

Relative Similarity Based Online Multiple Instance Learning Algorithm in Object Tracking

Min Chen, Qinghua Zhang, Xiaosen Chen, Jianghu Chen, Siqi Xie, Si Chen

College of Computer and Information Engineering, Xiamen University of Technology, Xiamen Fujian
Email: chensi@xmut.edu.cn

Received: Feb. 3rd, 2019; accepted: Feb. 15th, 2019; published: Feb. 22nd, 2019

Abstract

Object tracking is one of the hot topics in computer vision and has wide applications. At present, many scholars have introduced machine learning methods into target tracking. For example, multiple instance learning (MIL) based object tracking has been proposed. However, the traditional MIL tracking algorithms have some instability under the selection of positive and negative samples, and they are easy to appear the phenomenon of target drifting over time. In order to overcome the above problems, this paper proposes a simple, effective and efficient object tracking algorithm using online multiple instance learning based on relative similarity. The algorithm further selects and weights the samples in the positive bag by defining the relative similarity so as to improve the performance of object tracking. By contrast to the recent algorithms, the experiments show that the algorithm in this paper has a certain increase in accuracy, precision, and the robustness of object tracking.

Keywords

Object Tracking, Multiple Instance Learning, Relative Similarity, Online Learning

基于相关相似度的在线多示例学习目标跟踪算法

陈敏, 张清华, 陈晓森, 陈江湖, 谢思齐, 陈思

厦门理工学院计算机与信息工程学院, 福建 厦门
Email: chensi@xmut.edu.cn

收稿日期: 2019年2月3日; 录用日期: 2019年2月15日; 发布日期: 2019年2月22日

文章引用: 陈敏, 张清华, 陈晓森, 陈江湖, 谢思齐, 陈思. 基于相关相似度的在线多示例学习目标跟踪算法[J]. 计算机科学与应用, 2019, 9(2): 393-405. DOI: [10.12677/csa.2019.92044](https://doi.org/10.12677/csa.2019.92044)

摘要

目标跟踪是计算机视觉领域的研究热点之一,并得到了广泛的应用。目前许多学者将机器学习方法引入目标跟踪,例如,基于多示例学习的目标跟踪算法(即MIL)已经被提出。然而,传统的MIL跟踪算法在正负样本的选择上存在一定的不稳定性,容易在时间的推移下出现目标漂移的现象。为了克服上述问题,提出了一种简单、有效且高效的基于相关相似度的在线多示例学习目标跟踪算法。该算法通过定义相关相似度来对正包中的样本进行进一步的选择与加权,从而提高目标跟踪的性能。与新近算法的实验对比表明,本文提出的算法在目标跟踪的准确性、精度、鲁棒性等方面均有一定的提高。

关键词

目标跟踪,多示例学习,相关相似度,在线学习

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

目前机器学习已经被广泛应用于目标跟踪领域,包括多示例学习(Multiple Instance Learning, MIL)。区别于传统的机器学习方法,MIL方法是利用正负包进行学习,在训练阶段,样本保存在集合中(这里使用“包”的概念),然后为包贴上标签,而不是为每个样本贴标签(标签是用于表示正负样本)。MIL要求正包中至少含有一个正样本,而负包中必须全部是负样本。虽然MIL算法已被成功用于目标跟踪领域[1],但是仍有许多问题有待解决。传统的基于MIL的目标跟踪算法[1]在目标快速移动、遮挡等复杂场景下,跟踪器的目标位置可能不精确,模型在更新时又会用到这些样本作为正负包的选择,随着时间推移,就会导致出现漂移,最终可能导致跟踪丢失。而多数情况下,当选择负包的距离参数设置不准确时,负包中还可能含有部分正样本,这将导致与MIL的基本思想冲突,从而导致学习到的模型更新时遇到歧义,并使得跟踪器的分辨能力下降。

为了克服上述问题,我们发展了一种用于自适应外观模型更新的基于相关相似度的在线多示例学习算法(Relative Similarity Based Online Multiple Instance Learning,即SMILE),使得目标跟踪系统能处理部分或全局遮挡、漂移、快速移动等问题。在SMILE算法中,我们引入了相关相似度(Relative Similarity)的概念,其是一种度量目标模型的相似度度量方式。相关相似度的取值范围是0~1,其取值越大代表图像越有可能蕴含前景目标。在传统的MIL算法基础上,我们通过有效地引入相似度度量机制,进而在一定程度上能够提高目标跟踪的鲁棒性和准确性。

2. 相关工作

目前学者已经提出了大量的目标跟踪算法。根据目标跟踪过程是否更新目标外观模型,目标跟踪方法可以分为离线目标跟踪方法[2][3]和在线目标跟踪方法[4]-[9]。离线目标跟踪方法是使用在跟踪前学习好的目标模型来跟踪视频序列中的目标位置,而在跟踪过程中没有进一步更新模型。该类方法的不足之处在于难以准确跟踪复杂场景下的目标。在线学习的跟踪方法则在跟踪过程中不断更新目标外观模型,从而可以更加鲁棒地应对动态变化的目标,在一定程度上可以避免目标漂移(drifting)问题。本文重点关注

在线目标跟踪方法。

在线目标跟踪算法按照建模方式的不同,可以分为基于产生式学习的在线目标跟踪算法[2] [5] [6] [10] [11] [12] [13]、基于判别式学习的在线目标跟踪算法[1] [7] [8] [9] [14]-[20]。本节根据这两种建模方式来回顾现有的目标跟踪算法。

2.1. 基于产生式学习的在线目标跟踪算法

基于产生式学习的目标跟踪算法通常是通过建立一个目标模型将现实世界中的目标通过计算机视觉中的目标表示方法进行描述,在新的图像帧中,搜索与目标外观模型最相似的区域,而常用的产生式方法有:高斯学习、稀疏表示等方法。特征跟踪(Eigen-Tracking)方法[2]在跟踪过程中使用一种离线子空间模型描述感兴趣的目标。IVT 方法[10]利用增量空间模型适应外观变化。稀疏表示已经被用在了 ℓ_1 跟踪器中,在 ℓ_1 跟踪器中通过目标和干扰物的稀疏线性组合对物体建模[11]。Vojir T [12]等人解决规模适应问题,并提出了一种新的理论估计尺度机制,其只依赖于均值漂移过程的 HELGIGER 距离。Mei 和 Ling [11]将稀疏表示运用到目标跟踪中,其利用稀疏表示的重构误差作为候选目标的权重,将最大权重的候选目标选为跟踪结果。然而,这个跟踪系统的计算复杂度相当高,因此限制了其在实时场景的应用。Li 等人[5]使用正交匹配跟踪算法有效地求解最优化问题来实现对 ℓ_1 跟踪器的进一步扩展。Havangi 等人[6]采用重采样移动算法,在重采样之后进行 MCMC (Markov chain Monte Carlo)移动处理,使得粒子分布与状态概率密度函数更加接近,从而使样本分布更加合理。但所需概率转移次数大,收敛性判断困难。Lytu 等人[13]基于 KL (Kullback Leibler)距离采样的自适应粒子滤波,通过 KL 距离确定后验概率密度与估计值的误差最小时所需粒子数,在尽量减小粒子数目的同时粒子滤波性能也同样能被保证,从而提高了目标跟踪速度。

尽管已经提出了许多在线产生式目标跟踪算法,但仍有数个问题有待解决。第一,必须从连续帧中选取大量的训练样本以便在线学习一个外观模型。因为刚开始只有几个样本,大多数跟踪算法通常假设在这期间目标的外观并没有太大变化。但是,如果目标的外观在开始时有明显的改变,很可能发生漂移问题。第二,当多个样本被获得在当前目标位置,由于外观模型需要适应这些可能错误标记的样本,所以就可能造成漂移。第三,这些生成算法没有使用有助于提高跟踪稳定性和精确性的背景信息。

2.2. 基于判别式学习的在线目标跟踪算法

基于判别式学习的目标跟踪算法采用机器学习中分类的方法来找出与目标最为相似的区域,该方法通常学习一个分类器用于区分目标样本和背景样本。Avidan [3]通过对目标跟踪使用支持向量机分类器来扩展光流法。Collins 等人[14]将一种在线多特征选择机制嵌入在均值漂移跟踪方法中,通过对 RGB 色彩空间中的 3 个分量赋予不同的权重并进行组合来描述目标,从 49 种不同的特征中选择区分度较大的特征进行目标跟踪,该方法在一定程度上提高了目标跟踪的稳定性,但是当目标与背景特征相似时,其跟踪效果较差。Matthias Mueller 等人[15]提出了一种能够增加更多背景信息的框架,允许在 CF 跟踪器 (Correlation Filter Tracking)中整合全局上下文。并重新制定了原始优化问题,为原始和双域的单一和多维特征提供了一个封闭式的解决方案。Grabner 等人[16]提出了一种在线 boosting 算法为跟踪选择特征。然而,当更新分类器时,这些跟踪系统[14] [15] [16]仅使用了一个正样本(也就是,当前跟踪定位到的样本)和几个负样本。因为外观模型更新时会伴随着噪音和可能错误定位的样本,这通常导致跟踪漂移。

近期, Bhat 等人[17]研究了深度特征对目标跟踪性能的影响,分析准确性、鲁棒性的平衡受深度或浅层特征的影响,并提出了一种新的基于跟踪测试结果的质量测量策略。一些半监督学习方法[7] [8] [18] [19]被提出,在该方法里通过一种有结构限制的在线分类器选择正负样本。Grabner 等人[8]提出一种在线半监督 boosting 方法以减轻在第一帧的样本被标记而其它样本未标记状态下的漂移问题。李义翠等人

[19]依据原有的在线半监督 boosting 跟踪算法,通过增加正负样本的约束条件来实时地修正分类器分类的错误,而且结合了目标的先验模型和在线分类器,通过不断迭代使得分类器被更新,进而预测出未被标记样本的类别和相应的权重。Nam 等人[20]提出了一种卷积神经网络(CNN)算法来表示目标外观,多个 CNNs 在其中协作,最终构建成树的结构,并通过沿树路径的平滑更新来保持模型可靠性。刘雨情等人[9]运用超像素分割,获得多量目标前景以及背景的超像素训练数据,利用 $k-d$ 树聚类快速构建出目标前景和目标背景的判别式模型,并联合训练超限学习机,以及结合相关滤波估计出目标尺度,最终完成对目标的准确跟踪。

Babenko 等[1]引入多示例学习到在线目标跟踪里,利用正负包来进行学习。传统的基于 MIL 的目标跟踪算法[1]在选择正负包时常用的做法如下:选择当前跟踪器定位的目标位置的样本加入正包,以及从该位置的很近距离范围内抽取一些样本也作为正包中的样本。并从与目标位置有一定距离的位置抽取大量样本构成负包。该做法存在着一些问题,例如当跟踪定位的目标位置不精确时,会导致出现目标漂移。而多数情况下,当选择负包的距离参数设置不准确时,负包中还可能含有部分正样本,从而导致降低跟踪的准确性和鲁棒性。Zhang 等人[4]提出了一种新的在线加权 MIL 跟踪器(WMIL)。通过样本距离目标位置的远近距离设置相应的权重。而且,算法通过泰勒展开,使用近似计算的方法避免每次选择每个分类器都需做 M 次概率计算的问题,提高了算法计算效率,从而得到一个更加鲁棒和更快的跟踪器。

3. 基于相关相似度的在线多示例学习目标跟踪算法

基于离线学习的多示例学习算法是在训练时需要初始化所有的训练数据。而采用在线的学习方式时,训练数据是逐帧输入的,根据动态训练数据建立分类模型,并能依靠最新的数据整改分类模型,同时更新弱分类器,并能够在一定时间限制内去掉已经没有价值的分类信息。而我们通过加入相关相似度的度量,能够提高弱分类器的准确率,具体表现在对正样本重新进行筛选和加权,提高正包内样本的准确率。算法流程如图 1 所示。

3.1. 在线多示例学习框架

在线多示例学习的主要思想是[1]:结合多示例学习及在线学习来设计框架,从而成功地分类下一帧。在该框架中,更新外观模型是我们算法的重点。我们选取了一组图像块 $X^\gamma = \{x: \|d(x) - d_t^*\| < \gamma\}$, $\gamma < s$ 选取后标记该包为正包,其中 $\|\cdot\|$ 表示的是距离, γ 是距离中心点的半径(以像素为单位), X 表示一个包含图像块的包, x 表示图像块, $d(x)$ 表示图像块的位置, d_t^* 表示 t 时刻目标的位置, s 是跟踪器的搜索半径。对于负包,我们从与之前相同的 γ 与另一个标量 β 形成的环形区域 $X^{\gamma,\beta} = \{x: \gamma < \|d(x) - d_t^*\| < \beta\}$ 选取图像块,其中 β 是另一个距离中心点的半径,如图 2 所示。由于这产生了潜在的样本大集合,我们选取了这些图像块的随机子集,并标记其为负包。

用于训练二元分类器的传统判别式学习算法是利用 $\{(x_1, y_1), \dots, (x_n, y_n)\}$ 作为训练数据集,其中的 x_i 是示例(即图像块计算的特征向量), $y_i \in \{0, 1\}$ 是样本的二进制标签。不同的是,在多示例学习中,训练数据形如 $\{(X_1, y_1), \dots, (X_n, y_n)\}$, 其中包 $X_i = \{x_{i1}, \dots, x_{ij}, \dots, x_{iN_i}\}$, y_i 是包标签, x_{ij} 是第 i 个包中第 j 个示例, N_i 是第 i 个包的示例个数, n 是包的个数。本文中,示例的特征向量采用 haar 特征。包标签被定义为: $y_i = \max(y_{ij})$, 其中 y_{ij} 是第 i 个包中第 j 个示例的标记,它是在训练期间未知的示例标签。换句话说,一个正包至少含一个正示例,但也可能还有负示例。

目前已经提出了许多算法用于解决多示例学习问题[21]。在文献[21]中由 Viola 等人提出的离线学习的 MIL Boost 算法用于目标检测。MIL Boost 算法采用梯度 Boosting 框架[22]来训练集成分类器 MIL,即最大化包的对数似然函数 $L = \sum_i (\log p(y_i | X_i))$ 。

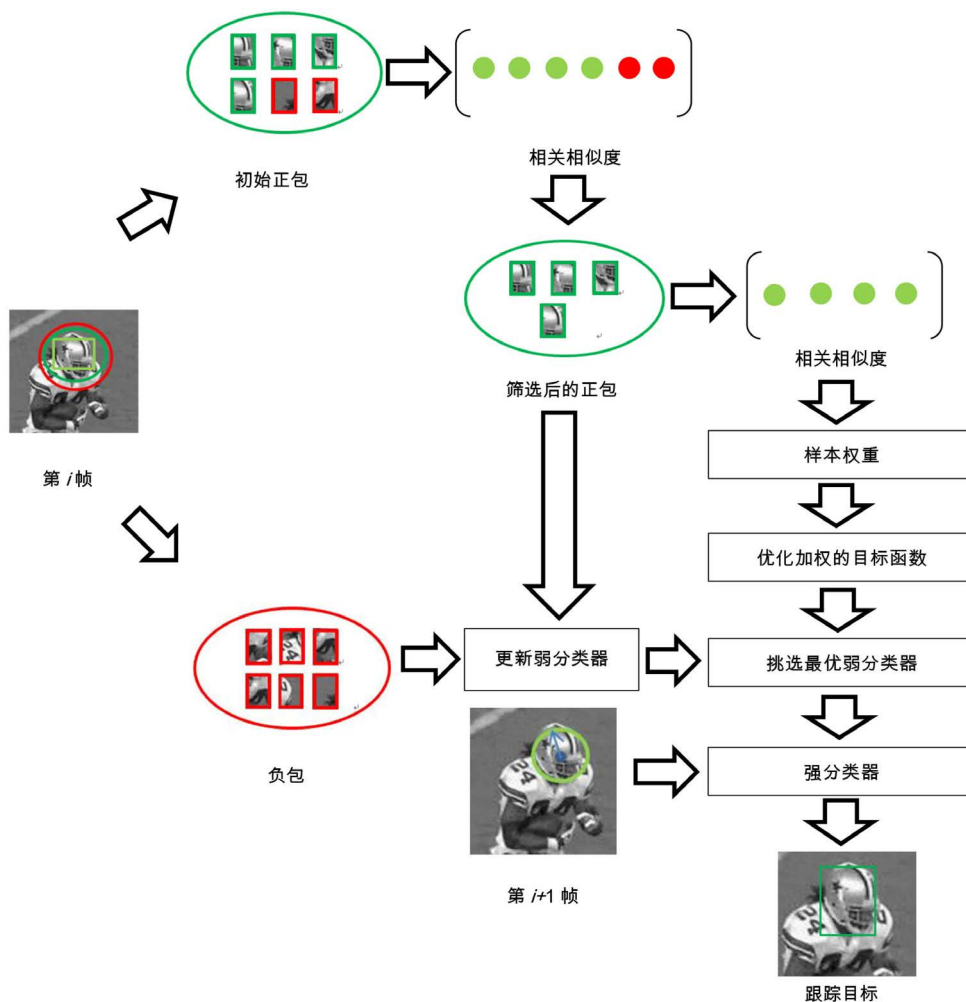


Figure 1. SMILE algorithm flow chart
图 1. SMILE 算法流程

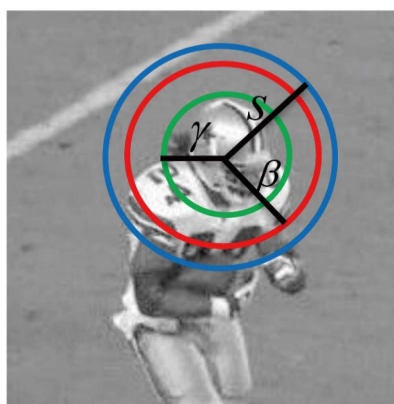


Figure 2. Sample range selection
图 2. 样本范围选择

注意是在包上定义可能性而不是示例上，因为在训练过程中示例标签是未知的。但我们的目标是训练估计 $p(y|x)$ 的示例分类器。因此，我们需要根据其示例概率 $p(y_i|x_{ij})$ 来计算包是正包的概率 $p(y_i|X_i)$ 。

3.2. 相关相似度

传统的在线多示例学习算法[1]在更新弱分类器过程中, 将正包中的样本都视为正样本, 这将导致降低分类器的性能。因为正包中可能还含有负样本。为了提高目标跟踪的鲁棒性和准确性, 本文的 SMILE 算法引入了相关相似度[23]的概念来进一步筛选正包中的样本, 并利用相关相似度对示例概率进行加权, 从而提高跟踪器的性能。

首先, 假设包 X_i 中一个示例用图像块 x_{ij} 表示, 图像块 x_{ij} 是从图像上的目标矩形框抽样得到的, 然后再归一化到 15×15 像素的大小, 归一化的时候不考虑矩形框的长宽比例。两个图像块 x_{ij} 、 x_{ik} 的相似度被定义为:

$$S(x_{ij}, x_{ik}) = 0.5(NCC(x_{ij}, x_{ik}) + 1) \quad (1)$$

其中 NCC 为归一化互相关系数, 用于归一化待匹配目标之间的相关程度, 就是描述所选目标间的相似性。

假设两个图像块 $x_{ij} \in R^{m \times n}$, $x_{ik} \in R^{m \times n}$, 那么 $NCC = \frac{\sum x_{ij} \cdot x_{ik}}{\sqrt{(\sum x_{ij} \cdot x_{ij}) \times (\sum x_{ik} \cdot x_{ik})}}$, 其中 \cdot 表示点乘运算。

对于任意给定一个包中示例 x_{ij} , 我们构建一个目标模型, 它是一些包中示例的集合

$E = \{x_{11}, \dots, x_{1N^+}, x_{01}, \dots, x_{0N^-}\}$, 当前帧为第 j 帧, x_{11} 到 x_{1N^+} 是从第 1 帧到上一帧跟踪到的目标样本, x_{01} 到 x_{0N^-} 是上一帧的负包中的示例。我们定义以下几个量化指标:

① 正样本最近邻相似度, 也叫正最近邻相似度,

$$S^+(x_{ij}, E) = \max_{x_{1j} \in E} S(x_{ij}, x_{1j}) \quad (2)$$

② 负样本最近邻相似度, 也叫负最近邻相似度,

$$S^-(x_{11}, E) = \max_{x_{0j} \in E} S(x_{11}, x_{0j}) \quad (3)$$

③ 相关相似度定义如下:

$$S^r(x_{11}, E) = \frac{S^+}{S^+ + S^-} \quad (4)$$

相关相似度变化范围从 0 到 1, 相关相似度 S^r 的取值越大表示图像块越有可能是目标区域。

相关相似度 S^r 被用于指出一个任意的图像块和目标模型中的部分有多大的相似。给定相关系数 θ , 如果 $S^r(x_{1j}, E) \geq \theta$, 那么图像块 x_{1j} 被筛选进正包, 否则将其从正包中删除。

除此之外, SMILE 算法还通过计算正包中的示例和目标的相关相似度对示例概率进行加权来更准确地估计包的概率 $p(y_i | X_i)$ 。本文算法设 $i=1$ 表示正包, $i=0$ 表示负包。具体公式如下:

$$p(y=1 | X_1) = \sum_{j=1}^{N_1} w_{1j} p(y_1=1 | x_{1j}) \quad (5)$$

不同 WMIL 算法[4], 我们的 SMILE 算法使用相关相似度进行加权, 权重为 $w_{1j} = S^r(x_{1j}, E)$, w 为正包中每个示例的权重。

而对于负包为负的概率, 我们类似文献[4], 具体计算公式如下:

$$p(y=0 | X_0) = \sum_{j=1}^{N_0} w \left(1 - p(y_0=1 | x_{0j}) \right) \quad (6)$$

由于所有负示例都距离目标样本较远，因此这里 w 是一个任意给定的权重值。

类似文献[1]，本文基于 boosting 思想，即将多个弱分类器集成为最终的强分类器，通过不断迭代提高跟踪算法的性能。具体为，通过最大化包对数似然函数 $L = \sum_i (\log p(y_i | X_i))$ 筛选出最优的弱分类器 $h_k^* = \arg \max L(H_{k-1} + h)$ ，其中 h 为弱分类器， H_{k-1} 是前 $k-1$ 次迭代后集成的强分类器。最后将选出的 K 个最优弱分类器集成为强分类器 $H(x_{ij}) = \sum_{k=1}^K h_k(x_{ij})$ ，从而利用该强分类器找到搜索区域中响应值最大的样本作为跟踪目标。

3.3. 本文算法

SMILE 算法的具体算法流程如下：

输入：视频帧号 k ，正负包 $\{X_i, y_i\}_{i=0}^1$ ，其中 $X_i = \{x_{i1}, x_{i2}, \dots, x_{iN_i}\}$ ， $y_i \in \{0, 1\}$ ， $i=1$ 表示正包， $i=0$ 表示负包。

初始化：强分类器 $H(\cdot) = 0$ 。

1：选取一组图像块 $X^s = \{x : \|d(x) - d_i^*\| < s\}$ 并计算特征向量。

2：使用 SMILE 分类器来对 $x \in X^s$ 估计 $p(y|x)$ 。

3：更新当前帧的目标位置 $d_i^* = d(\arg \max p(y|x))$ 。

4：选取两组正包 $X_1 = \{x : \|d(x) - d_i^*\| < \gamma\}$ 和负包 $X_0 = \{x : \gamma < \|d(x) - d_i^*\| < \beta\}$ 。

5：将包 X_i 中每一个示例进行归一化，任意两个图像块 x_{ij} 、 x_{ik} 的相似度被计算为：

$$S(x_{ij}, x_{ik}) = 0.5(NCC(x_{ij}, x_{ik}) + 1)。$$

6：计算正包中每个示例的正样本最近邻相似度， $S^+(x_{1j}, E) = \max_{x_{1k} \in E} S(x_{1j}, x_{1k})$ ，以及负样本最近邻相似度， $S^-(x_{1j}, E) = \max_{x_{0k} \in E} S(x_{1j}, x_{0k})$ 。

7：计算正包中每个示例的相关相似度 $S^r(x_{1j}, E) = \frac{S^+}{S^+ + S^-}$ 。

8：如果 $S^r(x_{1j}, E) \geq \theta$ ，则图像块 x_{1j} 被筛选进新的正包，并计算它的特征向量。

9：利用筛选后的正包、负包中的示例更新弱分类器。

10：计算正包中每个样本的权重 $w_{1j} = S^r(x_{1j}, E)$ 。

11：计算示例概率 $P(y_i | x_{ij}) = \sigma(H(x_{ij})) = \frac{1}{1 + e^{-H(x_{ij})}}$ 。

12：根据公式(5)和公式(6)计算加权的包概率。

13：得到 $L = \sum_i (\log p(y_i | X_i))$ (包的对数似然函数)，通过对它求最大值得到最优的弱分类器，

$$h_k^* = \arg \max L(H_{k-1} + h)。$$

14：重复步骤 11~13，直到选出的 K 个最优的弱分类器依次加入当前强分类器，即

$$H_k(x_{ij}) = H_{k-1}(x_{ij}) + h_k^*。$$

输出：强分类器 $H(x_{ij}) = \sum_{k=1}^K h_k(x_{ij})$ 。

4. 实验

为了评估本文算法的识别能力，我们选用了七个公共视频数据集[24]。这些数据集的详细信息列在了表 1。识别这些数据集的难点在于目标遮挡，运动模糊，光照变化，平面旋转和比例的变化以及快速移动等。

实验中, 我们对比本文的算法和五种新近提出的目标跟踪算法, 包括: CT 跟踪算法[25], CSK 跟踪算法[26], 改进的 MIL 跟踪算法[4], Online Boosting (Online Boosting) [27], Online Semi-supervised Boosting (Semi Boosting) [8]。我们根据论文原文作者所提供的源代码进行实验。为了保证数据的准确性和实验的公平性, 并且减少结果的偶然性, 每个数据集我们都反复测试 10 次, 最后取结果的平均值。我们的算法在 MATLAB 2016a、opencv2.4.4、visualstudio2010 上实现, 我们的计算机配置为 Intel Core i5-6300HQ 2.30 GHz CPU 和 8.0 GB RAM。

Table 1. The information about the seven video sequences

表 1. 七个数据集的相关信息

数据集	帧数	图片大小	目标大小	主要挑战因素
CarScale	250	640 × 272	42 × 26	快速运动, 比例变化, 遮挡, 平面内旋转
Crossing	100	360 × 240	17 × 50	比例变化, 变形, 平面外旋转, 快速运动
David3	250	640 × 480	35 × 131	遮挡, 变形, 平面外旋转
Deer	60	704 × 400	95 × 65	快速运动, 运动模糊, 平面内旋转
Dog1	950	320 × 240	51 × 36	比例变化, 平面外旋转, 平面内旋转
Doll	1500	400 × 300	32 × 73	光照变化, 遮挡, 平面内旋转, 平面外旋转
FaceOcc1	500	352 × 288	144 × 162	遮挡

4.1. 参数设置

在本文的算法中, 我们将学习率参数设置为 0.85。更新时正样本的搜索半径为 4.0, 负样本的搜索半径为 4.0 和 6.0 组成的圆环区域, 采样个数为 50 个。样本框的大小为 25 × 25 像素的矩形。弱分类器池由 150 个弱分类器组成。弱分类器的个数设置为 15。以上参数的设置是参照改进的 MIL 跟踪算法[4]。本文的相关系数 θ 设置为 0.7, 是根据实验结果得到的较好的跟踪效果中选择一个数值, 可以更好地对正包中的示例进行筛选, 对跟踪器的准确性有一定的提升。

4.2. 定量分析

4.2.1. 中心位置误差

我们按照常用的中心位置误差(CLE)的评估指标来测量每一帧的误差, 该公式用于计算每帧跟踪结果和实际目标位置的中心点的误差(即欧式距离)。计算第 i 帧误差的公式如下:

$$\text{CLE}(i) = \frac{1}{T} \sum_{i=1}^T \text{dist}(\text{center}_i(i), \text{gt}(i)) \quad (7)$$

其中, 每帧跟踪结果和实际目标位置的中心位置。由此可以得出, $\text{CLE}(i)$ 的值越小, 算法的误差就越低。**图 3** 列出了六种算法(包括我们的算法)对应的每一帧中心位置的误差。本文的算法在绝大部分的数据集中都取得了较低较平滑的曲线, 说明了我们跟踪器具有较好的精确性和稳定性。在数据集 ‘CarScale’ 中的 150 帧到 250 帧直接由于遮挡和比例变化等问题使得其他跟踪器出现了严重的漂移现象, 曲线出现了急剧上升(参见**图 3** 中的 CarScale)。

但是我们的跟踪器反应较良好, 所以在曲线上变化不大。在数据集 ‘David3’ 中 100 帧到 200 帧直接由于树木的遮挡和目标外观的变形和平面外旋转, 使得大部分跟踪器出现了程度不一的漂移现象。而我们的跟踪器则表现良好, 因此在曲线上表现得低稳平滑。数据集 ‘Deer’ 目标快速运动, 部分跟踪器发生了严重漂移现象, 而我们的跟踪器紧跟目标, 在曲线上表现良好。**表 2** 给出了各个算法对应各个测试集的平均中心位置误差。对于数据集 ‘CarScale’、‘Crossing’、‘David3’、‘Deer’, 本文的跟

跟踪器都取得了最小的 CLE 误差, 在其他的三个数据集上的误差也接近最少。本文提出的算法得到了最小的平均误差, 并且明显优于第二名。这说明本文的算法引入相关相似度来筛选和加权是有效、可行的, 是可以明显提高跟踪性能。

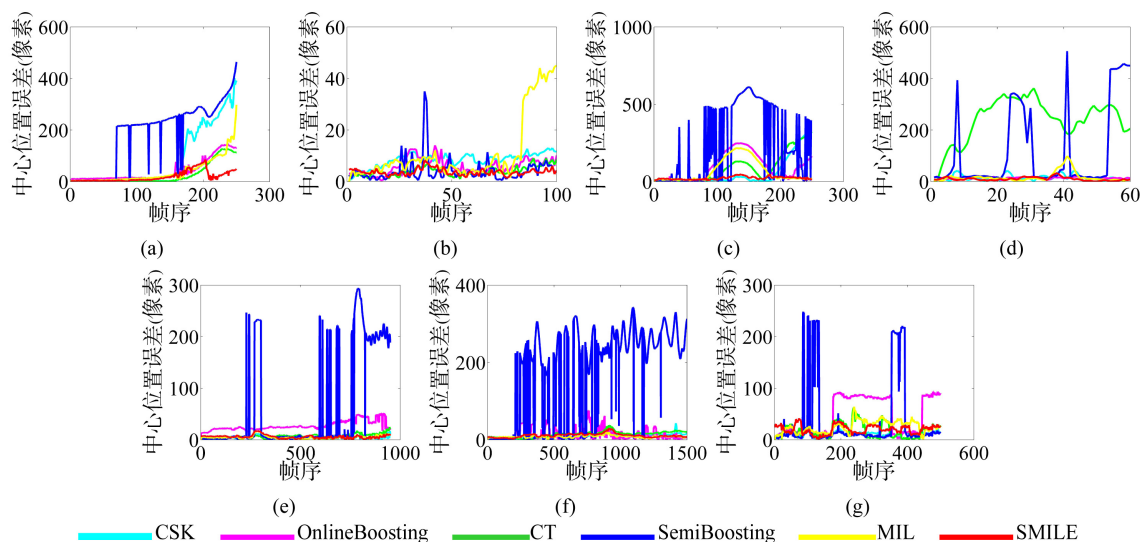


Figure 3. Center location error plots for the six trackers and our algorithm. (a) CarScale dataset; (b) Crossing dataset; (c) David3 dataset; (d) Deer dataset; (e) Dog1 dataset; (f) Doll dataset; (g) FaceOcc1 dataset

图 3. 各算法对应的每一帧中心位置的误差图。(a) CarScale 数据集; (b) Crossing 数据集; (c) David3 数据集; (d) Deer 数据集; (e) Dog1 数据集; (f) Doll 数据集; (g) FaceOcc1 数据集

Table 2. Mean center location errors (in pixels) for the six trackers and our algorithm. Bold fonts indicate the best performance while italic fonts indicate the second best

表 2. 各算法对应测试集的平均中心位置误差, 粗体表示最佳性能, 斜体表示次佳

数据集	CSK	Online Boosting	CT	Semi Boosting	MIL	SMILE
CarScale	80.70	40.97	25.63	183.24	37.74	18.87
Crossing	8.41	6.04	4.08	4.14	11.44	3.72
David3	54.01	91.08	89.25	232.59	67.40	22.30
Deer	15.31	16.50	240.20	117.82	16.92	10.04
Dog1	3.74	27.34	7.27	61.73	4.90	5.87
Doll	9.61	13.47	14.55	179.36	8.22	11.14
FaceOcc1	11.08	46.45	18.13	35.64	24.94	19.95

4.2.2. 跟踪精度

我们还采用精度[1] [28]来评估整体跟踪性能, 将精度定义为跟踪结果与实际目标位置的中心位置误差在某个阈值范围内的帧的百分比, 它的计算公式如下:

$$\text{precision}(\tau) = \frac{1}{T} \sum_{i=1}^T \left[\frac{1}{R} \sum_{r=1}^R \delta(\text{dist}(\text{center}_r(i), \text{gt}(i)) \leq \tau) \right] \quad (8)$$

其中 τ 表示阈值。 $\text{dist}(\cdot, \cdot) \leq \tau$, $\delta(\cdot) = 1$, 否则 $\delta(\cdot) = 0$ 。图 4 可以很直观地看出各个算法在阈值 $\tau \in [0, 100]$ 所对应的精度, 可以看出我们的跟踪器在绝大部分的数据集中都显现的较为笔直。并且随着阈值的变化在大部分数据集中都率先达到 1 或者与第一名相差不小, 说明我们的跟踪器在精度上也表现良好, 在数据集 ‘CarScale’、 ‘David3’、 ‘Deer’ 上表现尤为明显。 ‘CarScale’ 中随着阈值的变化只有我们的

跟踪器能达到精度 1，而其他跟踪器都无法达到精度 1。‘Crossing’和‘Deer’中随着阈值的变化我们的跟踪器对应曲线都是处于第一名或者与第一名相差甚微。‘David3’中也是只有我们的跟踪器达到精度 1 并且在阈值 1~50 之内。

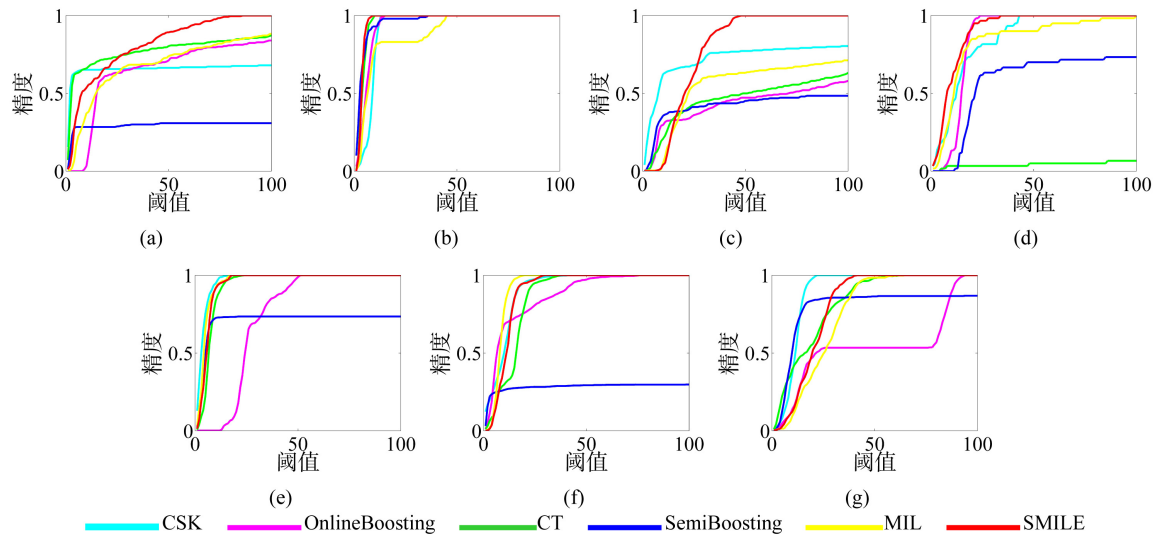


Figure 4. Precision plots for the six trackers and our algorithm. (a) CarScale dataset; (b) Crossing dataset; (c) David3 dataset; (d) Deer dataset; (e) Dog1 dataset; (f) Doll dataset; (g) FaceOcc1 dataset

图 4. 各算法随阈值变化的精度图。(a) CarScale 数据集; (b) Crossing 数据集; (c) David3 数据集; (d) Deer 数据集; (e) Dog1 数据集; (f) Doll 数据集; (g) FaceOcc1 数据集

4.3. 定性分析

在 CarScale 数据集中包括快速运动、比例变化和遮挡三种挑战因素(测试结果参见图 5(a))。在那些不出现遮挡以及小尺度变化的情况下,各算法的跟踪效果都表现良好(参见图 5(a)中的#55 帧)。然而在目标尺度发生较小变化之后, Semi Boosting 算法丢失目标(参见图 5(a)中的#133 帧)。紧接着,在遇到目标遮挡时, MIL 和 Semi Boosting 算法的表现不佳,而本文提出的 SMILE 算法在目标遮挡后依旧有较好的表现(参见图 5(a)中#163 的帧跟踪结果)。CSK、MIL 和 SemiBoosting 算法在经过尺度发生较大变化之后都出现严重漂移问题,而 Boosting 算法跟踪到的目标偏离目标的中心较远;相反,本文的 SMILE 算法在快速运动的数据集中表现得更加稳定和健壮(参见图 5(a)中的#249 帧)。Crossing 数据集不仅包括快速运动还有比例变化(测试结果参见图 5(b))。MIL 算法从#86 帧起逐渐远离目标。CT 算法在该数据集的最后也出现较为严重的漂移(参见图 5(b)#120 帧)。Online Boosting、Semi Boosting 和 CSK 算法可以跟踪到目标,但不是最准确的。本文的 SMILE 算法对比其他算法在比例变化上有较为出色的表现。在 David3 中,当目标被一颗大树遮挡时, Online-Boosting 和改进的 MIL 算法开始漂移(参见图 5(c)中的#86 帧), Semi Boosting 算法甚至出现跟踪失败的结果(参见图 5(c)中的#175 帧和#252 帧)。CSK 和 CT 算法虽然能应对遮挡的情况,但是当数据集加入平面外旋转的挑战因素后,两个算法都受到严重影响(参见图 5(c)中的#252 帧)。综上所述,本文的 SMILE 算法在该数据集中的表现比较稳定。由于快速移动以及运动模糊使得 Deer 数据集十分具有挑战性, CT 算法均出现了严重漂移。CSK、Semi Boosting、改进的 MIL 算法受快速运动的影响甚小,而当目标出现运动模糊时, CSK、Semi Boosting 和改进的 MIL 算法均受到不同程度的影响,其中改进的 MIL 算法出现较为严重的漂移。相反本文的 SMILE 算法跟踪效果非常稳定(参见图 5(d)中的#41 帧)。

在 FaceOcc1 中, 由于杂志书的部分或者完全遮挡导致目标跟踪容易发生漂移。CSK、Semi Boosting 和 Online Boosting 算法都非常容易受到杂志书的影响而出现漂移现象。CT、改进 MIL 算法和本文的 SMILE 算法适用于存在遮挡物的场景下的跟踪。

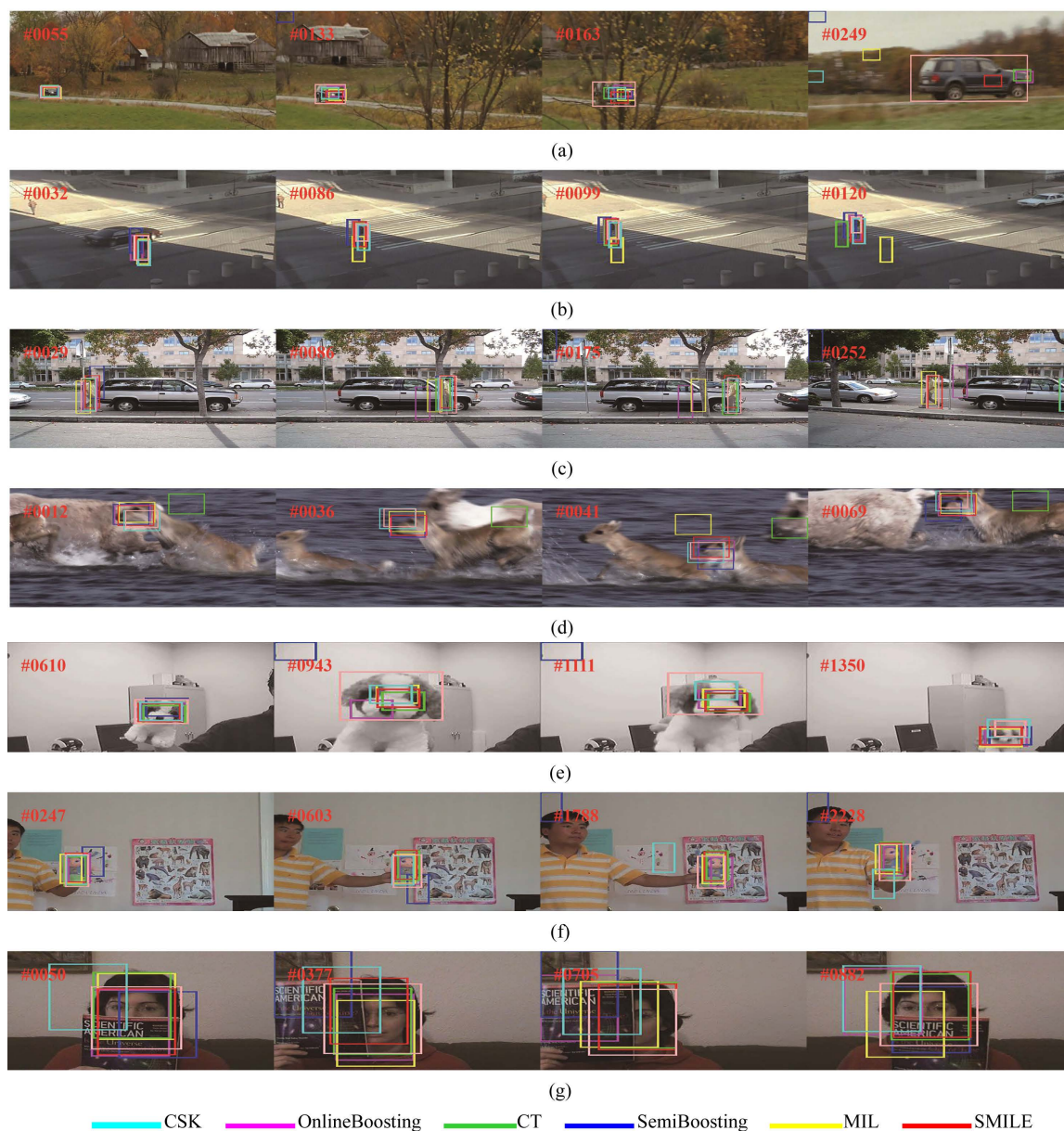


Figure 5. Sample tracking results on seven challenging sequences obtained by CSK, Online Boosting, CT, Semi Boosting, MIL and our proposed SMILE algorithms. (a) CarScale dataset; (b) Crossing dataset; (c) David3 dataset; (d) Deer dataset; (e) Dog1 dataset; (f) Doll dataset; (g) FaceOcc1 dataset

图 5. CSK、Online Boosting、CT、Semi Boosting、MIL 和我们提出的 SMILE 算法在七个具有挑战性的数据集上的目标跟踪结果。(a) CarScale 数据集; (b) Crossing 数据集; (c) David3 数据集; (d) Deer 数据集; (e) Dog1 数据集; (f) Doll 数据集; (g) FaceOcc1 数据集

5. 结论

本文提出了一种新的基于相关相似度的在线多示例学习目标跟踪算法 SMILE。该算法采用了相关相

似度的度量指标对传统的在线 MIL 目标跟踪算法进行改进, 通过相关相似度对正包中的示例进行筛选, 使得用于更新分类器的正包中的示例更加准确; 同时, SMILE 算法使用相关相似度来计算样本的权重, 从而对示例概率进行加权, 使得分类器的性能得以提升, 从而目标跟踪的鲁棒性也得到了提高。通过实验可以发现, 改进后的 SMILE 算法具有更好的目标跟踪性能, 在一定程度上提高了目标跟踪的准确性和精度。

基金项目

本文由国家自然科学基金(61503315), 福建省自然科学基金(2018J01576), 国家级大学生创新创业训练计划(40018025)资助。

参考文献

- [1] Babenko, B., Yang, M.H. and Belongie, S. (2011) Robust Object Tracking with Online Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**, 1619-1632. <https://doi.org/10.1109/TPAMI.2010.226>
- [2] Black, M.J. and Jepson, A.D. (1998) Eigen Tracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *IJCV*, **26**, 63-84.
- [3] Avidan, S. (2004) Support Vector Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1064-1072. <https://doi.org/10.1109/TPAMI.2004.53>
- [4] Zhang, K. and Song, H. (2013) Real-Time Visual Tracking via Online Weighted Multiple Instance Learning. *Pattern Recognition*, **46**, 397-411. <https://doi.org/10.1016/j.patcog.2012.07.013>
- [5] Li, H.X., Shen, C.N. and Shi, Q. (2011) Real-Time Visual Tracking Using Compressive Sensing. *Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, 20-25 June 2011, 1305-1312.
- [6] Havangi, R. (2017) Target Tracking Based on Improved Unscented Particle Filter with Markov Chain Monte Carlo. *IETE Journal of Research*, **64**, 873-885.
- [7] Bertinetto, L., Valmadre, J., Golodetz, S., et al. (2016) Staple: Complementary Learners for Real-Time Tracking. *Computer Vision & Pattern Recognition*, Las Vegas, 26 June-1 July 2016, 1401-1409.
- [8] Grabner, H., Leistner, C. and Bischof, H. (2008) Semi-Supervised On-Line Boosting for Robust Tracking. *Computer Vision*, Marseille, 12-18 October 2008, 234-237. https://doi.org/10.1007/978-3-540-88682-2_19
- [9] 刘雨情, 肖嵩, 李磊. 在线判别式超像素跟踪算法[J]. 西安电子科技大学学报, 2018, 45(3): 13-17.
- [10] Ross, D.A., Lim, J., Lin, R.S., et al. (2008) Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*, **77**, 125-141. <https://doi.org/10.1007/s11263-007-0075-7>
- [11] Mei, X. and Ling, H. (2011) Robust Visual Tracking and Vehicle Classification via Sparse Representation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **33**, 2259-2272. <https://doi.org/10.1109/TPAMI.2011.66>
- [12] Vojir, T., Noskova, J. and Matas, J. (2014) Robust Scale-Adaptive Mean-Shift for Tracking. *Pattern Recognition Letters*, **49**, 250-258. <https://doi.org/10.1016/j.patrec.2014.03.025>
- [13] Lytu, N., Letien, T. and Mai, L. (2017) A Study on Particle Filter Based on KLD-Resampling for Wireless Patient Tracking. *Industrial Engineering & Management Systems*, **16**, 92-102. <https://doi.org/10.7232/iems.2017.16.1.092>
- [14] Collins, R.T., Liu, Y. and Leordeanu, M. (2005) Online Selection of Discriminative Tracking Features. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **27**, 1631-1643. <https://doi.org/10.1109/TPAMI.2005.205>
- [15] Mueller, M., Smith, N. and Ghanem, B. (2017) Context-Aware Correlation Filter Tracking. *IEEE Conference on Computer Vision & Pattern Recognition*, 1387-1395.
- [16] Grabner, H., Grabner, M. and Bischof, H. (2006) Real-Time Tracking via Online Boosting. *Proceedings of the British Machine Vision Conference*, BMVA, Edinburgh, 47-56.
- [17] Bhat, G., Johnander, J., Danelljan, M., et al. (2018) Unveiling the Power of Deep Tracking. *European Conference on Computer Vision*, Springer, Cham, 493-509.
- [18] Cui, Z., Xiao, S., Feng, J., et al. (2016) Recurrently Target-Attending Tracking. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 1449-1458.
- [19] 李义翠, 元琳, 谭舒昆. 结合 PN 约束在线半监督 boosting 目标跟踪算法[J]. 计算机工程与应用, 2017, 53(23): 129-134.

- [20] Nam, H., Baek, M. and Han, B. (2016) Modeling and Propagating CNNs in a Tree Structure for Visual Tracking. *European Conference on Computer Vision*, Amsterdam, 8-16 October 2016, 1-10.
- [21] Viola, P., Platt, J.C. and Zhang, C. (2005) Multiple Instance Boosting for Object Detection. In: *International Conference on Neural Information Processing Systems*, MIT Press, Cambridge, 1417-1424.
- [22] Liu, X. and Yu, T. (2015) Gradient Feature Selection for Online Boosting. *11th International Conference on Computer Vision*, Rio de Janeiro, 14-21 October 2007, 1-8. https://doi.org/10.1007/978-3-662-48570-5_1
- [23] Kalal, Z., Mikolajczyk, K. and Matas, J. (2012) Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **34**, 1409-1422. <https://doi.org/10.1109/TPAMI.2011.239>
- [24] Wu, Y., Lim, J. and Yang, M.H. (2015) Object Tracking Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, 1834-1848. <https://doi.org/10.1109/TPAMI.2014.2388226>
- [25] Zhang, K., Zhang, L. and Yang, M.H. (2012) Real-Time Compressive Tracking. In: *European Conference on Computer Vision*, Springer-Verlag, Berlin, 864-877.
- [26] Henriques, J.F., Caseiro, R., *et al.* (2012) Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In: *Computer Vision*, Springer, Berlin, Heidelberg, 702-715.
- [27] Grabner, H. and Bischof, H. (2006) On-Line Boosting and Vision. *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, New York, 17-22 June 2006, Vol. 1, 260-267.
- [28] Wu, Y., Lim, J. and Yang, M.H. (2013) Online Object Tracking: A Benchmark. *IEEE Conference on Computer Vision and Pattern Recognition*, Portland, 23-28 June 2013, Vol. 9, 2411-2418.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org