doi:10.11959/j.issn.1000-436x.2016181

融合遮挡感知的在线 Boosting 跟踪算法

王亚文，陈鸿昶，李邵梅，高超
（国家数字交换系统工程技术研究中心，河南 郑州 450002）

摘 要：提出融合遮挡感知的在线 Boosting 跟踪算法，该算法对跟踪结果实时进行遮挡检测，根据检测结果自适应调整分类器更新策略。该方式能够有效维护分类器特征池的纯净，提高算法在遮挡环境下的顽健性。实验结果表明，与传统的在线 Boosting 跟踪算法相比，改进的算法能有效解决目标遮挡问题。
关键词：在线 Boosting；遮挡感知；ORB 特征；目标跟踪
中图分类号：TP391.41 **文献标识码**：A

Online Boosting tracking algorithm combined with occlusion sensing

WANG Ya-wen, CHEN Hong-chang, LI Shao-mei, GAO Chao
(National Digital Switching System Engineering & Technology R&D Center, Zhengzhou 450002, China)

Abstract: Online Boosting tracking algorithm combined with occlusion sensing was presented. In this method, occlusion sensor was introduced to check the tracking results, and classifier updating strategy was adjusted depending on the occlusion checking results. By this way, the feature pool of the classifier can be kept pure, which will improve the tracking robustness under occlusion. Experimental results show that compared with traditional Boosting tracking algorithm, improved algorithm can solve the problem of occlusion very well.
Key words: online Boosting, occlusion sensing, ORB feature, object tracking

1 引言

目标跟踪的目的是在图像序列中实时提取兴趣目标的运动轨迹[1]。目标跟踪在行为分析、智能监控、交通监管等领域都扮演着重要的角色。目标跟踪作为计算机视觉领域的研究热点，近年来有了较大的发展，但同时仍然面临着复杂背景、遮挡、目标形变等许多挑战。

为了适应跟踪过程中目标外观的变化，研究人员提出了一系列的在线学习跟踪算法，主要有基于向量子空间学习的跟踪算法、基于稀疏子空间学习的跟踪算法以及基于分类器学习的跟踪算法等[2]。基于分类器学习的跟踪算法主要包括基于 SVM 分类器的跟踪算法[3]和基于 Boosting 分类器的跟踪算法[4]。由于基于 Boosting 分类器的跟踪算法具有对训练数据需求量小、实时性更好的优点，所以研究更为广泛。

Grabner[4]最先将 Boosting 算法应用到目标跟踪中，该方法首先从特征池中挑选训练误差最小的特征，然后利用它们训练弱分类器，并按照各自的权重线性组合成强分类器来进行目标跟踪。跟踪过程中，不断根据跟踪结果采集正负样本来更新 Boosting 分类器以适应目标外观变化。该算法的优点在于其较强的适应能力，能够有效跟踪外观不断变化的目标。但是该算法在进行模型更新过程中，对跟踪结果高度依赖，如果跟踪结果发生偏差，如目标受到遮挡，那么目标区域会包含背景信息，并且这些背景信息会被当成正样本，用于对目标外观

收稿日期：2016-01-25；**修回日期**：2016-08-10
基金项目：国家自然科学基金资助项目（No.61379151, No.61521003）；国家科技支撑计划基金资助项目（No.2014BAH30B01）；河南省杰出青年基金资助项目（No.144100510001）
Foundation Items：The National Natural Science Foundation of China (No.61379151, No.61521003), The National Science and Technology Support Program of China (No.2014BAH30B01), The Henan Province Science Found for Distinguished Young Scholars of China (No.144100510001)

模型进行更新，从而污染分类器特征池，影响后续跟踪。为了解决此问题，文献[5]提出基于离线 Boosting 分类器和在线 Boosting 分类器融合的跟踪算法，通过给离线分类器和在线分类器分配合理的权重来平衡算法的适应性和抗遮挡性，但是该算法对权重的选择缺乏理论指导。文献[6]将分块跟踪[7]的思想引入到在线 Boosting 跟踪算法中来解决遮挡问题。该方法将目标分成多个子区域，对每个子区域分别利用在线 Boosting 算法进行跟踪评价。该算法的缺点在于子区域的划分会降低分类器特征池中特征的数量，影响分类器性能。文献[8]引入非线性递归最小二乘法来构建自适应非线性弱分类器，然后利用 Boosting 算法将弱分类器加权组合成强分类器进行跟踪。文献[9]提出 MKB（multiple kernel Boosting）跟踪算法，该算法构建一组 SVM 分类器，每个分类器对应不同的核函数和特征，并将每一个 SVM 分类器作为一个弱分类器，然后利用 Boosting 算法将所有弱分类器加权组合成强分类器，用于目标跟踪。文献[10]将半监督学习引入到 Boosting 框架中来改善更新机制，该方法仅对第一帧样本进行标记，并用有标记的样本对新采样得到的样本进行监督，一定程度上缓解了漂移问题，但随着跟踪过程中目标外观的变化，监督机制将逐渐失去作用。文献[11]提出一种协同训练框架来提高 Boosting 分类器性能，但该框架至少需要 2 个 Boosting 分类器才能生效。文献[12]提出一种多实例学习跟踪算法，该方法用样本包的形式代替了单个样本进行标记，提高了样本标记的质量，但在遮挡环境下，该算法依然存在跟踪漂移的问题。

针对目标受到遮挡时，在线 Boosting 跟踪算法顽健性差的问题，本文提出一种融合遮挡感知的在线 Boosting 跟踪算法，有效感知跟踪过程中可能发生的遮挡，从而自适应调整分类器的更新策略，提高算法在遮挡环境下的顽健性。

2　遮挡感知器

为了提高在线 Boosting 跟踪算法在遮挡环境下的顽健性，一个合理的思路是引入遮挡感知机制，在未感知到遮挡时，采集目标区域的正样本进行分类器更新；在感知到遮挡后，自适应调整分类器更新策略，暂不进行基于正样本的分类器更新。

本节将从遮挡感知方法、局部特征选择、模糊特征过滤以及感知器更新 4 个方面进行介绍。

2.1　遮挡感知方法

如图 1 所示，遮挡物对目标进行遮挡是一个时变的过程，假设在第 k 帧，遮挡物即将遮挡目标，遮挡物一般会先出现在背景域 R_{B}（图 1 中白框以内，黑框以外的回形区域），然后出现在目标域 R_{T}（图 1 中黑框以内的区域）。遮挡感知的具体实现是通过记忆特定帧数中（如图 1 中的 m）背景域出现的局部特征，形成感知特征池 P_{B}。

$$P_{\mathrm{B}}=\{(\vec{\boldsymbol{p}},\vec{\boldsymbol{d}})\,\big|\,\vec{\boldsymbol{p}}\in R_{\mathrm{B}}\} \tag{1}$$

其中，$\vec{\boldsymbol{p}}$ 是一个表示特征点坐标的二维向量，$\vec{\boldsymbol{d}}$ 表示局部特征描述符，P_{B} 用于训练背景特征分类器 C_{B}。当新的一帧图像到来时，分类器 C_{B} 作用于目标域 R_{T}，对从 R_{T} 中提取的全体局部特征逐个进行分类。其中，与分类器的距离低于判决门限 λ_{B} 的特

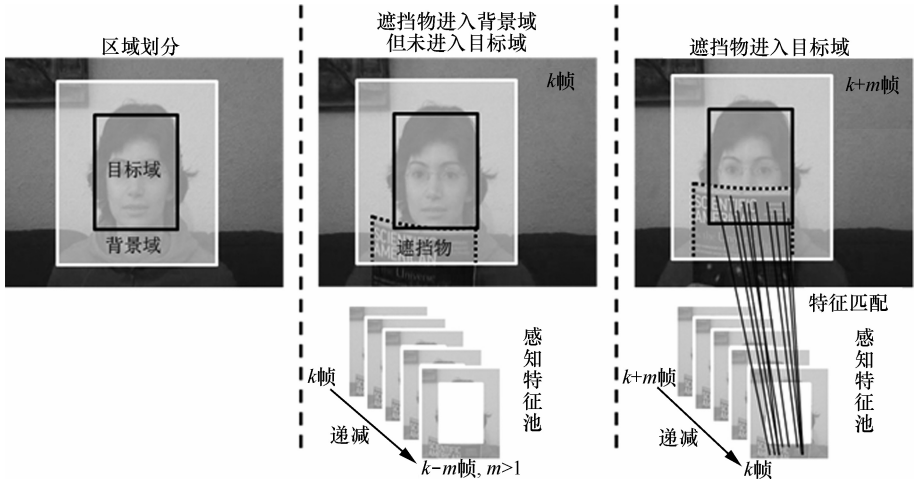


图 1　遮挡感知方法

征被归为式(2)遮挡特征集合 F_{O} 中，然后根据式(3)判断目标域是否受到遮挡。

$$F_{\mathrm{O}}=\{(\vec{\boldsymbol{p}},\vec{\boldsymbol{d}})\,\big|\,\vec{\boldsymbol{p}}\in R_{\mathrm{T}},\left\|C_{\mathrm{B}}(\vec{\boldsymbol{d}})\right\|<\lambda_{\mathrm{B}}\} \tag{2}$$

$$\begin{cases}\|F_{\mathrm{O}}\|\geqslant\lambda_{\mathrm{O}}, & \text{遮挡}\\ \|F_{\mathrm{O}}\|<\lambda_{\mathrm{O}}, & \text{未遮挡}\end{cases} \tag{3}$$

其中，$\|\cdot\|$ 表示特征集合中特征数量，λ_{O} 表示遮挡判决阈值，其设置规则如 4.2 节所述。如果感知出目标受到遮挡，则不采集受污染的正样本来更新分类器。

本文提出的遮挡感知方法是建立在局部特征匹配基础之上，因此，选择合适的局部特征是一个关键问题，下面将介绍局部特征选择依据。

2.2 局部特征选择

目前，应用最广泛的局部特征是 SIFT 及其各种变换特征，如 SURF、PCA-SIFT 等。但是这些特征普遍具有处理速度慢的特点，难以满足本文实时处理的需求。综合处理精度和速度，本文采用 Rublee[13]提出的 ORB 算法，它改进了 FAST 角点和 BRIEF 描述子无方向性的缺点，提出了方向 oFAST 角点和旋转 BRIEF 描述子并将其结合，使特征具备了旋转不变性[14]。更值得关注的是，由于旋转 BRIEF 是一种二值描述方法，因此基于 ORB 特征进行相似性匹配时，用汉明距离代替了 SIFT 特征匹配中的欧式距离，从而大大减少了匹配时间。因此本文以 ORB 局部特征为基础，构建遮挡感知器。

2.3 基于双分类器判决的模糊特征过滤

目标区域和背景区域可能存在颜色、纹理相近的局部区域，如图 2 所示，目标域（图 2 中黑框以内的区域）中的 R_1 区域与背景域（图 2 中白框以内，黑框以外的回形区域）中的 R_2 区域外观近似，因此在利用 2.1 节内容进行遮挡感知的时候，会造成遮挡误判。本文将这种存在于背景中且与目标外观高度相似的局部特征定义为模糊特征[15]（ambiguous feature），用 $F_{\mathrm{ambiguous}}$ 表示。模糊特征不仅自身不具有对目标与背景的区分能力，并且会在目标特征集合 F_{target} 和背景特征集合 $F_{\mathrm{background}}$ 之间构建过渡桥梁来弱化 F_{target} 和 $F_{\mathrm{background}}$ 之间的区分性，如式(4)。

$$F_{\mathrm{target}}\sim F_{\mathrm{ambiguous}}\sim F_{\mathrm{background}}\rightarrow F_{\mathrm{target}}\sim F_{\mathrm{background}} \tag{4}$$

其中，符号“～”表示匹配关系。

为了避免这类模糊特征对遮挡感知的影响，本文提出基于双分类器判决的模糊特征过滤方法。该方法需要增加一个与背景特征分类器 C_{B} 相对的目标特征分类器 C_{T}。初始化时，它们分别由首帧中的目标区域和背景区域的 ORB 特征训练得到。后续根据各帧的跟踪结果，不断进行特征池更新。

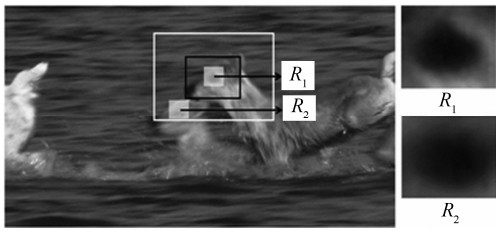


图 2 模糊特征说明

利用双分类器进行模糊特征过滤，对于新获得的目标域特征集合 $F_{\mathrm{target}}^{\mathrm{new}}$

$$F_{\mathrm{target}}^{\mathrm{new}}=\{(\vec{\boldsymbol{p}},\vec{\boldsymbol{d}})\big|\vec{\boldsymbol{p}}\in R_{\mathrm{T}}\} \tag{5}$$

其中，$\vec{\boldsymbol{p}}$ 是一个表示特征坐标的二维向量，$\vec{\boldsymbol{d}}$ 是一个代表 ORB 描述符的 n 维二进制字符串。利用分类器 C_{T} 和 C_{B} 对特征集合中的特征进行分类，根据分类结果将特征分别放入目标临时集合 S_{T} 和背景临时集合 S_{B}。

$$S_{\mathrm{T}}=\{(\vec{\boldsymbol{p}},\vec{\boldsymbol{d}})\,\big|\,\ (\vec{\boldsymbol{p}},\vec{\boldsymbol{d}})\in F_{\mathrm{target}}^{\mathrm{new}},\left\|C_{\mathrm{T}}(\vec{\boldsymbol{d}})\right\|<\lambda_{\mathrm{T}}\} \tag{6}$$

$$S_{\mathrm{B}}=\left\{(\vec{\boldsymbol{p}},\vec{\boldsymbol{d}})\,\big|\,\ (\vec{\boldsymbol{p}},\vec{\boldsymbol{d}})\in F_{\mathrm{target}}^{\mathrm{new}},\left\|C_{\mathrm{B}}(\vec{\boldsymbol{d}})\right\|<\lambda_{\mathrm{B}}\right\} \tag{7}$$

其中，λ_{T} 表示分类器 C_{T} 的判决门限，表示能被分类器 C_{T} 接受的最大汉明距离；λ_{B} 表示分类器 C_{B} 的判决门限，表示能被分类器 C_{B} 接受的最大汉明距离。本文将 λ_{T} 和 λ_{B} 均设为 45，这样那些既属于 S_{T} 又属于 S_{B} 的特征就是模糊特征，然后通过过滤模糊特征的方式来对临时特征集合提纯，获得遮挡特征集合 F_{O}。

$$F_{\mathrm{O}}=S_{\mathrm{B}}\setminus S_{\mathrm{T}} \tag{8}$$

其中，“\”符号表示差集。最后根据式(3)进行遮挡判决。

2.4 遮挡感知器的更新

为了提高遮挡感知器的适应能力，感知器需要不断更新，即丰富分类器 C_{T} 和 C_{B} 的特征池，因此，在每帧图像完成遮挡感知后，需要对 C_{T} 和 C_{B} 的特征池进行更新。

对于第 k 帧获得的目标域特征 F_{target}^{k}，过滤掉遮挡特征后，用来更新分类器 C_{T} 的特征池。

$$P_{\mathrm{T}}^{k}=\bigcup_{\tau=1}^{k}(F_{\mathrm{target}}^{\tau}\setminus F_{\mathrm{O}}^{\tau}) \tag{9}$$

其中，P_{T}^{k} 表示分类器 C_{T} 在第 k 帧时对应的特征池，F_{O}^{k} 表示第 k 帧遮挡特征。

类似地，对于第 k 帧获得的背景域特征 $F^k_{\text{background}}$，合并遮挡特征后，用来更新分类器 C_{B} 的特征池。

$$P^k_{\text{B}}=\bigcup_{\tau=1}^{k}(F^{\tau}_{\text{background}}\cup F^{\tau}_{\text{O}}) \tag{10}$$

其中，P^k_{B} 表示分类器 C_{B} 在第 k 帧时对应的特征池。

理论上，为了保证遮挡感知的效果，感知器记忆的特征越多越好，但是受存储空间和算法效率的约束，遮挡感知器记忆的 ORB 特征不可能无限制地增长，因此就需要限定遮挡感知器的记忆容量为 l。这样，分类器 C_{T} 和 C_{B} 的更新方式如式(11)、式(12)。

$$P^k_{\text{T}}=\begin{cases}\bigcup\limits_{\tau=1}^{k}(F^{\tau}_{\text{target}}\setminus F^{\tau}_{\text{O}}),\ 1\leqslant k\leqslant l\\ \bigcup\limits_{\tau=k-l}^{k}(F^{\tau}_{\text{target}}\setminus F^{\tau}_{\text{O}}),\ k>l\end{cases} \tag{11}$$

$$P^k_{\text{B}}=\begin{cases}\bigcup\limits_{\tau=1}^{k}(F^{\tau}_{\text{background}}\cup F^{\tau}_{\text{O}}),\ 1\leqslant k\leqslant l\\ \bigcup\limits_{\tau=k-l}^{k}(F^{\tau}_{\text{background}}\cup F^{\tau}_{\text{O}}),\ k>l\end{cases} \tag{12}$$

很明显，记忆容量的限定在一定程度上会影响遮挡感知器的判决准确度。如假设分类器只有 10 帧的记忆，那么当遮挡持续 10 帧之后，背景分类器 C_{B} 中能有效区分遮挡的特征就会逐渐被遗忘，随着时间的推移，遮挡感知器将失去对遮挡情况的正确感知，从而跟踪器再次面临漂移问题。实验部分将会详细分析遮挡感知器的记忆容量对于算法性能的影响，并给出其设置依据。

整个遮挡感知器的结构如图 3 所示。

3　遮挡感知器与跟踪器融合

本文提出了一种遮挡感知方法，利用该方法来监督在线 Boosting 跟踪算法中分类器的更新机制，避免采集受污染的正样本对分类器进行错误的更新，保证 Boosting 分类器特征池的纯净，从根本上解决在线 Boosting 跟踪算法面对目标遮挡时产生的漂移问题。

改进的算法分别构建遮挡感知器和跟踪器，针对每帧图像，将跟踪器的跟踪结果作为感知区域，并利用遮挡感知器对区域图像进行检测，根据返回的遮挡感知结果监督跟踪器的更新。改进的算法步骤如算法 1 所示。

算法 1　改进的跟踪算法

target=人为标记要跟踪的目标；

对在线 Boosting 分类器进行初始化；

对目标特征分类器 C_{T} 和背景特征分类器 C_{B} 初始化；

while 获得一帧新的图像 do

sample[ ]=采集样本；

target=Boosting(sample[ ])；

F[ ]=从 target 中提取 ORB 特征；

C_{T} 和 C_{B} 对 F[ ]分类并过滤模糊特征得到 F_{O}[ ]；

if(size(F_{O}[ ])> λ_{O})

只采集负样本更新 Boosting 分类器；

else

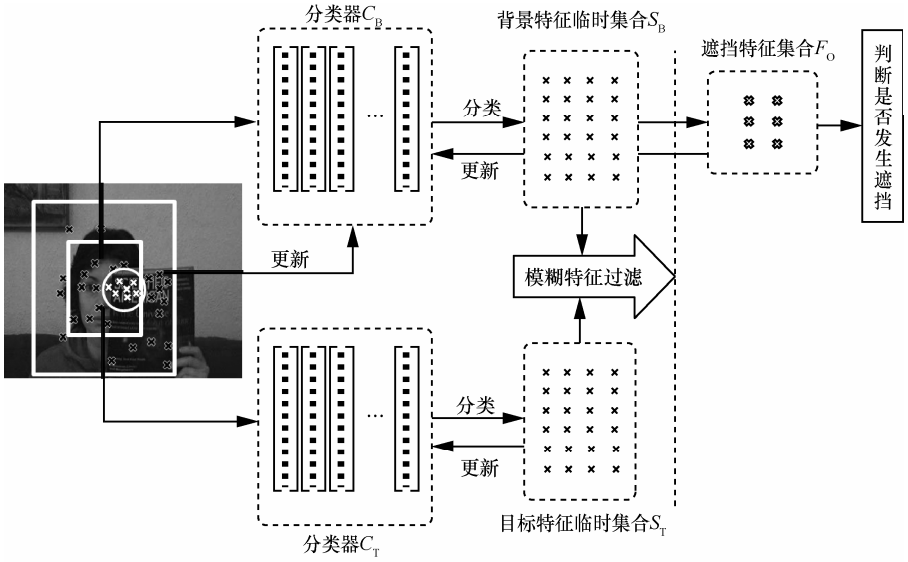


图 3　遮挡感知器结构

采集正负样本更新 Boosting 分类器；

end

更新遮挡感知器；

end

遮挡感知器与跟踪器是相互依赖的关系，遮挡感知域的构建依赖跟踪器的跟踪结果，而同时跟踪器的更新策略依赖于遮挡感知器的遮挡检测结果。并且遮挡感知器与跟踪器也是相互促进的关系，有效的遮挡感知能优化分类器的更新策略，提高跟踪器的跟踪精度，而同时准确的目标定位又会帮助遮挡感知器构建准确的感知域，保证遮挡感知器的有效应用。

4 实验结果及分析

因为本文的核心思想是对在线 Boosting 跟踪算法引入遮挡感知机制以提高其在遮挡环境下的顽健性，所以实验部分首先对遮挡感知器进行性能测试，其中包括遮挡漏检率和误检率的统计、记忆容量的确定；随后挑选了传统在线 Boosting 跟踪算法[4]（OAB）、半监督 Boosting 跟踪算法[10]（SBT）、多实例学习跟踪算法[12]（MIL）以及一种静态抗遮挡跟踪算法[7]（Frag）进行遮挡环境下的跟踪对比实验，评估本文算法面对遮挡问题的顽健性；最后利用 12 幅图片序列对本文算法与文献[16]中提到的 10 种现阶段著名跟踪算法进行综合性能对比，并深入分析本文算法的适应性。实验中用到的标准测试序列来源于文献[16]。

4.1 实验参数

1) Boosting 算法参数

参考已有的在线 Boosting 跟踪算法，选择器数量为 30，每个选择器包含的弱分类器个数为 100，采样区域覆盖率为 0.95，搜索系数为 2。

2) ORB 算法参数

ORB 算法中基于 oFAST 角点检测来定位兴趣点，为了能使 oFAST 算法在跟踪窗口中检测出尽可能多的角点，本文采用 FAST-9 进行角点检测，其原理是对每一个候选点周围的 16 个像素点进行检测，如果存在连续 9 个像素点比候选点像素值加上数值 T 后（T 为任意整数）还大或是比候选点像素值减去数值 T 后还小，则该候选点就是角点，参数 T 设为 25。

3) 实验环境

电脑配置为 3.2 GHz CPU，7.86 GB 内存，实验平台为 VS2010。

4.2 遮挡感知器性能测试

本节实验利用标准测试序列 faceOcc1 和 faceOcc2 来测试遮挡感知器性能，其方法是利用已标记好的真实目标区域构建感知区域，利用不同的遮挡阈值 λ_o 来测试遮挡感知器的效果。统计在不同遮挡阈值条件下，遮挡感知器的漏检率（目标区域实际受到遮挡，但未触发遮挡感知器）和误检率（目标区域实际未受遮挡，但触发了遮挡感知器），其结果如表 1 所示。

从表 1 结果来看，对于序列 faceOcc1，当遮挡阈值 λ_o 定为 50 时，其感知结果较为理想，对于序列 faceOcc2，当遮挡阈值 λ_o 定为 35 时，其感知结果较为理想。但是本节实验是利用标记好的真实跟踪区域来测试，实际跟踪中，由于窗口与目标会产生一定偏差，因此实际感知结果要弱于表 1 结果。

因为从一幅图片的某个区域提取出的 ORB 特征数量与该区域的大小有着直接关系，在对 faceOcc1 序列进行测试的时候，跟踪窗口大小为 114×162 像素，而在对 faceOcc2 序列进行测试的时候，跟踪窗口大小为 82×98 像素。因此后续实验中，遮挡阈值 λ_o 大小的设置依据为

$$\lambda_o = \left(\frac{50}{114\times162}+\frac{35}{82\times98}\right)\times 0.5\times \text{区域面积} \quad (13)$$

4.3 评估遮挡感知器记忆容量对算法性能影响

前面已经讨论过遮挡感知器的更新是一个两难选择，从遮挡感知准确度的角度来考虑，感知器记忆的 ORB 特征一定是越多越好，但是从存储空间和算法效率来考虑，感知器记忆的 ORB 特征不宜太多。本节实验利用遮挡持续时间最长的 faceOcc1 来测试。测试过程中，分别限定遮挡感知器的记忆容量为 10 帧、20 帧、30 帧和 40 帧，并统计每帧图像的定位误差（定位误差用跟踪窗口位置和目标实际位置间的距离来衡量），其结果如图 4 (a)所示。从图中可以看出，前 50 帧时 4 条曲线基本吻合，50 帧之后，遮挡情况出现，由于遮挡感知器记忆的特征数量不一样导致判决结果不一致，从而使 Boosting 分类器维护的特征池发生变化，所以 4 条曲线逐渐分离。4 条曲线的分布表明，遮挡感知器记忆容量的不同会对跟踪器的定位产生较大影响，随着遮挡感知器记忆的特征数量的减小，跟踪器的定位误差会逐渐上升。但同时，根据图 4 (b)结果可以发现，记忆容量的限定有助于提高算法效

率。如图 4 (b)所示，当记忆容量为 40 帧时，每秒只能处理 5.5 帧图像；当记忆容量降到 10 帧时，每秒能处理至少 9 帧图像。

表 1　　遮挡感知器性能测试

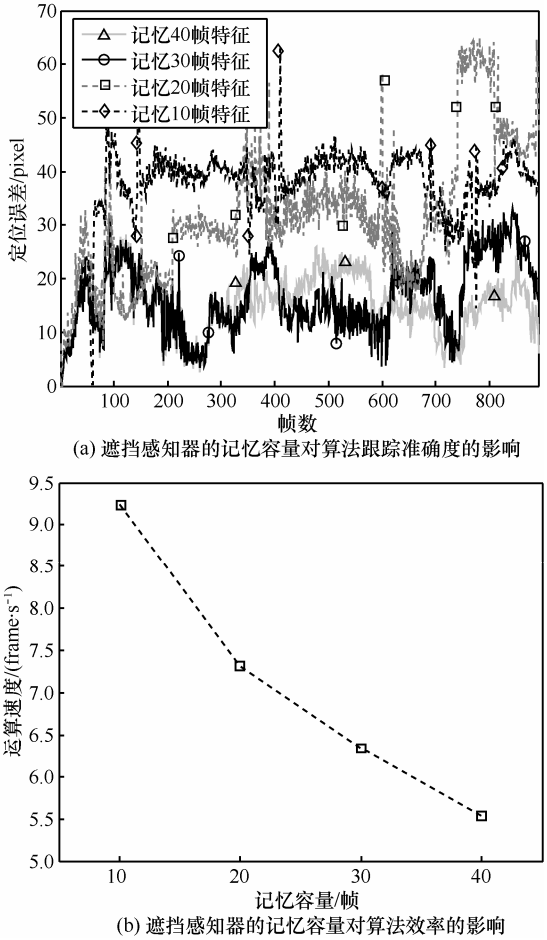
λ_o	利用 faceOcc1 测试的结果		利用 faceOcc2 测试的结果	
	漏检率	误检率	漏检率	误检率
5	3.6%	55.2%	4.4%	55.4%
10	5.1%	48.8%	6.4%	51.7%
15	5.9%	40.1%	6.8%	29.1%
20	9.2%	30.0%	10.8%	21.6%
25	11.5%	22.4%	15.5%	16.1%
30	14.2%	17.7%	17.0%	10.1%
35	14.8%	15.4%	17.9%	7.2%
40	15.1%	10.7%	23.6%	6.4%
45	15.6%	7.0%	24.8%	6.1%
50	16.0%	6.4%	32.7%	5.9%
55	19.8%	4.9%	36.4%	5.9%
60	25.4%	3.3%	41.9%	5.7%
65	29.0%	2.9%	50.5%	5.4%
70	35.3%	1.2%	54.9%	5.1%
75	40.8%	0.8%	57.7%	4.7%
80	48.4%	0.6%	62.7%	4.6%

综合考虑算法性能和效率，后续实验中，设置遮挡感知器的记忆容量为 30 帧，即每次保存最新的 30 帧图像的 ORB 特征。

4.4　遮挡环境下跟踪算法性能比较

本文挑选了传统在线 Boosting 跟踪算法（OAB）、半监督 Boosting 跟踪算法（SBT）、多实例学习跟踪算法（MIL）以及一种静态抗遮挡跟踪算法（Frag）与本文算法进行比较，利用 6 种包含遮挡内容的序列 coke、tiger2、faceOcc1、faceOcc2、walking2 和 woman 进行测试（粗略定义遮挡情况为目标被遮挡的区域超过 $\frac{1}{4}$）。每帧图像定位误差（定位误差用跟踪窗口位置和目标实际位置间的距离来衡量）和跟踪结果分别如图 5 和图 6 所示。

标准测试序列 coke 在第 39、165、220、253 帧处出现遮挡情况，其中，在第 220 和 253 帧处出现严重遮挡（目标被遮挡的区域超过 $\frac{3}{4}$），但从图 5 中可以看出在第 39、165、220 帧处，本文算法的定位误差维持在较小数值上，而在第 253 帧处，虽然本文算法的定位误差有所增加，但依然保持在可控范围之内，没有丢失目标。



(a) 遮挡感知器的记忆容量对算法跟踪准确度的影响

(b) 遮挡感知器的记忆容量对算法效率的影响

图 4　遮挡感知器的记忆容量对算法性能的影响

标准测试序列 tiger2 不仅包含有遮挡内容，同时目标外观还在不断发生变化，因此跟踪难度较大。其中，在第 97、253、337 帧等处出现遮挡情况，同时在第 41、125、273、316 帧等处目标外观发生较大程度的改变。本文算法由于增加了遮挡感知器，能有效降低遮挡问题对算法的影响。此外在线 Boosting 跟踪算法本身具有较强的在线学习适应能力，能有效跟踪外观变化的目标，因此在 tiger2 序列中，本文算法能对目标进行有效地跟踪。

faceOcc1 和 faceOcc2 是用来测试跟踪算法面对遮挡问题顽健性的标准测试序列，其中，faceOcc1 在 115、177、308、496、828 帧等处出现遮挡情况，由于在该序列中，遮挡持续时间较长，超出了遮挡感知器的记忆容量，导致定位误差出现局部波动，但本文算法自始至终没有出现窗口漂移现象。faceOcc2 在 141、420、702 帧等处出现遮挡情况，在 343、594

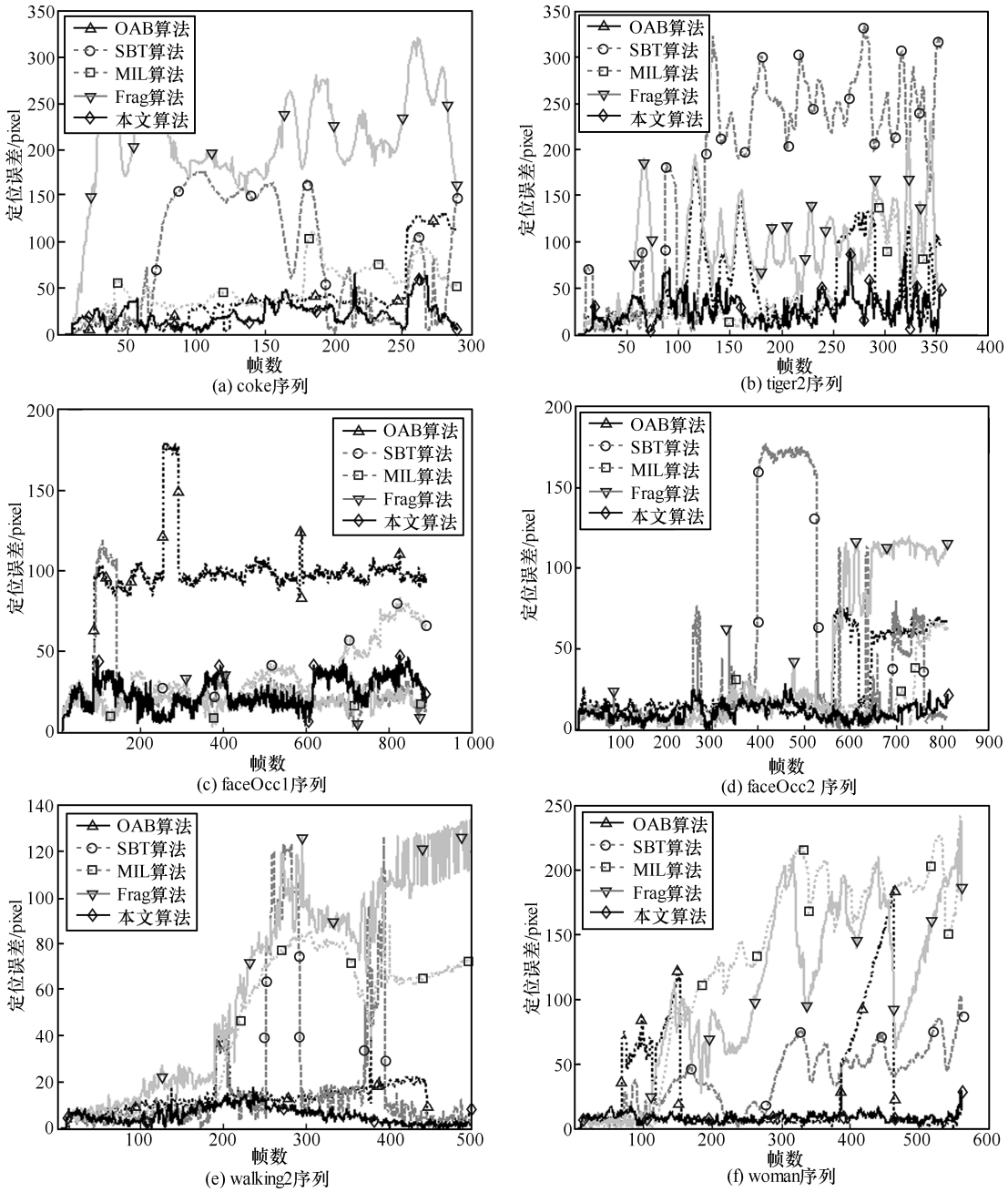


图 5　定位误差定量分析

帧等处目标外观发生变化，但无论是遮挡情况还是目标外观变化，本文算法始终保持准确的目标定位。

标准测试序列 walking2 在 204 帧处出现目标遮挡情况，并且目标尺度发生变化，本文算法虽然不具备尺度调节功能，但是由于目标尺度是缓慢变化的，因此跟踪算法能够通过在线学习的方式有效适应这种变化，不会出现丢失目标的情况。

标准测试序列 woman 在 133、390 帧等处出现目标遮档情况，此外目标还发生一定程度的形态和尺度变化，但是从图 6 结果来看，本文算法能有效克服目标遮挡以及目标形态、尺度变化对跟踪算法的影响。

为了检验遮挡感知器在实际跟踪过程中的性能，本文统计遮挡感知器在实际跟踪过程中的误检率和漏检率，结果如表 2 所示，可以看出在实际跟踪过程中，遮挡感知器能够正常应用，但依然存在着一定程度的漏检和误检，下面将分析遮挡漏检和误检对跟踪算法的影响。

在线 Boosting 跟踪算法产生窗口漂移的原因是不正确的样本训练产生误差累积，因此遮挡感知器的作用就是延缓 Boosting 分类器的误差累积过程，即使

图 6　6 种测试序列的跟踪结果

一些遮挡情况被漏判，也不会对跟踪器产生严重后果。对于遮挡误判的情况，会限制跟踪器的学习能力，降低算法适应性，但是从表 2 结果看出，误检率普遍较低，不会对跟踪算法的适应性产生根本影响。

表 2　遮挡感知器实际漏检率和误检率统计

测试序列名称	遮挡感知器漏检率	遮挡感知器误检率
coke	23.1%	14.2%
tiger2	26.4%	18.5%
faceOcc1	22.9%	14.9%
faceOcc2	23.7%	10.6%
walking2	18.0%	12.3%
woman	12.0%	8.7%

4.5　与其他顽健跟踪算法的跟踪性能比较

4.4 节实验说明本文算法对遮挡问题具有较好的顽健性，为了进一步评估本文算法全方位的性能，根据文献[16]的研究成果挑选了 10 种现阶段性能较好的跟踪算法（TLD、MIL、CT、DFT、IVT、SCM、VTD、ASLA、Struck、CSK）做对比，在 12 幅图片序列（deer、girl、mountainBike、singer2、fish、soccer、doll、liquor、matrix、motorRolling、shaking、walking1）上进行测试，测试序列包括目标形变、光照变化、复杂背景、运动模糊等挑战内容。每种算法在所有测试序列中的平均定位误差记录在表 3 中，其中，定位误差用跟踪窗口位置与目标真实位置之间的距离来

表示，单位是像素。表格中最右列的数据是利用本文跟踪算法获得的平均定位误差，括号内的数值表示本文算法就定位准确度而言，在 11 种跟踪算法中的排名。

下面就目标外观变化、运动模糊、目标尺度变化、复杂背景干扰等方面对算法进行适应性分析。

目标外观变化。当利用包含有目标外观变化的标准测试序列 motorRolling 和 girl 进行测试时，本文算法的表现较好，在 11 种跟踪算法比较中分别排第 1、第 2。出现这样的情况是因为在线 Boosting 跟踪算法是一种在线学习跟踪算法，该算法通过不断地采集正负样本来训练分类器使分类器能够较好地适应目标外观的变化。

运动模糊。由于本文算法采用的是 Haar-like 特征，对图像模糊顽健性较好，因此本文算法在面对运动模糊问题时表现较好，如利用标准测试序列 fish、deer 和 mountainBike 进行测试时，在 11 种跟踪算法比较中分别排第 1、第 2、第 2。

目标尺度变化。由于本文算法无法自适应调节跟踪窗口大小，因此在跟踪尺度发生变化的目标时表现一般，如利用标准测试序列 walking1、doll 和 singer2 进行测试的时候，在 11 种跟踪算法比较中分别排第 5、第 4、第 3。

复杂背景干扰。从表 3 可以看出，当利用标准测试序列 matrix、soccer 和 shaking 进行测试的时候，本文算法的表现很不理想，在 11 种跟踪算法比较中分别排第 6、第 6、第 4。matrix、soccer 以及 shaking 都是包含有复杂背景干扰的标准测试序列，其中，序列 matrix 背景中洒落的雨水以及闪电带来的亮度变化、序列 soccer 背景中漂浮的红色烟花、序列 shaking 背景中灯光的变化都属于复杂背景干扰的范畴。由于跟踪窗口中除了跟踪目标外，不可避免的会包含一定量的背景信息，因此复杂的背景变化依然会对分类器特征池产生一定量的污染，导致产生窗口漂移现象，所以本文算法在复杂背景下跟踪的顽健性较差。

4.6 算法处理效率测试

为了评估本文算法的处理效率，分别统计了本文算法与 OAB、SBT、MIL 和 Frag 在 6 个测试序列（coke、tiger2、faceOcc1、faceOcc2、walking2、woman）上的处理速度，如表 4 所示，表中数据记录的是 5 种算法分别在 6 个测试序列上处理速度的平均值，单位为每秒处理的帧数，表中最右侧数据为本文算法的处理速度。可以看出，由于本文改进在线 Boosting 跟踪算法增加了遮挡感知器，所以运算复杂度有所增加，因此和 OAB 算法相比，本文算法在效率方面有所下降，但和 SBT 以及 Frag 算法相比，本文算法在运算速率方面依然具有一定优势。另外需要说明的是，表 4 结果是在遮挡感知器记忆容量为 30 帧特征的条件下测得的，如果想要进一步提高算法速度，可以通过减小遮挡感知器的记忆容量来实现。

表 3 与现有跟踪算法在定位准确度（像素）方面的比较

序列	TLD	MIL	ASLA	Struck	VTD	IVT	CT	DFT	SCM	CSK	本文算法（排名）
fish	13.1	19.2	8.8	11.3	9.4	10.7	14.9	20.4	9.5	21.4	7.7(1)
motorRolling	173.0	163.9	201.6	149.3	161.1	181.4	160.6	177.8	157.6	424.4	127.5(1)
liquor	36.4	140.5	71.0	65.6	100.7	112.8	175.3	220.4	81.9	127.3	49.2(2)
deer	25.1	55.4	142.1	10.6	207.5	179.8	233.5	259.7	75.9	363.0	15.8(2)
girl	9.8	16.9	5.3	4.6	11.6	26.8	17.3	23.2	4.1	14.7	4.2(2)
mountainBike	193.0	7.5	23.5	11.7	9.7	13.3	188.8	8.4	17.4	8.2	8.0(2)
singer2	196.6	169.1	9.7	166.6	184.2	182.4	182.1	39.4	111.5	183.4	99.6(3)
doll	7.9	21.7	17.4	30.3	8.9	16.2	11.1	94.5	7.4	437.8	13.0(4)
shaking	36.6	14.1	19.5	106.4	13.2	85.5	161.9	102.4	13.8	169.2	18.6(4)
walking1	95.4	4.4	2.9	5.1	5.4	2.9	5.4	81.8	3.6	7.1	4.8(5)
soccer	72.6	36.8	121.2	110.2	22.6	146.5	80.0	112.0	157.7	30.8	104.0(6)
matrix	55.2	41.9	61.6	167.1	70.2	109.8	72.4	127.8	51.9	65.1	69.4(6)

表 4　　跟踪算法运算速度比较

跟踪算法	运算速度/(frame·s^{-1})
OAB	15.23
SBT	7.47
MIL	34.40
Frag	4.47
本文算法	8.84

5　结束语

本文针对传统在线 Boosting 跟踪算法在目标发生遮挡时易造成窗口漂移的缺点，引入遮挡感知机制，对目标是否受到遮挡进行检测，并根据检测结果调整分类器更新策略，避免了采集受污染的正样本进行训练，从而维护分类器特征池的纯净。为了提高遮挡感知的精度，对遮挡感知中的特征选择、过滤、匹配及更新等各个环节进行了设计，保证了算法的应用效果。本文在实验部分对遮挡感知器的性能和算法的整体跟踪性能进行了测试，并对跟踪算法的适应性进行了深入分析和讨论，实验结果表明，本文算法在遮挡条件下具有较好的顽健性。值得一提的是，本文提出的遮挡感知算法还可以与其他的在线自适应跟踪算法相结合，提高这些算法应对遮挡问题的顽健性。

参考文献：

[1] 罗会兰, 杜芳芳, 孔繁胜. 像素点特征加权的尺度自适应跟踪算法[J]. 通信学报, 2015, 36(10): 200-211.
LUO H L, DU F F, KONG F S. Pixel feature-weighted scale-adaptive object tracking algorithm[J]. Journal on Communication, 2015, 36(10): 200-211.

[2] 张焕龙, 胡士强, 杨国胜. 基于外观模型学习的视频目标跟踪方法综述[J]. 计算机研究与发展, 2015, 52(1): 177-190.
ZHANG H L, HU S Q, YANG G S. Video object tracking based on appearance models learning[J]. Journal of Computer Research and Development, 2015, 52(1):177-190.

[3] BAI Y, TANG M. Robust tracking via weakly supervised ranking SVM[C]//The IEEE Conference on Computer Vision and Pattern Recognition. 2012: 1854-1861.

[4] GRABNER H, BISCHOF H. On-line boosting and vision[C]//The IEEE Conference on Computer Vision and Pattern Recognition. 2006: 260-267.

[5] 程有龙, 李斌, 张文聪, 等. 融合先验知识的自适应行人跟踪算法[J]. 模式识别与人工智能, 2009, 22(5): 704-708.
CHENG Y L, LI B, ZHANG W C, et al. An adaptive pedestrian tracking algorithm with prior knowledge[J]. PR & AI, 2009, 22(5): 704-708.

[6] 颜佳, 吴敏渊. 遮挡环境下采用在线 Boosting 的目标跟踪[J]. 光学精密工程, 2012, 20(2): 439-446.
YAN J, WU M Y. On-line boosting based target tracking under occlusion[J]. Optics and Precision Engineering, 2012, 20(2): 439-446.

[7] ADAM A, RIVLIN E, SHIMSHONI I. Robust fragments-based tracking using the integral histogram[C]//The IEEE Conference on Computer Vision and Pattern Recognition. 2006: 798-805.

[8] HU H, MA B, WU Y, et al. Kernel regression based online boosting tracking[J]. Journal of Information Science and Engineering, 2015, 31(1): 267-282.

[9] YANG F, LU H, YANG M H. Robust visual tracking via multiple kernel boosting with affinity constraints[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2014, 24(2): 242-254.

[10] GRABNER H, LEISTNER C, BISCHOF H. Semi-supervised on-line boosting for robust tracking[C]//The European Conference on Computer Vision. 2008: 234-247.

[11] 陈思, 苏松志, 李绍滋, 等. 基于在线半监督 Boosting 的协同训练目标跟踪算法[J]. 电子与信息学报, 2014, 36(4): 888-895.
CHEN S, SU S Z, LI S Z, et al. A novel co-training object tracking algorithm based on online semi-supervised Boosting[J]. Journal of Electronics & Information Technology, 2014, 36(4): 888-895.

[12] BABENKO B, YANG M H, BELONGIE S. Robust object tracking with online multiple instance learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(8): 1619-1632.

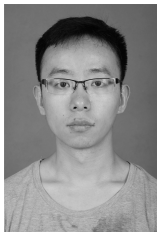
[13] RUBLEE E, RABAUD V, KONOLIGE K, et al. ORB: an efficient alternative to SIFT or SURF[C]//The International Conference on Computer Vision. 2011: 2564-2571.

[14] 唐剑琪, 谢林江, 袁庆生, 等. 基于 ORB 的镜头边界检测算法[J]. 通信学报，2013，34(11): 184-190.
TANG J Q, XIE L J, YUAN Q S, et al. Shot boundary detection algorithm based on ORB[J]. Journal on Communication, 2013, 34(11): 184-190.

[15] PERNICI F, DEL B A. Object tracking by oversampling local features[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(12): 2538-2551.

[16] WU Y, LIM J, YANG M H. Online object tracking: a benchmark[C]//The IEEE Conference on Computer Vision and Pattern Recognition. 2013: 2411-2418.

作者简介：



王亚文（1990-），男，河南郑州人，硕士，国家数字交换系统工程技术研究中心博士生，主要研究方向为多媒体技术、模式识别、计算机视觉等。

陈鸿昶（1964-），男，河南新密人，国家数字交换系统工程技术研究中心教授、博士生导师，主要研究方向为电信网安全防护技术。

李邵梅（1982-），女，湖北钟祥人，博士，国家数字交换系统工程技术研究中心讲师，主要研究方向为多媒体技术、模式识别、计算机视觉等。

高超（1982-），男，河南新郑人，博士，国家数字交换系统工程技术研究中心讲师，主要研究方向为多媒体技术、模式识别、计算机视觉等。