 这是以  $ua - w_0$  (或  $vb - h_0$ ) 为自变量、 $x_{u,v}^t$  (或  $y_{u,v}^t$ ) 为应变量的直线方程。

以  $x$  方向为例, 完成多个分块的跟踪后, 得到了  $(ua - w_0, x_{u,v}^t)$  平面上  $U \times V$  个点, 将具有相同横坐标的  $V$  个点合并为一个点  $(ua - w_0, x_u^t)$ , 其中,  $x_u^t = (\sum_v x_{u,v}^t) / V$ , 用最小二乘法对得到的  $U$  个点进行直线拟合, 则拟合直线的斜率为  $s_x^t$ , 截距为  $x^t$ 。

$$s_x^t = \frac{U \sum_u (ua - w_0) x_u^t - \sum_u (ua - w_0) \sum_u x_u^t}{U \sum_u (ua - w_0)^2 - (\sum_u (ua - w_0))^2} \quad (6)$$

$$x^t = \frac{\sum_u (ua - w_0)^2 \sum_u x_u^t - \sum_u (ua - w_0) \sum_u (ua - w_0) x_u^t}{U \sum_u (ua - w_0)^2 - (\sum_u (ua - w_0))^2} \quad (7)$$

同理,

$$s_y^t = \frac{V \sum_v (vb - h_0) y_v^t - \sum_v (vb - h_0) \sum_v y_v^t}{V \sum_v (vb - h_0)^2 - (\sum_v (vb - h_0))^2} \quad (8)$$

$$y^t = \frac{\sum_v (vb - h_0)^2 \sum_v y_v^t - \sum_v (vb - h_0) \sum_v (vb - h_0) y_v^t}{V \sum_v (vb - h_0)^2 - (\sum_v (vb - h_0))^2} \quad (9)$$

## 2.2 分块综合

跟踪过程中不可避免地存在遮挡、分块间相互干扰等问题, 导致部分分块跟踪发生漂移或失败, 继而降低了目标的跟踪性能, 所以, 为了实现稳定、鲁棒的跟踪, 需要评估分块跟踪结果的置信度, 以此优化分块的综合方式。

CSK 的检测响应  $\hat{r}$  表示当前帧中目标的分布概率, 理想的检测响应是, 目标处具有最大值, 离目标越远、与目标差异越大的地方响应越小。可用训练标签  $r(m, n)$  作为理想的检测响应, 用分块的检测响应与理想响应的相似度来度量分块的置信度  $C$ , 即

$$C = g(r(m, n), r_{u,v}^t) = \sum_{i=1}^m \sum_{j=1}^n r(m, n)_{i,j} (r_{u,v}^t)_{i,j} \quad (10)$$

式中,  $g(\cdot)$  为相似度函数, 用 Bhattacharyya 系数<sup>[11]</sup>表示。

图 2a 为目标分块及理想响应的示意图, 为表示方便, 对分块编号如图所示。表 1 为由式(10)计算的分块置信度, 其结果与观测是相符的。图 2b 展示了分块 2 的采样区域、检测响应和置信度, 由于分块 2 与背

景及相邻分块具有明显的差异, 所以其响应与理想响应近似, 且具有较高的置信度。然而, 当存在相邻分块干扰或遮挡时, 分块的置信度就会降低。图 2c 中分块 4 和分块 6 的干扰, 使得分块 5 的检测响应出现了较大的噪声。分块 8 存在着严重的遮挡, 无法利用检测响应进行分块定位。

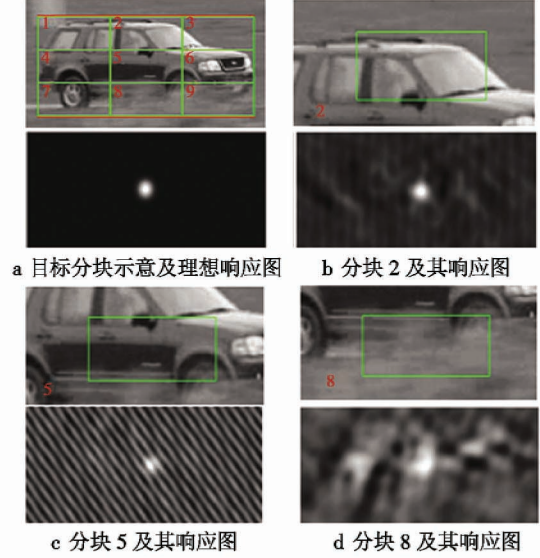


图 2 目标分块及分块响应示意图

Fig. 2 The patches and corresponding response maps

表 1 目标分块置信度结果

Table 1 The confidence coefficient of the patches

分块编号	1	2	3	4	5
置信度	1.000	0.996	0.779	0.735	0.649
分块编号	6	7	8	9	
置信度	0.795	0.753	0.371	0.351	

为提高跟踪性能, 选择置信度最大的  $Q$  个分块计算目标尺度和位置。利用式(6)、式(8)计算目标尺度。位置的计算则采用叠加响应图的方法<sup>[7]</sup>, 相比于式(7)、式(9)只利用了响应的最大值, 该方法相当于用  $Q$  个分块的响应对目标位置综合投票。同时, 叠加响应图的方法能够根据当前帧的目标尺度调整叠加时的位置对应关系, 有效地利用了尺度跟踪结果。

## 2.3 算法流程

综上所述, 本文得出了以 CSK 算法为基础、采用分块策略的尺度自适应跟踪算法。算法第  $t$  帧的输入为当前图像  $I^t$ , 上一帧目标参数  $(x^{t-1}, y^{t-1}, s_x^{t-1}, s_y^{t-1})$ , 分块参数  $U$  和  $V$ , 输出为当前帧目标参数  $(x^t, y^t, s_x^t, s_y^t)$ 。第 1 帧的目标由人工给定, 从第 2 帧开始循环迭代, 每个循环的操作步骤为:

- 1) 根据目标和分块参数, 对目标分块, 对分块尺寸进行调整;
- 2) 用 CSK 算法跟踪各分块;

3) 根据式(10)计算各分块置信度,并排序选出置信度最大的  $Q$  个分块;

4) 将选出的  $Q$  个分块的跟踪结果代入式(6)、式(7),求出目标尺度( $s'_x, s'_y$ );

5) 根据2.2节所述方法叠加置信图,将叠加置信图的最大值对应位置作为目标位置( $x^t, y^t$ );

6)  $t \leftarrow t + 1$ ,返回步骤1)。

### 3 实验及分析

实验平台为 CPU 2.6 GHz Core i5,内存 4 GB 的笔记本电脑,仿真环境为 Matlab2011b。

分块策略为  $U = V = 3$ ,共 9 块,通过实验调参,信任的分块数  $Q = 6$ ,这可能导致在某一帧中  $x$  或  $y$  方向上分块过少,无法进行尺度计算,此时帧的  $x$  或  $y$  方向的尺度不更新。为提高尺度跟踪的稳定性,尺度更新为  $s^t = (1 - \rho)s^{t-1} + \rho s^t$ , $s^t$  为  $t$  时刻目标尺度, $s^t$  为  $t$  时刻本文方法估计的目标尺度, $\rho$  为更新速率。对式(10)中的响应进行归一化和最大值对齐操作,以确保相似性度量的客观、准确。

#### 3.1 评价准则

为了客观地对比所提算法及相关算法的跟踪性能,从跟踪精度和跟踪框重叠率两个方面对跟踪算法进行评价。

跟踪精度的评价采用精度曲线<sup>[1-2,11]</sup>。精度曲线的横坐标是精度误差阈值,纵坐标是单帧跟踪精度高于某个阈值的视频帧数占总帧数的比率。单帧的跟踪精度为跟踪位置与真实位置的像素差,精度曲线反映了跟踪精度在跟踪过程中的整体情况。

跟踪框重叠率定义为  $S = |B_t \cap B_a| / |B_t \cup B_a|$ ,其中, $B_t$  为算法求得的跟踪框, $B_a$  为真实的跟踪框, $\cap$ 和  $\cup$  分别为跟踪框的交和并操作, $| \cdot |$  表示区域内像素数,重叠率的取值范围为 0~1。视频序列的重叠率用成功率曲线评价<sup>[1]</sup>,横轴为重叠率阈值,纵轴为重叠率大于某个阈值的帧数占视频总帧数的比率。

上述两个曲线能够给出跟踪算法对整个视频序列跟踪性能的全面描述,比传统的平均中心误差等准则更具鲁棒性。

#### 3.2 实验结果及分析

选择原始 CSK 跟踪算法<sup>[2]</sup>和自适应尺度跟踪算法 STC<sup>[12]</sup>作为对比算法,在视频序列 mhyang 和 carScale<sup>[1]</sup>上验证本文算法在目标尺度变化和遮挡环境下的跟踪性能。

实验 1 尺度变化。

序列 mhyang 中目标为尺度连续变化的人脸,同时还存在光照变化和目标在平面外的旋转。CSK 算法和

STC 算法都能够基本实现对该序列的跟踪,但由于尺度变化和目标旋转,对其精确跟踪较为困难。图 3 显示了第 265,440 和 1380 帧的跟踪结果,本文算法、CSK 算法、STC 算法的跟踪框颜色分别为红色、绿色和蓝色。



图 3 序列 mhyang 的跟踪结果

Fig. 3 The tracking results of mhyang sequence

当目标变大(440 帧)和侧身旋转(1380 帧)时,CSK 算法定位准确,但尺度固定不变使得跟踪结果变差,STC 算法在跟踪中表现出尺度逐渐变小的特点,在跟踪的后期跟踪目标变为人脸的五官部分。本文算法在上述情况下实现了精确的位置和尺度跟踪,尤其当 1380 帧前后目标出现了平面外旋转(以  $y$  方向为轴)时, $x$  和  $y$  方向具有不同的尺度变化,本文算法依然能够独立跟踪  $x$  和  $y$  方向上的尺度。

图 4 为 3 种算法对序列 mhyang 的跟踪精度曲线和成功率曲线。

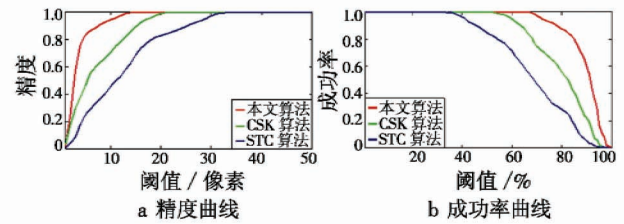


图 4 序列 mhyang 的精度曲线及成功率曲线

Fig. 4 Precision and success plots of mhyang sequence

在精度曲线中,当阈值为 10 时,本文算法、CSK 算法和 STC 算法的取值分别为 0.945,0.705 和 0.445,可以看出,本文算法中占总帧数 94.5% 的图像帧的跟踪结果与目标真实位置的距离小于 10 个像素,远高于其他算法。从成功率曲线可以看出,本文算法的成功率在所有阈值上都高于其他两种算法,3 种算法在重叠率阈值为 70% 时的取值分别为 0.97,0.74 和 0.45。

实验 2 尺度变化+遮挡。

对序列 carScale 进行了跟踪实验,该序列中目标做直线运动,首先保持直线运动,尺度基本保持不变,随后被出现的树枝严重遮挡,同时随着目标靠近摄像机,目标尺度发生较大变化。图 5 和图 6 分别为 3 种算法对序列 carScale 的跟踪结果和精度曲线及成功率曲线。当出现遮挡时(170 帧左右),CSK 和 STC 算法均丢失目标,且无法重新捕获,导致精度曲线和成功率曲线在阈值很低时也无法达到 0.7(170/252(总帧数)=0.67)。由于本文算法基于分块策略,选择高置信度的分块进

行跟踪,有效避免了遮挡引起的目标丢失问题。尺度自适应策略保证了在后续帧目标尺度剧变时本文算法的稳定跟踪。



图 5 序列 carScale 的跟踪结果

Fig. 5 The tracking results of carScale sequence

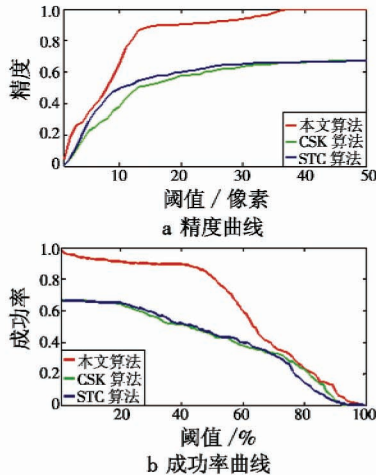


图 6 序列 carScale 的精度曲线及成功率曲线

Fig. 6 Precision and success plots of carScale sequence

本文算法、CSK 算法和 STC 算法的实时性测试结果如表 2 所示,可见 3 种算法的运算速度都远高于 25 帧/s 的实时性要求。表 2 中,序列 carScale 的跟踪速度更快是由于其目标像素数更少,跟踪中所需处理的矩阵维度更小。

表 2 3 种算法实时性分析

Table 2 The real-time analysis of three methods 帧/s

序列	本文算法	CSK 算法	STC 算法
mhyang	151	145	259
carScale	364	371	734

#### 4 结论

视觉跟踪中尺度对跟踪算法的性能有重要的影响,本文利用分块跟踪策略将尺度跟踪引入 CSK 算法。用参数拟合的方法推导出分块跟踪策略下目标尺度和位置的表达。为了提高跟踪的精度和鲁棒性,计算每个分块的置信度,仅利用高置信度的分块跟踪结果计算目标尺度和位置,实验验证了本文提出的算法对遮挡和尺度变化的有效性。但是,对目标分块使得 CSK 算法对目标的潜在搜索范围减小,所以本文算法对高速运动目标或低帧率目标的跟踪效果不是很理想,这将是后续研究的问题。

#### 参考文献

- [1] WU Y, LIM J W, YANG M H. Online object tracking: a benchmark [ C ] // IEEE Conference on Computer Vision and Pattern Recognition, Portland, 2013 : 1354-1362.
- [2] HENRIQUES F, CASEIRO R, MARTINS P, et al. Exploiting the circulant structure of tracking-by-detection with kernels [ C ] // European Conference on Computer Vision, Firenze, 2012 : 702-715.
- [3] 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, Columbus, 2014 : 812-818.
- [4] HENRIQUES F, CASEIRO R, MARTINS P, et al. High-speed tracking with kernelized correlation filters [ J ]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 31 ( 10 ) : 1156-1174.
- [5] AGARWAL S, AWAN A, ROTH D. Learning to detect objects in images via a sparse, part-based representation [ J ]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 20 ( 11 ) : 1475-1490.
- [6] SHU G, DEHGHAN A, OREIFEJ O, et al. Part-based multiple person tracking with partial occlusion handling [ C ] // IEEE Conference on Computer Vision and Pattern Recognition, Oregon, 2012 : 1298-1306.
- [7] ADAM A, RIVLIN E, SHIMSHONI I. Robust fragments-based tracking using the integral histogram [ C ] // IEEE Conference on Computer Vision and Pattern Recognition, 2006 : 798-805.
- [8] NEJHUM S M S, HO J, YANG M H. Visual tracking with histograms and articulating blocks [ C ] // IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 2008 : 546-553.
- [9] ZHANG T Z, JIA K, XU C S, et al. Partial occlusion handling for visual tracking via robust part matching [ C ] // IEEE Conference on Computer Vision and Pattern Recognition, Columbus, Ohio, 2014 : 1552-1559.
- [10] SCHOLKOPF B, SMOLA A J. Learning with kernels: support vector machines, regularization, optimization, and beyond [ M ]. Boston: The MIT Press, 2002 : 89.
- [11] KAILATH T. The divergence and Bhattacharyya distance measures in signal selection [ J ]. IEEE Transactions on Communication, 1967, 15 : 52-60.
- [12] ZHANG K H, ZHANG L, YANG M H, et al. Fast tracking via spatio-temporal context learning [ C ] // European Conference on Computer Vision, Zurich, 2014 : 744-757.