第51卷第5期 南京航空航天大学学报 Vol. 51 No. 5
2019年10月 Journal of Nanjing University of Aeronautics & Astronautics Oct. 2019

DOI:10.16356/j.1005-2615.2019.05.007

面向复杂场景的鲁棒KCF行人跟踪方法

成科扬[1,3] 师文喜[2,3] 周博文[1] 吴金霞[1]

(1.江苏大学计算机科学与通信工程学院,镇江,212013;2.新疆联海创智信息科技有限公司,乌鲁木齐,830001;
3.社会安全风险感知与防控大数据应用国家工程实验室,北京,100041)

摘要: 经典核相关滤波(Kernel correlation filter,KCF)目标跟踪算法是判别式跟踪算法中效果最好的一种跟踪算法。但该算法不能很好地适应目标尺度的变化,且在遇到目标短暂消失或被其他物体遮挡等复杂情形时不具备处理目标重显的能力,因此,为使得目标跟踪能够有效地应对遮挡情形,本文从提高特征表达能力、增加尺度匹配策略和抗遮挡3个方面对经典KCF算法进行改进,提出了一种鲁棒的KCF行人跟踪算法。首先对方向梯度直方图(Histogram of oriented gradients,HOG)特征和色调、饱和度、值(Hue-saturation-value,HSV)特征的响应分布进行特征融合。其次,设置动态选择尺度池来改进滤波器的固定尺寸匹配。最后,通过滤波器响应最大值的变化率衡量目标的遮挡情况,并根据上一成功帧的目标信息,通过EdgeBoxes和感知哈希算法找回目标,更新滤波器。本文所提方法在公开视频跟踪数据集Benchmark上进行测试,实验结果表明与其他目标跟踪方法相比,本文算法提高了尺度变化、遮挡等复杂情形下跟踪的鲁棒性,确保了较高的跟踪精度。

关键词: 目标跟踪;特征融合;EdgeBoxes;感知哈希;重检测;鲁棒性

中图分类号: TP391 **文献标志码:** A **文章编号:** 1005-2615(2019)05-0625-11

Robust KCF Pedestrian Tracking Method with Complex Scene

CHENG Keyang[1,3], *SHI Wenxi*[2,3], *ZHOU Bowen*[1], *WU Jinxia*[1]

(1. School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, 212013, China; 2. Xinjiang Lianhaichuangzhi Information Technology Co., Ltd., Urumqi, 830001, China; 3. National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data, Beijing, 100041, China)

Abstract: The traditional kernel correlation filter (KCF) tracking algorithm is the best one of the discriminant tracking algorithms. However, the KCF algorithm is not robust in case of the target scale change and occlusion. Therefore, we improve the algorithm from three aspects: Improving feature expression ability, increasing scale matching strategy and anti-occlusion. First, we combine the histogram of oriented gradients (HOG) feature and the hue-saturation-value (HSV) feature. Then, a dynamic selection scale pool is employed to replace the way of fixed size matching of filters. Finally, the occlusion of the target is measured by the change rate of maximum response of the filter. And based on the target information, EdgeBoxes and the perceptual hash algorithm are used to retrieve the target and update the filter. Results of the experiments on the benthmark video trocking dataset indicate that comparing with other trackers, the proposed algorithm can effectively improve the tracking accuracy and the robustness even the target with scale change and occlusion.

Key words: target tracking; feature fusion; EdgeBoxes; perceptual hash; re-detection; robustness

基金项目: 社会安全风险感知与防控大数据应用国家工程实验室主任基金资助项目;国家自然科学基金(61972183,61602215)资助项目。

收稿日期: 2019-08-10;**修订日期:** 2019-09-01

通信作者: 成科扬,男,副教授,E-mail: kycheng@ujs.edu.cn。

引用格式: 成科扬,师文喜,周博文,等. 面向复杂场景的鲁棒KCF行人跟踪方法[J]. 南京航空航天大学学报,2019,51(5):625-635. CHENG Keyang, SHI Wenxi, ZHOU Bowen. Robust KCF Pedestrian Tracking Method with Complex Scene[J]. Journal of Nanjing University of Aeronautics & Astronautics,2019,51(5):625-635.

目标跟踪技术是计算机视觉领域的重要研究方向，而行人目标是最为特殊的一种目标。在实际应用场景中，行人目标往往由于背景变化、目标形变、目标遮挡、光照变化以及目标运动过快等影响，造成行人跟踪的效果欠佳甚至失败。因此，设计出一个面向复杂场景的鲁棒行人跟踪算法具有挑战性。目前的跟踪算法主要分为生成式跟踪方法[1-3]和判别式跟踪方法[4-5]。相比较于生成式，判别式跟踪方法速度快，且同时考虑了目标和背景信息，在目标跟踪算法中具有明显优势。相关滤波算法[6]是其中的代表之一，算法把实数域的数据处理过程转化为频率域处理，处理速度大大提升，其在单目标跟踪领域获得成功应用。Henriques等在相关滤波算法的基础上进行改进，提出了核循环结构(Circulant structure kernel，CSK)运动目标跟踪算法[7]，通过循环位移的方式为分类器提供大量良好的训练样本，提高了分类器的性能，但由于使用灰度特征作为目标外观的特征描述子，易受复杂背景、相似颜色物体的干扰。Henriques等随后改进CSK算法，并提出了KCF算法[8]，该算法采用HOG[9]进行特征提取，并构建能够处理多通道的核函数，提升了跟踪的性能。Hamdi等通过学习旋转滤波器，利用HOG特征的循环结构来猜测从一帧到另一帧的旋转并增强对KCF的检测[10]。Galoogahi等使用了HOG特征，通过从背景区域以密集方式采集的负样本，实现了对目标跟踪过程中背景信息的感知[11]。虽然采用单一的HOG特征能够较好地描述目标的轮廓，但是在目标运动模糊或背景受噪声干扰严重时，HOG特征的描述能力会变弱，易导致目标跟踪失败[12-13]，而当背景较为复杂、行人发生非刚性形变时，KCF容易发生跟踪漂移。针对上述问题，本文提出了面向复杂场景的鲁棒KCF行人跟踪算法。

1 KCF算法基本原理

KCF目标跟踪算法是判别式跟踪算法中效果最好的一种跟踪算法。KCF算法通过循环位移的方式为分类器提供大量优秀的训练样本，以训练出性能更好的分类器，并在分类器训练和检测过程中利用快速傅里叶对角化的性质将大量的复杂计算转换为频域中的乘积计算，大大降低了计算量；为了提高跟踪效果，KCF算法使用多通道HOG特征代替灰度特征来提高对目标外观的表达能力。与传统判别式跟踪算法相比，该算法在速度和准确性上都达到了较高的水平。

KCF算法的实质就是求解岭回归问题[8]。在KCF中，从前一帧中选取大小为$M \times N$的图像块作为基样本x_0，通过循环位移得到大量训练样本$\{x_i|i=1,2,\cdots,n\}$，并输入至分类器中完成训练。将样本的训练视为岭回归问题，问题可描述为：假设有n个训练样本$\{x_i|i=1,2,\cdots,n\}$，其对应的回归值为$\{y_i|i=1,2,\cdots,n\}$，因此分类器训练的目的就是找到一个回归函数$f(x_i)=\boldsymbol{w}^{\mathrm{T}}\varphi(x_i)$使得所有训练样本$x_i$和对应的回归值$y_i$的均方误差最小化得到$\boldsymbol{w}$，有

$$\boldsymbol{w}=\min_{\boldsymbol{w}}\sum_{i=1}^{m}(f(x_i)-y_i)^2+\lambda\|\boldsymbol{w}\|^2 \tag{1}$$

式中：λ是正则化参数，映射函数$\boldsymbol{\varphi}(x_i)$为训练样本$x_i$的非线性变换，在求解式(1)的过程中引入了核函数，映射函数$\varphi(x_i)$的核函数为$k(x,x')=\boldsymbol{\varphi}^{\mathrm{T}}(x)\varphi(x')$，在问题求解过程中并不需要知道$\boldsymbol{\varphi}(x_i)$的具体形式，只需要知道核函数和核空间矩阵，矩阵$\boldsymbol{K}$为$n \times n$的矩阵，其元素$K_{ij}=k(x_i,x_j)$，权重向量计算为

$$\boldsymbol{w}=\sum_i \alpha_i\varphi(x_i) \tag{2}$$

式中α为跟踪器模板，训练的目标转换为$f(x_i)=\boldsymbol{w}^{\mathrm{T}}\varphi(x_i)$，将函数和$\boldsymbol{w}$进行替换后，$\boldsymbol{w}$的优化问题就转变为对$\alpha$的求解，利用循环矩阵的性质求解可得到$\hat{\alpha}$，有

$$\hat{\alpha}=\frac{\hat{y}}{\hat{k}^{xx}+\lambda} \tag{3}$$

式中：$\hat{\alpha}$表示α的离散傅里叶变换[14]，$\hat{k}^{xx'}$表示$k(x,x')$中的元素，使用循环稠密采样来循环构造检测样本$\{z_i|i=1,2,\cdots,n\}$，$z_i=P^i z$，定义检测样本的循环矩阵：$\boldsymbol{K}^z=C(k^{xz})$，其中$x$为训练样本的基样本，$z$为检测样本的基样本，KCF跟踪器的检测过程为

$$\hat{f}(z)=\hat{k}^{xz}\odot\hat{\alpha} \tag{4}$$

将$\hat{f}(z)$从频域转化为时域后，数值最大的区域即是跟踪目标的位置。从式(4)可知，在样本检测阶段中也采用了核函数方法，在极大降低算法运算量的同时，也提高了算法的判别能力，常用的核函数有线性、多项式和高斯核等，其中高斯核函数表达式为

$$k^{xx'}=\exp\left(-\frac{1}{\sigma^2}\left(\|x\|^2+\|x'\|^2-2\boldsymbol{F}^{-1}(\hat{x}'\odot\hat{x}^*)\right)\right) \tag{5}$$

式中：σ为高斯核函数参数，$\boldsymbol{F}^{-1}$为逆傅里叶矩阵。

2　鲁棒KCF行人跟踪方法

2.1　算法总体结构

针对传统KCF算法存在的问题，本文提出了面向复杂场景的鲁棒KCF行人跟踪算法模型（如图1所示），该改进算法主要分为3步：

（1）特征融合。将HOG特征和HSV特征的响应分布进行特征融合，以提高特征对目标外观的描述能力。

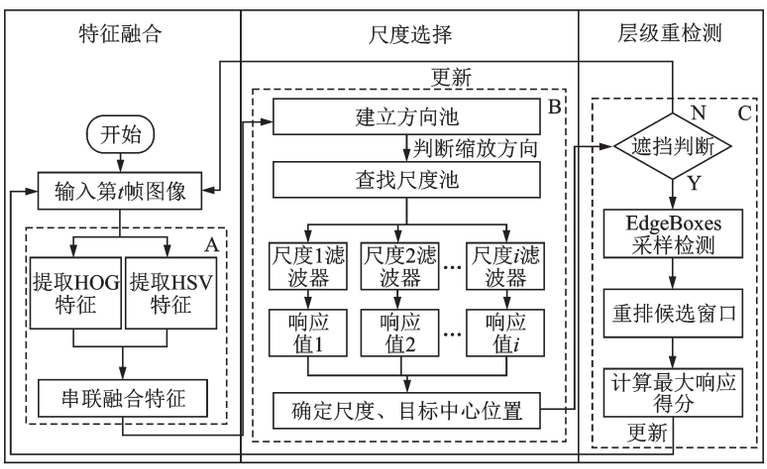


图1　算法总体流程图

Fig. 1　Overall algorithm flow chart

（2）尺度选择。由于KCF算法采用固定大小的跟踪窗口，不能较好地处理目标尺度的变化。因此建立方向池和尺度池，通过方向池的响应值判断目标尺度的缩放方向。然后，根据该方向定向查找尺度池，以匹配最佳尺度。

（3）层级重检测。为了应对因遮挡导致的跟踪漂移或丢失。通过滤波器响应最大值的变化率衡量目标的遮挡情况，并根据上一成功帧的目标信息，通过EdgeBoxes[15]和感知哈希算法[16]找回跟踪目标，更新滤波器。

2.2　多特征融合

KCF算法是一种基于检测的判别式跟踪算法。这类方法将跟踪问题转换为分类问题，因此，如何训练出一个好的分类器就显得至关重要。为此设计通过提升特征的表达能力来提升分类器的性能。KCF跟踪算法中使用了HOG特征作为跟踪目标的描述子，虽然该特征可以很好地描述行人轮廓信息，但是在进行跟踪的过程中丢弃了图像的颜色信息。当遇到背景较为复杂、行人发生非刚性形变时，KCF容易发生跟踪漂移的情况。为了提高算法对复杂环境的抗干扰性和准确性，所提方法通过多特征融合方式对算法进行改进，以提高算法对目标外观的描述能力。

虽然融合越多数量的特征会增强算法的性能和稳定性，但也会增加算法的运算量，导致跟踪速度下降。因此在正常情况下融合特征的数量一般不超过3个。本文将HSV颜色特征与HOG特征进行融合，充分利用各自的优点，有效地将它们结合起来。

HSV颜色空间能够直观的表达色彩的明暗、色调以及饱和程度，受图像尺度变化、拍摄角度等情况的影响较小，是图像检索、图像处理中应用最多的颜色特征。相比于RGB颜色空间，HSV更接近于人眼的视觉特征。故本改进算法中选取HSV颜色与HOG特征进行特征融合，提高算法的特征描述能力。考虑到HOG特征和HSV特征是两个不同类别的图像描述子，如果通过串联的方式实现特征融合，容易忽略不同特征代表的信息量，导致最终的效果有误差。故所提方法采用后融合的方式进行特征融合操作，主要是在响应层上进行融合，采用HSV颜色特征的响应来过滤HOG特征的噪声，辅助滤波器对响应区域的定位，特征融合过程图如图2所示。特征融合的具体步骤如下。

步骤1　同时提取HOG特征和HSV颜色特征，各自通过核化的最小二乘法训练分类器。

步骤2　代入式（4），以G和H表示HOG和HSV特征，则HOG特征的最大响应值r^{G}和HSV颜色特征的最大响应值r^{H}具体计算公式为

$$r^{G}=K^{G}(I)\qquad r^{H}=K^{H}(I)\tag{6}$$

式中：K表示滤波器，I表示当前图像。

步骤3　将HOG特征和HSV特征的响应分布进行融合，得到融合特征响应值r^{GH}，具体计算公式为

$$r^{GH}=r^{G}\,e\,r^{H}\tag{7}$$

式中e表示元素点乘操作。

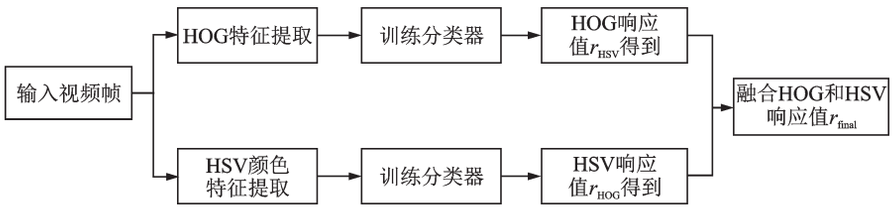


图2 特征融合过程图

Fig. 2 Feature fusion process diagram

2.3 尺度选择

由于传统KCF算法的滤波器输出目标尺寸是固定不变的，因此在跟踪过程中因目标靠近、远离摄像头导致的尺寸变化会严重影响跟踪效果。当目标尺寸缩小时，跟踪框内包含的背景信息会不断增加，分类器会不断地学习背景信息。当目标尺寸增大时，跟踪区域会错误地跟踪目标的局部特征。考虑到这两种情况都会造成跟踪模型漂移和丢失，因此，提出一种基于尺度池的动态选择方法来解决该问题。通过方向池获得目标缩放方向，然后通过该方向定向二分查找尺度池，以匹配最佳尺度，具体内容如下。

设当前帧为第 t 帧，第 $t-1$ 帧的尺度模板为 $s_{t-1}=(s_{t-1,x},s_{t-1,y})$，设置尺度池 Scale$=\{u_1,u_2,\cdots,u_{\lceil i/2\rceil},\cdots,u_{i-1},u_i\}$，其中 $u_i(i=1,2,\cdots,q)$ 表示尺度因子，尺度因子之间的间隔为 d，其中 q 为正整数，尺度池分为正尺、负尺、中尺3个部分，中尺为尺度 $u_{\lceil i/2\rceil}=1$，负尺为尺度 $u_j<1(0<j<u_{\lceil i/2\rceil})$ 的部分，正尺为尺度 $u_k>1(u_{\lceil i/2\rceil}<k<i)$ 的部分。第 t 帧的尺度模板为 $s_t=s_{t-1}u_i$，本文的尺度估计采用动态选择策略进行。从尺度池的正尺、负尺和中尺各挑选出1个尺度因子组成1个方向池 $\text{Scale}_0=\{u_{\lceil i/2\rceil-1},u_{\lceil i/2\rceil},u_{\lceil i/2\rceil+1}\}$，从第 $t-1$ 帧的目标中心区域提取方向池中的尺度所对应的图像块，并使用双线性插值方法将其缩放至 s_{t-1} 的尺寸大小，然后将统一尺寸后的候选图像块输入至式(4)计算响应分值，最佳尺度计算式为

$$G_0=\text{argmax}F^{-1}(\hat{f}(v)) \tag{8}$$

式中：G_0 为最佳尺度的得分，v 为方向池 Scale_0 对应图像块得到的矩阵，$\hat{f}(z)$ 为式(4)得到的值。当尺度缩放方向和目标尺度变化方向一致时，响应值得分增加，因此方向池中得分最高的尺度即是目标尺度变化的方向。根据得到的信息，在尺度池中开始查找，具体过程如下。

(1)若得分最高的尺度为方向池 Scale_o 的 $u_{\lceil i/2\rceil}$，目标不存在尺度变化，则将 $u_{\lceil i/2\rceil}$ 设置为当前帧的尺度。

(2)若得分最高的尺度为方向池 Scale_o 的 $u_{\lceil i/2\rceil-1}$，目标尺度在缩小，设置 $Ori_t=-1$。采用二分查找法在尺度池 Scale 中的负尺部分中寻找最佳尺度因子，查找区间为 $[u_1,u_{\lceil i/2\rceil-1}]$，该区间的中间位置 $m=\left\lfloor\lceil i/2\rceil/2\right\rfloor$。计算中间位置的尺度因子 u_m 对应的响应值为 G_{u_m}，与 G_0 进行比较。若相等，取尺度 $u_{\lceil i/2\rceil-1}$ 作为当前帧的尺度，否则确定新的查找区域，继续二分查找。确定策略如下。

(1)若 $G_{u_m}>G_0$，由目标尺度变化的有序性可知响应值 $G_0\leqslant G_{u_e}\leqslant G_{u_m}$，其中 e 的范围为 $m\leqslant e\leqslant\lceil i/2\rceil-1$，故而新的查找区间为 $[u_1,u_m]$，继续循环执行上述查找操作。

(2)若 $G_{u_m}<G_0$，由目标尺度变化的有序性可知响应值 $G_0\geqslant G_{u_e}$，其中 e 的范围为 $1\leqslant e\leqslant m$，故而新的查找区间为 $(u_m,u_{\lceil i/2\rceil-1}]$，继续循环执行上述查找操作。

(3)若得分最高的尺度为方向池 Scale_o 的 $u_{\lceil i/2\rceil+1}$，目标尺度在增大，设置 $Ori_t=1$。采用二分查找法在尺度池 Scale 中的正尺部分中寻找最佳尺度因子，查找区间为 $[u_{\lceil i/2\rceil+1},u_i]$，确定区间的中间位置 $m=\left\lfloor\left(\lceil i/2\rceil+1+i\right)/2\right\rfloor$，计算该位置尺度因子 u_m，对应的响应值为 G_{u_m}，与 G_0 进行比较。若相等，取尺度 $u_{\lceil i/2\rceil+1}$ 作为当前帧的尺度。否则确定新的查找区域，继续二分查找。确定策略如下。

(1)若 $G_{u_m}>G_0$，由目标尺度变化的有序性可知响应值 $G_0\leqslant G_{u_e}\leqslant G_{u_m}$，其中 e 的范围为 $\lceil i/2\rceil+1\leqslant e\leqslant m$，故而新的查找区间为 $[u_m,u_i]$，继续循环执行上述查找操作。

(2)若 $G_{u_m}<G_0$，由目标尺度变化的有序性可知响应值 $G_0\geqslant G_{u_e}$，其中 e 的范围为 $m\leqslant e\leqslant i$，故而新的查找区间为 $[u_{\lceil i/2\rceil+1},u_m)$，继续循环执行上述查找操作。

利用动态选择尺度池找到最佳尺度后，更新当前帧的尺度即可。

2.4　层级重检测算法

通过一种结合EdgeBoxes和感知哈希的层级重检测方法来解决目标在跟踪过程中遭遇遮挡时造成的目标丢失的问题。该方法主要分为遮挡激活模块和重检测模块。

(1) 遮挡激活模块

通过大量的实验证明，KCF算法在跟踪的过程中发生目标遮挡后，滤波器的响应最大值会下降，跟踪结果偏移程度越大，响应最大值下降越明显。因此，改进算法提出一种判断当前跟踪状态的方法，使用滤波器响应最大值的变化率来衡量跟踪状态。

假定当前帧为第 t 帧，保存第 $t-1,t-2,\cdots,t-h+1,t-h$ 帧的响应最大值，表示为集合 $Q=\{q_i|i=1,2,\cdots,h\}$，通过式(9)可以得到前 h 帧的平均变化量 C_{average}，具体表达式为

$$C_{\text{average}}=\frac{\sum_{i=1}^{h-1}|q_i-q_{i+1}|}{h-1} \tag{9}$$

然后计算出当前帧相对于前 h 帧的响应最大值的变化率 c_i，有

$$c_i=\frac{|q_{h-1}-q_h|}{C_{\text{average}}} \tag{10}$$

设定阈值 T^{m}，当 $c_i\leqslant T^{\text{m}}$ 时，目标跟踪正常，更新模板。当 $c_i>T^{\text{m}}$ 时，目标跟踪发生遮挡导致跟踪结果偏移严重，停止更新模板，激活层级重检测模块来修正跟踪结果。

(2) 重检测模块

由于在视频序列帧中行人目标的移动具有连续性，而且受到于人体生物特性限制，不会在相邻帧中移动较长距离。因此所提算法充分利用跟踪时空信息，根据前一帧确定的跟踪目标的中心位置来设定EdgeBoxes算法的搜索区域 I^{s}，以便得到可信度更高的候选窗口。

本文所提方法认为某一帧的响应最大值变化率 $c_i\leqslant T^{\text{m}}$，就称其为正确帧。假定当前帧为第 t 帧，保存前 k 个正确帧中目标的中心位置，表示为中心位置集合 $D=\{(\text{loc}_i,\text{loc}_i)|i=t-1,t-2,\cdots,t-k+1,t-k\}$，通过式(11)可以得到平均运动量 d_{average}，具体表示为

$$\begin{aligned}d_{i,i+1}(x,y)&=\sqrt{(\text{loc}_{x,i}-\text{loc}_{x,i+1})^2+(\text{loc}_{y,i}-\text{loc}_{y,i+1})^2}\\ d_{\text{average}}&=\frac{\sum_{i=t-k}^{t-2}d_{i,i+1}(x,y)}{k-1}\end{aligned} \tag{11}$$

其中 $d_{i,i+1}(x,y)$ 表示两个正确帧目标的距离计算函数，根据前一正确帧跟踪目标的中心位置和 d_{average} 确定搜索区域，设定搜索区域 I^{s} 如图3所示，具体表示为

$$\begin{aligned}I^{\text{s}}&=\{(\text{wid},\text{hei})|\alpha_1\leqslant\text{wid}\leqslant\alpha_2,\beta_1\leqslant\text{hei}\leqslant\beta_2\}\\ \alpha_1,\alpha_2&=\text{loc}_{x,t-1}\mp d_{\text{average}}\mp\frac{1}{2}\tau s_{x,t-1}\\ \beta_1,\beta_2&=\text{loc}_{y,t-1}\mp d_{\text{average}}\mp\frac{1}{2}\tau s_{y,t-1}\end{aligned} \tag{12}$$

式中：α_1,α_2 是区域 I^{s} 的左右边界横坐标，β_1,β_2 是区域 I^{s} 的上下边界的纵坐标，wid和hei分别表示搜索区域 I^{s} 的宽和高，$(\text{loc}_{x,t-1},\text{loc}_{y,t-1})$ 表示前1帧跟踪目标的中心位置，参数 τ 是用来动态调整搜索区域的尺寸。

相比较于传统检测算法的滑动窗口，使用EdgeBoxes算法会得到更少的完整包含物体的候选窗口。因此，当计算得到重检测的搜索区域 I^{s}

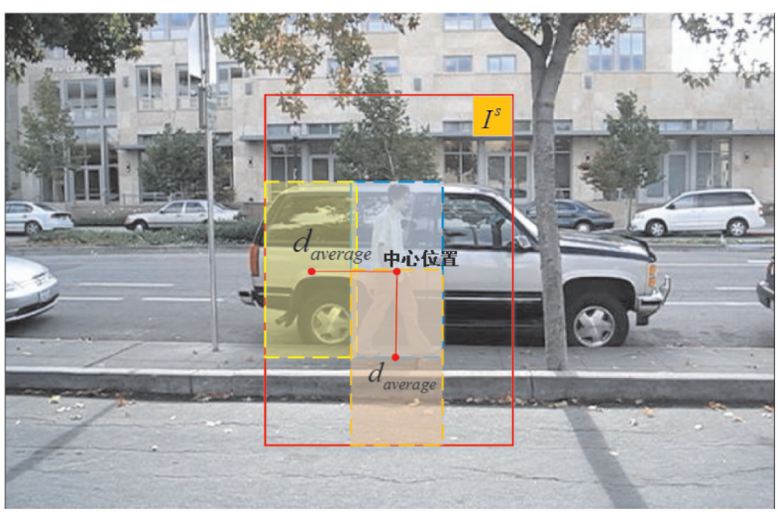


图3　基于跟踪时空信息的目标重检测示意图

Fig. 3　Schematic diagram of target re-detection based on tracking spatio-temporal information

后，采用EdgeBoxes算法对区域I^s进行采样，以得到候选检测窗口组B^b，表示为$B^b=\{b_i^c|i=1,2,\cdots,N\}$，每个候选窗口都需要通过式(13)[23]计算得分

$$h_b=\frac{\sum_i w_b(s_i)g_i-\sum_{i,\bar{x}_i\in b,w_b(s_i)<1}(1-w_b(s_i))g_i}{2(b_w+b_h)^\kappa}$$

$$h_b in=h_b-\frac{\sum_{p\in b^{in}}g_p}{2(b_w+b_h)^\kappa} \tag{13}$$

式中：b_w和b_h为滑动窗口b的宽和高，b^{in}是边界框b的中心位置，b^{in}的宽度和长度分别为$b_w/2$和$b_h/2$，p表示b^{in}的像素，g_p表示边缘强度，κ是一个常数(EdgeBoxes中取$\kappa=1.5$)，用来调节尺寸较大的滑动窗口b含有更多边缘的情况。

通过式(13)得到的对应分值可表示为$S^b=\{s_i{}^b|i=1,2,\cdots,N\}$，其中间可视化结果如图4所示。EdgeBoxes算法检测得到候选检测窗口组B^b是按照分值大小排序的，设置阈值T^z，得分$s_i{}^b$大于阈值T^z的候选窗口才可以被接受。最终的候选窗口组B^b表示为

$$B^b=\{B^b|s_i{}^b>T^z\} \tag{14}$$

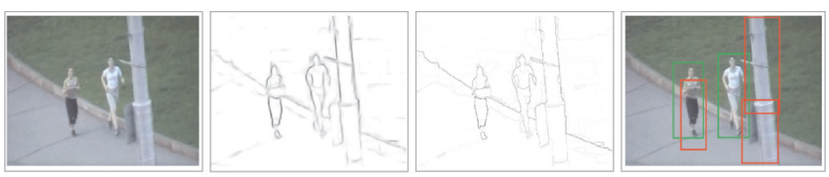


图4 EdgeBoxes似物性采样

Fig. 4 EdgeBoxes physical sampling

相比较目标检测，EdgeBoxes能以更快的速度生成可能包含物体的窗口提议，值得注意的是，由于EdgeBoxes算法的评分机制是评价窗口中是否完整包含物体，所以采取到的候选窗口中，包含跟踪目标的窗口不一定是得分最高，若直接使用候选检测窗口组B^b作为重检测的样本，则很可能会错过包含跟踪目标的窗口。因此，应该考虑到目标跟踪成功时的目标特征信息，而不应该直接使用EdgeBoxes算法得到的候选检测窗口组B^b。

设当前帧为第t帧，设置跟踪目标信息池X^w，以保存前$g-1$个跟踪成功的目标模板，引入感知哈希算法对EdgeBoxes算法采样得到的候选检测窗口组B^b重新打分。具体步骤如下：

(1)第$t-1$帧的目标模板图像为x_{t-1}，保存前$g-1$个成功帧的目标模板池X^w，表示为$X^w=\{x_i|i=t-1,t-2,\cdots,t-g+1,t-g\}$，采用感知哈希函数Hash[16]对目标模板池$X^w$进行求解，得到对应的哈希描述子集合$H^w=\{h_i{}^w|i=t-1,t-2,\cdots,t-g+1,t-g\}$，具体计算公式为

$$H^w=\{X^w|\mathrm{Hash}(X^w)\}_{g-1} \tag{15}$$

具体示意图如图5所示。

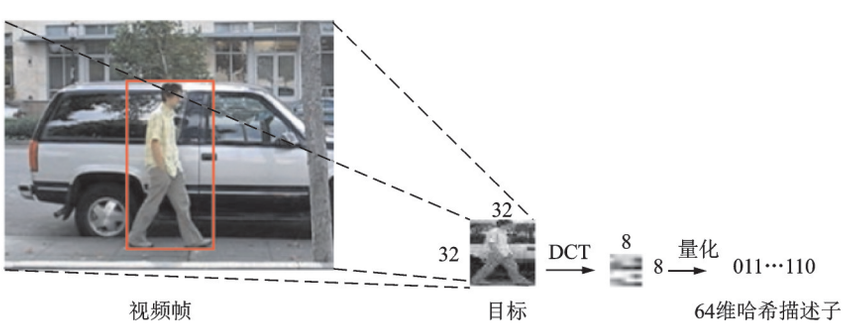


图5 哈希描述子生成算法示意图

Fig. 5 Schematic diagram of the hash descriptor generation algorithm

(2)采用感知哈希函数Hash对候选窗口B^b进行求解，得到对应的哈希描述子集合$H^B=\{h_i{}^B|i=1,2,\cdots,N\}$，具体计算公式为

$$H^B=\{B^b|\mathrm{Hash}(B^b)\}_N \tag{16}$$

(3)感知哈希算法使用汉明距离计算每个描述子之间的相似度，在信息论中，两个等长字符串之间的汉明距离是两个字符串对应位置不同字符的个数，因此汉明距离的数值越小，表示字符串间的相似度越高。相似度计算函数表示为Hamdis。候选窗口的哈希描述子集合H^B中的描述子为h_i^B，目标模板池的哈希描述子集合H^w中的描述子为h_i^w，将集合H^B中的描述子h_i^B依次与集合H^w中的所有

描述子都进行相似度计算，并计算各自平均值 $\mathrm{dis}_i^{\mathrm{average}}$，形成相似度集合 $\mathrm{Dis}=\{\mathrm{dis}_i^{\mathrm{average}}|i=1,2,\cdots,N\}$，具体计算公式为

$$\mathrm{Dis}=\sum_{j=1}^{N}\frac{\sum_{i=t-g}^{t-1}\mathrm{Hamdis}(h_j^B,h_i^w)}{g-1} \tag{17}$$

(4)将相似度集合Dis按分值大小逆向排序，选出排名前M个候选窗口，然后将这些候选窗口输入至KCF跟踪器中，代入到式(4)中进行求解，响应值最大的值即是当前帧最佳结果。由于某一段时间的跟踪过程具有一定的特殊性，保存前l个成功帧的目标响应值，表示为$R^s=\{r_i|i=t-1,t-2,\cdots,t-l+1,t-l\}$，若得到最终响应值小于$R^s$中最小的响应值，则认为当前响应值对应的窗口不包含正确的跟踪目标。那么，5帧之后重新进行一次重检测。以此循环，直至找到正确的目标窗口，并将该结果作为当前帧的目标模板x_t，并更新跟踪器。

3　实验结果与分析

实验的参数都是经过大量测试后得出的最佳取值，尺度池Scale=$\{0.900,\cdots,1,\cdots,1.100\}$的长度为41，间隔$d=0.005$，方向池为$\mathrm{Scale}_o=\{0.995,1,1.005\}$。层级重检测算法中遮挡激活模块的参数$h=6$，响应值变化率阈值$T^{\mathrm{m}}\approx0.1$时效果最好。重检测模块中的参数$k=l=6$，参数$g=7$，参数$\tau$初始取值$\tau=1$，若在重检测模块被激活后的第1帧中修正后的跟踪目标无法被采用，每2帧参数会自增1，直至搜索区域尺寸大于视频尺寸。EdgeBoxes算法筛选候选窗口的得分阈值$TZ=0.005$，感知哈希算法重排候选窗口的数量$M=300$。

3.1　实验环境

本文的实验环境为Intel(R)Pentium(R)CPU，3.6 GHz主频，8 GB内存，开发平台为Pycharm+openCV3.3.0，没有进行任何硬件优化。为了证明本文所提跟踪算法的有效性，选取了当下主流的7种目标跟踪算法在同一实验平台下对14组跟踪行人目标且具有多项挑战性的视频序列上进行综合性能测试，视频序列是从公开视频跟踪数据集Benchmark[17]中挑选而来。该数据集包含了多种挑战性场景，每个测试视频都包括了至少2种挑战场景，以便测试本文算法能否在多种因素干扰的情况下鲁棒的跟踪目标。

3.2　评价指标

为了客观地评估跟踪算法的性能，本文从定性和定量两个方面对算法进行分析。其中参与对比的主流跟踪算法都是使用先前研究工作中得到的实验结果。对于定量分析部分，在对比实验中运用了两种具有代表性的评价指标来衡量跟踪的效果：中心位置误差(Center location error，CLE)和跟踪精度(Detection precision，DP)。

中心位置误差CLE是检测到的目标中心位置与真实中心之间的平均欧几里得距离，平均中心位置误差$\mathrm{CLE}_{\mathrm{average}}$是在视频序列的所有帧中的中心位置误差平均值，其用来评估算法的跟踪性能，其数值越小代表跟踪到的目标位置越精确。具体描述为目标跟踪位置c_{t}与目标真实位置c_{g}之间的欧氏距离D，当D小于给定阈值时，则认为当前跟踪是正确的。其中阈值η通常设置为20(该阈值是经验值)，具体计算公式为

$$\mathrm{CLE}_{\mathrm{average}}=\frac{1}{N}\sum_{i=1}^{N}\sqrt{\big(c_g(x)-c_t(x)\big)^2+\big(c_g(y)-c_t(y)\big)^2} \tag{18}$$

式中$i=1,2,\cdots,N$为视频的帧序号。

跟踪精度DP是在视频序列中成功跟踪的帧数与总帧数的比值，其用于评估跟踪器的精确程度，DP越大则代表跟踪算法精度越高。一种被广泛接受的实践标准是，若某一视频帧中CLE小于阈值η，则当前帧可看作为跟踪成功，具体计算公式为

$$\mathrm{DP}=\frac{\mathrm{count}(\mathrm{CLE}_i<\eta)}{\mathrm{number}} \tag{19}$$

3.2.1　定量分析

表1和表2将本文方法与其他几种经典算法(KCF[8]，CNSGM[18]，Struck[19]，CT[20]，ASLA[21]，CXT[22]和L1APG[23])在14组具有多项挑战性的视频序列上进行对比分析。每个测试视频序列中效果最好的用粗体标注，效果次好的用下划线标注。

从表1可见，本文算法在14组测试集上跟踪结果的平均跟踪精度为73.64%。相比KCF，提高7.78%，相比CNSGM，Struck，CT，ASLA，CXT和L1APG分别提高2%，17.27%，51.14%，20.5%，40.35%和34.71%。从表2可见，本文算法的平均中心位置误差为27.42。相比KCF，降低了18.07%，相比CNSGM，Struck，CT，ASLA，CXT和L1APG分别降低了19.34%，45.23%，91.8%，53.6%，82.17%和81.67%。在DP和CLE准则上的结果显示本文鲁棒的跟踪算法整体最优，更加接近真实结果。

本文结合HSV特征进行多特征融合，设置动态选择尺度池匹配尺度，添加层级重检测模块解决

表1 跟踪精度DP结果对比

Tab. 1 Tracking accuracy DP results comparison %

视频序列	算法							
	Ours	KCF	CNSGM	Struck	CT	ASLA	CXT	L1APG
Basketball	93	92	**99**	12	9	60	4	31
BlurBody	62	58	84	75	2	1	46	**87**
Bolt	**100**	99	46	2	1	2	3	2
Couple	34	26	56	56	31	23	**64**	61
Crossing	**100**	**100**	**100**	**100**	**100**	**100**	56	25
Crowds	99	**100**	39	**100**	1	**100**	**100**	**100**
David3	**100**	**100**	96	99	40	72	16	46
Diving	56	54	**67**	40	5	37	20	20
Girl2	70	8	**82**	27	7	38	11	7
Gym	81	79	**83**	54	27	78	63	2
Human6	34	30	46	27	26	**52**	21	43
Skating2	**40**	38	2	1	6	24	4	3
Walking2	65	44	**100**	**100**	40	63	44	98
Woman	**97**	94	93	96	20	94	14	20
Average	**73.64**	65.86	71.64	56.37	22.5	53.14	33.29	38.93

表2 中心位置误差CLE(像素)结果对比

Tab. 2 Center position error CLE (pixels) result comparison

视频序列	算法							
	Ours	KCF	CNSGM	Struck	CT	ASLA	CXT	L1APG
Basketball	7.68	8.07	**5.58**	118.00	100.58	102.48	171.16	137.81
BlurBody	61.21	64.03	14.20	15.40	87.12	145.89	22.74	**12.05**
Bolt	**6.71**	6.74	79.00	399.00	379.41	390.85	376.59	408.53
Couple	44.37	47.17	24.60	**18.50**	76.43	87.82	40.74	28.72
Crossing	1.50	2.42	1.53	3.70	4.64	**1.37**	30.73	63.29
Crowds	4.25	**3.00**	67.00	4.97	413.86	4.26	4.44	4.61
David3	**3.85**	4.06	7.50	5.85	78.91	54.35	221.97	90.03
Diving	39.40	42.22	**14.70**	39.30	98.97	80.79	66.14	95.76
Girl2	50.56	264.55	**24.00**	139.00	103.47	86.98	135.45	220.53
Gym	14.83	16.45	**12.80**	19.60	25.65	14.83	19.18	79.91
Human6	84.56	107.67	132.00	95.80	**53.24**	79.22	87.48	63.60
Skating2	**29.86**	30.76	259.00	143.00	64.53	45.32	203.63	189.21
Walking2	26.54	29.57	**2.67**	8.79	62.06	29.91	34.00	4.70
Woman	8.66	10.09	10.10	**6.13**	120.16	10.22	120.05	128.51
Average	**27.42**	45.49	46.76	72.65	119.22	81.02	109.59	109.09

在跟踪过程中因为遮挡问题造成的跟踪失败的情况。在一定程度上提高了跟踪的性能,加强了跟踪的稳定性。例如Bolt,Crossing,Women等视频,虽然KCF已经有超过90%的跟踪精度,但本文的算法仍然可以继续提高跟踪的性能。在对Gym,Diving等视频的跟踪过程中,目标会发生尺度变化,而且变化的速度和幅度都比较大,一定程度会增加跟踪难度。但由于在尺度变化时没有过多地损失颜色特征,因此动态选择尺度池和融合特征能够一定程度减少尺度变化带来的影响,提高跟踪算法的稳定性。

在Crossing,Crowds等背景颜色与目标相近的视频序列中,本文的算法依旧可以取得较好的跟踪效果。但是在Crowds视频序列中,相比于传统的KCF算法,成功率出现轻微的降低,中心位置误差也出现轻微的升高。这是因为本文所提方法通过后融合的方式将HSV颜色特征和HOG特征进行融合,保留了各自代表的信息量,其在大多数情况下都能提升跟踪器的性能,但在目标遭受光照变化、背景颜色干扰这类情况时,会发生轻微的漂移误差。但是不会影响跟踪效果导致跟踪失败,总体来说本文算法在背景颜色干扰的情况下仍然取得

了较好的跟踪效果。

对于 Girl2 视频序列，KCF 算法和其他大部分对比算法的初始跟踪结果正常。随后被其他物体长期遮挡之后，未能重新检测到正确的目标。虽然本文算法的成功率和中心位置误差排名第二，但是相比较 KCF，ASLA 等算法，本文算法大幅度提高了成功率，降低了中心位置误差。这是因为所提方法采用重检测机制重新找回跟踪目标，即通过计算判断出目标跟踪失败后，利用 EdgeBoxes 似物性采样算法和感知哈希算法找到丢失的跟踪目标，并更新跟踪器。

从定量分析的结果来看，本文的算法能够在大部分视频序列中取得较好的跟踪结果，并且从表格最后一栏可以看出本文算法的综合性能是最优的，能够在不同场景下较为准确地跟踪目标，说明相比较于其他几个算法更具优势。

3.2.2　定性分析

在图 6—9 中，通过 5 个视频序列的具体跟踪结果来进行本文算法和 KCF 跟踪算法的对比分析，分别包含了常见的 4 种跟踪场景：目标的尺度发生变化、目标被遮挡甚至出现短暂性消失、目标发生形变或姿态的变化、背景杂乱且受到背景的干扰。其中绿色矩形框为 KCF 算法的跟踪结果，红色为我们所提方法的跟踪结果。

(1)目标尺度变化

Women 视频序列的跟踪目标是 1 名行走的女性，该目标经历了遮挡、尺度剧烈变化的情况。从图 6 可以看出，在刚开始的第 103 帧中跟踪环境较为简单，本文算法和 KCF 算法都能正确跟踪目标，随后多次遭遇到路边轿车的部分遮挡，在这种情况下，两种算法都可以比较准确的跟踪目标位置和尺度，在第 585 帧中，视角突然被拉近，目标被迅速放大，尺度变化剧烈，KCF 算法的目标检测框漂移明显，导致跟踪错误。虽然本文算法的目标检测框同样发生了漂移，由于融合特征的鲁棒性较高，并且动态选择尺度池能够处理一定程度的尺度变化，使得本文算法在目标位置上的偏移程度不大，尺度也能较好地适应目标的真实大小，把误差控制在一定范围之内。

测试视频 Women

图 6　对尺度变化显著的目标跟踪结果示例

Fig. 6　Tracking results of a target with significant scale changes

(2)遮挡与目标短暂消失

Girl2 视频序列需要对 1 个女孩进行跟踪，小女孩在运动过程中被另一个行人目标逐渐遮挡，直至被完全遮挡。从图 7 可以看出，在刚开始的第 110 帧中本文算法和 KCF 算法都能正确跟踪目标，KCF 算法在目标被遮挡时易发生漂移，在第 110 帧中女孩被其他行人目标整个遮挡，KCF 算法出现较大的偏移误差，导致跟踪失败，并且在后续视频帧中也保持错误的跟踪结果，我们的算法在经历整体遮挡后，同样发生较大偏移，但是从第 330 帧中可以看出本文的算法在遮挡结束后，自主恢复对于目标的跟踪，这是因为算法在通过计算判断出跟踪失败后，激活重检测机制寻找到丢失的目标并更新跟踪器，从而继续保持正常跟踪，在第 1 390 帧中发生第 2 次严重遮挡，本文的算法在目标重新出现后依旧可以再找到丢失的目标后恢复跟踪。

测试视频 Girl 2

图 7　目标被遮挡或短暂消失的跟踪结果示例

Fig. 7　Tracking results of an occluded or briefly disappeared target

(3)杂乱背景和背景干扰

Basketball 视频序列的跟踪目标是篮球比赛中的 1 位运动员，该场景的背景复杂且干扰严重，在目标周围存在多个与目标相似行人，同时，目标在运动过程中也存在一定非刚性物体形变。从图 8 可以看出，从第 40 帧、第 643 帧和 712 帧中可以看

出目标被周围的运动员严重干扰后，KCF算法的目标检测框发生了偏移，使得检测框内背景部分比例增大。而本文算法在遭受干扰后，目标检测框的位置和尺度更接近于真实的目标情况，这益于本文算法融合了目标的HOG特征和HSV颜色特征，并采用动态选择尺度池估计目标的最佳尺度。

测试视频 Basketball

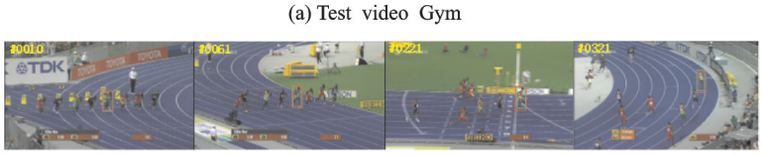
图8 目标在杂乱背景中移动的跟踪结果示例

Fig. 8 Tracking results of a target moving in a messy background

(4)目标形变与姿态改变

Gym和Bolt视频序列的跟踪目标都是运动员，在运动过程中存在频繁的肢体动作，发生了较大的非刚性形变。从图9可以看出采用HOG特征的KCF算法能够在一定程度上适应运动员目标的形变，这是因为HOG特征原本就用于行人检测，但是在序列Gym的第694帧图像中，由于运动员丰富的肢体动作和姿态切换导致算法发生跟踪漂移。由于本文算法融合了目标的HOG特征和HSV颜色特征，HSV颜色特征可以有效利用目标的颜色信息，因此可以在很大程度上降低非刚体形变造成的影响，对目标发生姿态变化具有较好的鲁棒性。

(a) Test video Gym



(b) Test video Bolt

图9 目标发生剧烈形变和姿态变化时的跟踪结果示例

Fig. 9 Tracking results of a target with deformation and pose change

4 结 论

经典的KCF算法不能很好地适应目标尺度的变化，且采用单一特征表示目标。因此，本文提出一种鲁棒的KCF行人跟踪算法。该算法在KCF算法的基础上，就提高特征表达能力、增加尺度匹配策略和抗遮挡3个方面进行改进，具体操作为：先对特征和特征的响应分布进行特征融合。其次通过方向池确目标缩放方向，然后在尺度池中定向查找最佳尺度。最终以滤波器响应最大值的变化率衡量目标的遮挡情况，若判断为遮挡，则通过EdgeBoxes和感知哈希算法筛选出最可能的K个候选目标窗口，响应值最大的即跟踪结果。本文算法具有较高精度，提高了发生尺度变化、遮挡等复杂场景下跟踪的鲁棒性。但是在层级重检测方法中，对EdgeBoxes算法得到的候选窗口进行相似度评价的效果影响着实际跟踪性能的好坏，未来可以通过融合颜色、纹理等特征进行相似度评估的措施增加相似度匹配的精度。

参考文献：

[1] JIA X, LU H, YANG M H. Visual tracking via adaptive structural local sparse appearance model[C]//IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2012: 1822-1829.

[2] ZHANG S, YAO H, ZHOU H, et al. Robust visual tracking based on online learning sparse representation[J]. Neurocomputing, 2013, 100: 31-40.

[3] WANG N, WANG J, YEUNG D Y. Online robust non-negative dictionary learning for visual tracking[C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway: IEEE, 2013: 657-664.

[4] 廖秀峰，侯志强，余旺盛，等. 基于核相关的尺度自适应视觉跟踪[J]. 光学学报，2018，38(7)：0715002.
LIAO Xiufeng, HOU Zhiqiang, YU Wangsheng, et al. A scale adapted tracking algorithm based on kernelized correlation[J]. Acta Optica Sinica, 2018, 38(7): 0715002.

[5] KALAL Z, MIKOLAJCZYK K, MATAS J.

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

[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. Piscataway: IEEE, 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. Berlin, Heidelberg: Springer, 2012: 702-715.

[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 forhuman detection [C]//Computer Vision and Pattern Recognition, IEEE Computer Society Conference on. Piscataway: IEEE, 2005: 886-893.

[10] HAMDI A, GHANEM B. Learning rotation for kernel correlation filter [J]. arXiv: Computer Vision and Pattern Recognition, 2017.

[11] GALOOGAHI H K, FAGG A, LUCEY S, et al. Learning background-aware correlation filters for visual tracking [C]//International Conference on Computer Vision. Piscataway: IEEE, 2017: 1144-1152.

[12] RUAN Y, WEI Z. Extended kernelised correlation filter tracking[J]. Electronics Letters, 2016, 52(10): 823-825.

[13] JIAN L M, JIA X L, QIANG L, et al. Vehicle detection method based on improved HOG-LBP [C]//International Conference on Vehicle, Mechanical and Electrical Engineering (ICVMEE). Lancaster: DEStech, 2016: 3.

[14] GRAY R M. TOEPLITS and circulant matrices: A review (Foundations and Trends (R) in Communications and infomation Theory) [M]. Hanover: Now Publishers Inc, 2006.

[15] ZITNICK C L, DOLLÁR P. Edge boxes: Locating object proposals from edges [C]//European Conference on Computer Vision. Cham: Springer, 2014: 391-405.

[16] NIU X, JIAO Y. An overview of perceptual hashing [J]. Acta Electronica Sinica, 2008, 36(7): 1405-1411.

[17] WU Y, LIM J, YANG M H. Online object tracking: A benchmark [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2013: 2411-2418.

[18] TIAN C, GAO X, WEI W, et al. Visual tracking based on the adaptive color attention tuned sparse generative object model [J]. IEEE Transactions on Image Processing, 2015, 24(12): 5236-5248.

[19] HARE S, SAFFARI A, TORR P H S. Struck: Structured output tracking with kernels [C]//IEEE International Conference on Computer Vision. Barcelona: [s.n.], 2011: 263-270.

[20] ZHANG K, ZHANG L, YANG M H. Real-time compressive tracking [C]//European Conference on computer vision. Berlin, Heidelberg: Springer, 2012: 864-877.

[21] JIA X, LU H, YANG M H. Visual tracking via adaptive structural local sparse appearance model [C]//IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2012: 1822-1829.

[22] DINH T B, VO N, MEDIONI G. Context tracker: Exploring supporters and distracters in unconstrained environments [C]//IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2011: 1177-1184.

[23] BAO C, WU Y, LING H, et al. Real time robust L1 tracker using accelerated proximal gradient approach [C]//IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2012: 1830-1837.

（编辑：刘彦东）