控制与决策
Control and Decision

特征响应权重自适应的IoU网络跟踪算法改进

陈志旺, 王莹, 宋娟, 刁华康, 彭勇

引用本文:
陈志旺,王莹,宋娟,刁华康,彭勇. 特征响应权重自适应的IoU网络跟踪算法改进[J]. 控制与决策, 2022, 37(7): 1752−1762.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.0148

特征响应权重自适应的IoU网络跟踪算法改进

陈志旺[1,2†], 王 莹[1], 宋 娟[3], 刁华康[1], 彭 勇[4]

(1. 燕山大学 智能控制系统与智能装备教育部工程研究中心，河北 秦皇岛 066004；2. 燕山大学 工业计算机控制工程河北省重点实验室，河北 秦皇岛 066004；3. 国网黑龙江省电力有限公司 佳木斯供电公司, 黑龙江 佳木斯 154002；4. 燕山大学 电气工程学院，河北 秦皇岛 066004)

摘 要: 基于IoU网络提出一种IT-AWCR(IoU network tracking with adaptive weighted characteristic responses)目标跟踪算法. 首先,根据目标运动速度设计目标搜索区域确定策略,通过理论分析使用ResNet50的block 3、block 4卷积块的输出分别作为目标的浅层和深层特征表示;然后,以目标定位准确度和滤波模型抗干扰能力为评价指标,通过优化算法自适应计算目标深、浅特征响应加权权重,从加权融合响应中获取目标粗略位置和边界框,经扰动操作获取多个候选边界框输入IoU调制-预测网络预测IoU值,取最大IoU对应边界框为最终预测目标边界框;最后,根据训练样本的相关学习权重和样本间相似度更新生成样本集,基于样本集采用稀疏优化策略实现滤波模型更新. OTB2015和VOT2018数据集上的实验结果验证了所提出算法的有效性.

关键词: 目标跟踪；IoU；ResNet50；权重自适应；样本集更新；滤波模型

中图分类号: TP391.4 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.0148 **开放科学(资源服务)标识码(OSID):**

引用格式: 陈志旺,王莹,宋娟,等. 特征响应权重自适应的IoU网络跟踪算法改进[J]. 控制与决策, 2022, 37(7): 1752-1762.

Improvement of IoU network tracking with adaptive weighted characteristic responses

CHEN Zhi-wang[1,2†], *WANG Ying*[1], *SONG Juan*[3], *DIAO Hua-kang*[1], *PENG Yong*[4]

(1. Engineering Research Center of the Ministry of Education for Intelligent Control System and Intelligent Equipment, Yanshan University, Qinhuangdao 066004, China; 2. Key Laboratory of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China; 3. Jiamusi Electric Power Company, State Grid Heilongjiang Electric Power Co., Ltd., Jiamusi 154002, China; 4. School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China)

Abstract: A target tracking algorithm of IoU network tracking with adaptive weighted characteristic responses (IT-AWCR) based on the IoU network is proposed in this paper. Firstly, a determination strategy for the target searching area is designed according to the velocity of a target, the outputs of block3 and block4 convolutional layers in ResNet50 are used as the shallow and deep feature representations of the target respectively by means of theoretical analysis. Then, with performance indexes of the accuracy of target location and the anti-interference ability of the filter model, the weights of the target deep and shallow feature responses are computed adaptively through the optimization algorithm. The rough target position and bounding box are obtained by the weighted fusion response, and multiple candidate bounding boxes are obtained by perturbation operation, which are input into the IoU modulation-prediction network to predict IoU values, taking the bounding box corresponding to the largest IoU as the final predicted target bounding box. Finally, the sample set is updated according to the relevant learning weights of the training samples and the similarities between this samples. Based on the sample set, the sparse optimization strategy is used to achieve the filter model update. The results of experiments on the OTB2015 and VOT2018 show the effectiveness of the proposed algorithm.

Keywords: object tracking; IoU; ResNet50; adaptive weighted; sample set update; filter model

0 引 言

计算机视觉是人工智能领域中的一个研究热点,目标跟踪是该研究热点的一个重要方向. 目标跟踪技术在高级人机交互、智能视频监控和自动驾驶等现实生活中有着广泛的应用前景,但由于真实跟踪场景中目标的多变性和场景的复杂性,仍需深入研究高

收稿日期: 2021-01-24；**录用日期:** 2021-04-21.
基金项目: 国家自然科学基金项目(61573305).
†通讯作者. E-mail: czwaaron@ysu.edu.cn.

性能的跟踪算法.

随着深度学习的不断发展,将深度神经网络用于目标跟踪算法引起了国内外专家学者的广泛关注. 目前,基于深度学习的主流跟踪算法有在线更新的相关滤波跟踪算法和应用线下训练的孪生网络跟踪算法. 对于基于深度学习的相关滤波跟踪算法,文献[1]提出了经典算法ECO(efficient convolution operators for tracking),其通过对融合卷积特征进行降维、简化样本集、降低滤波模型更新频率,有效提高了跟踪速度;文献[2]提出了STRCF(learning spatial-temporal regularized correlation filters for visual tracking),将空间正则化和时间正则化应用到相关滤波框架中,解决了目标大尺度变化问题;文献[3]提出了LADCF(learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking)算法,通过一种自适应的特征选择策略灵活压缩了时间信息,保证跟踪精度的同时提升了跟踪速度;文献[4]提出的ROAM(recurrently optimizing tracking model)将跟踪模型分为生成响应和边界框回归两部分,其每部分都使用可调整大小的相关滤波模型来适应目标形态变化,同时提升了跟踪速度和精度.

以上结合深度学习和相关滤波的跟踪算法,通过在线自适应学习强辨识性的深度卷积特征,很好地捕捉了跟踪过程中目标因形变导致的变化信息,通过学习得到了强抗干扰力的滤波模型,提高了跟踪精度. 但因深度神经网络结构较为复杂,含参量巨大,导致特征提取时间消耗较大,又加上在线迭代更新滤波模型的计算量大,致使算法跟踪速度大大减弱. 对此,专家学者们提出了基于孪生网络的一系列跟踪算法,这类算法利用端到端的孪生网络结构将目标跟踪问题转化为相似性学习问题. 文献[5]将跟踪看作局部的一阶段检测任务,引入faster R-CNN中的区域推荐网络;文献[6]提出了SiamRPN(high performance visual tracking with siamese region proposal network)算法,使用边界框回归代替目标多尺度搜索,利用可变宽高比的边界框估计目标位置和目标尺寸大小,获得了一个高精度的目标边界框,提高了算法的跟踪精度;文献[7]提出的SiamRPN++通过逐通道互相关操作进一步提升了算法精度;文献[8]提出的CGACD(correlation-guided attention for corner detection based visual tracking)先采用孪生网络将目标与背景区分开获得目标的ROI(region of interest)区域,然后使用逐像素相关性引导的空间注意模块和逐通道相关性引导的通道注意模块得到目标模型与ROI之间的关系,提高了角点检测的准确性,提升了算法边界框估计的精度.

以上基于孪生网络的跟踪算法,由于利用大量数据集中图像分类的标注数据训练端到端的孪生网络,采用离线方式实现跟踪,大大提升了算法速度,又使用RPN等方法充分利用目标先验特征替代了传统的目标多尺度搜索,获得了高精度目标边界框. 但这些方法因没有模型在线更新,一旦遇到离线训练使用的训练集中未包含跟踪目标,将无法获得视频帧的上下文信息. 此外,离线训练的网络一般是针对目标分类的,并不完全适应具有复杂多变性的目标跟踪任务,只有在线更新模型才能更好地及时适应目标、背景在跟踪过程中的变化. 对此,文献[9]将跟踪分为目标估计和分类两部分,其基于文献[10]构造、离线训练了一个类似于孪生网络的IoU调制-预测器,用于目标估计,使用两层卷积核作为在线更新模型用于分类,使算法增速的同时进一步提高了精度.

综上所述,文献[9]吸收了相关滤波跟踪算法模型在线更新以及基于孪生网络跟踪算法的孪生结构的优势,进一步提高了跟踪算法的精度. 本文受此启发,又考虑到目标的浅层特征空间信息强,深层特征语义信息强,为结合两者优势使算法性能最优化,提出具有特征响应权重自适应的IoU网络在线跟踪算法IT-AWCR,该算法主要包括特征提取、目标估计、模型更新3部分. 其中:特征提取部分设计了目标搜索区域确定策略并使用ResNet50[11]作为特征提取主干网络;目标估计主要包含特征响应权重自适应计算和最大IoU预测两部分,IoU预测过程使用的孪生结构IoU调制-预测网络与传统基于孪生网络的跟踪算法不同,不涉及目标分类任务,且调制功能分支在跟踪过程中仅使用一次,利用第1帧图像产生调制向量用于后续所有帧,旨在进一步优化特征响应自适应加权的目标估计结果;模型更新属于本文算法的分类部分,旨在应用有效样本集训练一个强大的滤波模型.

1　IT-AWCR跟踪算法

本文提出的IT-AWCR目标跟踪算法主要由特征提取模块、目标估计模块和模型更新模块3部分构成,跟踪过程如图1所示. 图1中:$\otimes$表示卷积运算,$\oplus$表示求和运算.

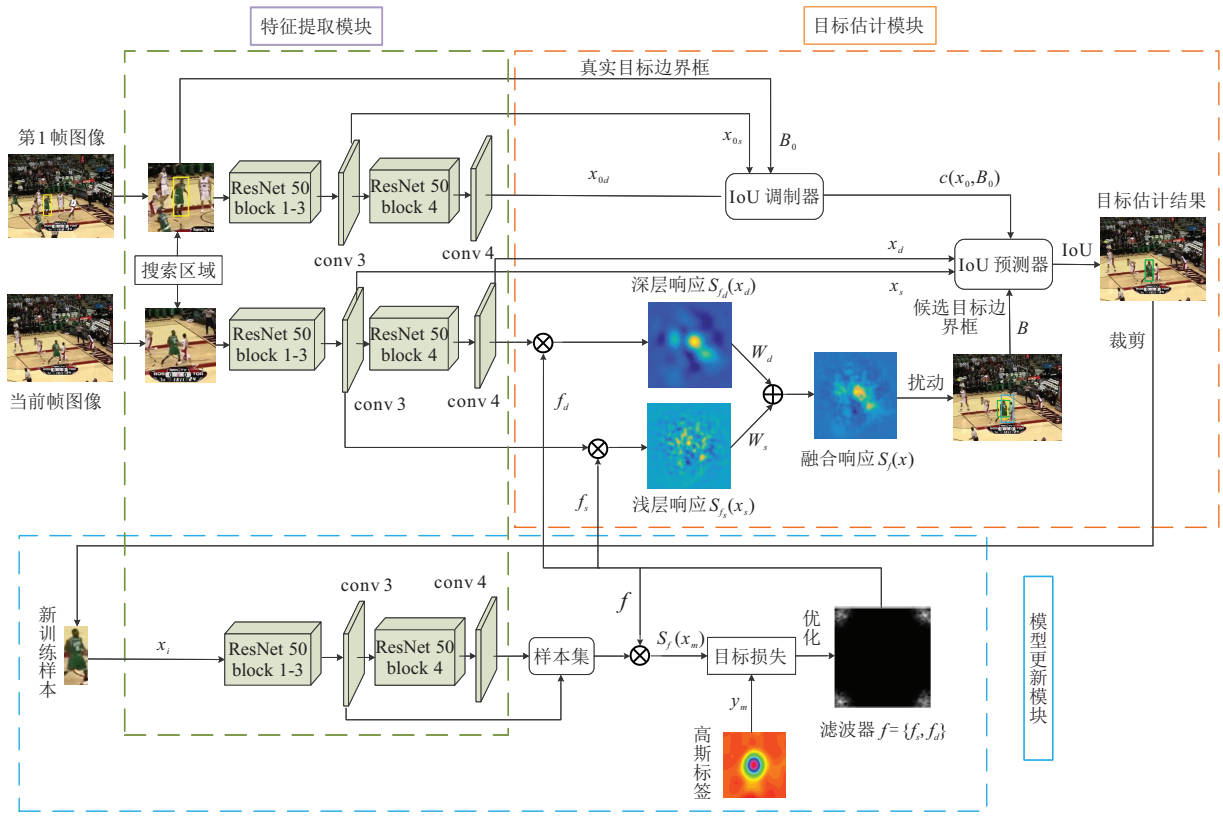

图 1 IT-AWCR 在线跟踪框架

1.1 特征提取模块

1.1.1 搜索区域的确定

如图1特征提取模块所示,对于视频序列的每一帧图像都需先确定目标搜索区域. 搜索区域过大,则模型会学习到很多无用的背景信息,甚至因一些背景干扰导致模型效果下降,产生跟踪漂移现象,且会增加算法复杂度;搜索区域过小可能包含目标信息不全,甚至目标缺失,导致模型学习到很多负面信息,使算法稳定性下降.传统目标搜索区域计算如下:

$$S_a^i = \alpha B_a^{i-1}. \tag{1}$$

其中: S_a^i 表示当前帧的目标搜索区域面积, α 表示搜索尺度因子, B_a^{i-1} 表示上一帧估计得到的目标边界框面积.

为得到有快速运动特性目标的搜索区域,根据目标在相邻帧间的移动距离很小这一事实,将相邻帧目标位置间距视为目标运动速度,根据当前已确定目标位置的图像帧间目标速度重新确定搜索尺度因子,进而确定下一帧的目标搜索区域.

首先,根据式(1)得到搜索区域宽和高为 $\sqrt{S_a^i}$,记当前帧目标位置坐标为 (r_i, c_i),计算该位置相对搜索区域中心 $\left(\frac{\sqrt{S_a^i}}{2}, \frac{\sqrt{S_a^i}}{2}\right)$ 的距离 D,即

$$D = \sqrt{\left(r_i - \frac{\sqrt{S_a^i}}{2}\right)^2 + \left(c_i - \frac{\sqrt{S_a^i}}{2}\right)^2}. \tag{2}$$

其次, 计算当前帧目标位置 (r_i, c_i) 与上一帧目标位置 (r_{i-1}, c_{i-1}) 的间距, 即本文认为的速度

$$v = \sqrt{(r_i - r_{i-1})^2 + (c_i - c_{i-1})^2}. \tag{3}$$

为增强算法针对不同运动特性视频序列的普适性,以及避免过多的计算量,将包含当前帧在内的已检测4帧图像的目标位置进行邻近组合: (r_{i-3}, c_{i-3}) 与 (r_{i-2}, c_{i-2})、(r_{i-2}, c_{i-2}) 与 (r_{i-1}, c_{i-1})、(r_{i-1}, c_{i-1}) 与 (r_i, c_i),并依次代入式(3)得到3种运动速度 v_1、v_2、v_3,对它们取平均可得

$$v_{av} = \frac{v_1 + v_2 + v_3}{3}. \tag{4}$$

最后,设置带有 ε 标志的相关属性阈值 ε^v、$\varepsilon^{v_{av}}$、ε^D;设置重置搜索尺度因子的条件为: 1) $v \geqslant \varepsilon^v, \widetilde{v}_{av} \geqslant \varepsilon^{v_{av}}$; 2) $D \geqslant \varepsilon^D$. 若上述条件1)、2)的其中一条得到满足,则令 $\alpha = \alpha^*$. 由上述判断条件可知:若目标运动速度较快,则应调高 α^* 值,便于生成的搜索区域更好地捕捉目标,不至于导致目标在搜索区域边角处,有碍后续有效特征提取和精确定位;若目标判为快速运动目标,则由式(1)得到下一帧的目标搜索区域 $S_a^{i+1} = \alpha^* B_a^i$, B_a^i 表示当前帧的目标边界框面积.

1.1.2 特征的选取

视觉跟踪需要从低到高、从细到粗的分辨率的多种信息数据,为了获得较好的目标表征,本文采用ResNet50作为特征提取网络,由文献[11]易知,该网

络主要由block 1～block 5组成,本文在后续表述中分别用conv 1～conv 5表示block 1～block 5的卷积输出. 为找到合理的深、浅特征组合表示,给出如图2所示的某一图像的conv 1～conv 5的可视化图.

(a) 原图像　(b)　(c)　(d)　(e)　(f)

图 2　ResNet50各block输出可视化

由图2可知: ResNet50每层提取的目标特征各不相同,随着层数的加深,卷积特征分辨率逐渐降低,抽象度逐渐提高; 由图2(b)能够清晰地看到目标(人)、背景的空间位置,几乎不含目标的语义信息,与原图像相比,除了变为灰度图外,无其他明显不同,且图2(b)包含的背景干扰信息太多,会对有效目标表征造成严重影响; 图2(c)包含目标的轮廓信息和微量的语义信息,但背景信息(白色区域)占比仍比较大; 图2(d)开始将目标(白色区域)与背景(黑色区域)划分开,卷积特征明显表现出语义信息,无法辨清目标轮廓细节,但能了解目标基础外形和大致的空间位置; 图2(e)较图2(d)更加抽象,包含的目标语义信息更加精细; 图2(f)包含的目标语义信息最为丰富,但它的维度太高,分辨率太低,所含目标位置信息太少,又极度抽象,目标为白色的连通区域不仅丢失了目标结构信息,还湮没了图中人与人间的位置空间信息.

综上,淘汰包含背景信息较多的conv 1、没有明显优势的conv 2和分辨率极低的conv 5,选择搜索区域的conv 3和conv 4作为目标表征用于后续帧的目标估计; 提取如图1中的新训练样本的conv 3和conv 4进入样本集,用于后续滤波模型的更新.

1.2　目标估计模块

1.2.1　特征响应自适应加权

由1.1节内容可知: 目标的浅层特征包含位置信息多,空间分辨率高,有利于目标定位; 深层特征能捕捉高辨识度的语义信息,对目标外观变化具有不变性,有利于滤波模型学习. 可见深、浅特征在目标被跟踪过程中所承担的责任是不同的. 文献[12]指出,简单地融合目标深、浅特征用于定位和学习滤波模型虽有一定的功能互补作用,但因对深、浅特征的处理以及相关参数使用的一致性,使浅层、深层特征在目标跟踪过程中无法完全发挥自身优势,只能使用其折中后的一个效果. 对此,将目标特征响应全局最大值处称为主峰,其他局部最大值称为次峰,响应图中主峰越高,形状越尖锐,说明是目标的可能性越大,定位越准确,主峰与次峰之间的距离越大,说明滤波模型的抗干扰能力越强.

综上,本文对深、浅特征与滤波模型的响应做自适应加权处理,以便得到一个能提升算法性能的最优融合响应,具体自适应加权过程如下.

首先,分别用x_s、x_d表示当前帧目标搜索区域的浅层特征和深层特征,且令$x=\{x_s,x_d\}$. 对x_s、x_d做插值运算转换到连续域,此处以x_s为例,即

$$J_s\{x_s\}(t)=\sum_{c=1}^{C_s}\sum_{n=0}^{N_c-1}x_s^c[n]b_c\Big(t-\frac{T}{N_c}n\Big). \tag{5}$$

其中: $J_s\{x_s\}(t)$表示x_s的C_s个通道的插值结果的累加和,$t\in[0,T)$; $x_s^c[n]$表示关于变量$n\in\{0,1,\dots,N_c-1\}$的函数; N_c、$b_c\Big(t-\frac{T}{N_c}n\Big)$分别表示$x_s^c$的分辨率和插值函数. 将式(5)中的$s$换为$d$表示深层变量,可得$J_d\{x_d\}(t)$,以下将其与$J_s\{x_s\}(t)$简记为$J_d$和$J_s$,且令$J=\{J_s,J_d\}$.

其次,将J_s和J_d分别与滤波模型$f=\{f_s,f_d\}$中的f_s与f_d做卷积响应加权融合

$$\begin{aligned}S_f\{x\}=&W_sS_{f_s}\{x_s\}+W_dS_{f_d}\{x_d\}=\\&W_s(f_s\otimes J_s)+W_d(f_d\otimes J_d).\end{aligned} \tag{6}$$

其中: $\otimes$表示卷积运算符; $S_f\{x\}$表示浅层响应$S_{f_s}\{x_s\}$与深层响应$S_{f_d}\{x_d\}$的加权融合响应结果,同J_s原型$J_s\{x_s\}(t)$一样是一个关于t的连续函数,原型为$S_f\{x\}(t)$,$S_{f_s}\{x_s\}$和$S_{f_d}\{x_d\}$亦同.

由式(6)易知,$S_{f_s}\{x_s\}=(f_s\otimes J_s)$、$S_{f_d}\{x_d\}=(f_d\otimes J_d)$分别是浅层特征$x_s$、深层特征$x_d$的插值结果$J_s$、$J_d$与对应滤波模型$f_s$、$f_d$的卷积响应,$W_s\geqslant 0$、$W_d\geqslant 0$是它们对应的加权权重,且$W_s+W_d=1$.

然后,构造损失函数$L_{t^*}(W)$用于学习权重W,即

$$\begin{aligned}&L_{t^*}(W)=-\beta_{t^*}(S_f)+\mu(W_s^2+W_d^2),\\&\beta_{t^*}(S_f)=\min_t\frac{S_f(t^*)-S_f(t)}{1-\mathrm{e}^{-\frac{p}{2}|t-t^*|^2}},\end{aligned} \tag{7}$$

其中$S_f=S_f\{x\}(t)$表示融合响应. 观察$\beta_{t^*}(S_f)$表达式可知,分子预测位置t^*与其他位置t处的响应值$S_f(t^*)$、$S_f(t)$的误差由分母大小确定,而分母处于[0　1]之间,由融合响应中的预测位置t^*与非预测位置t距离的远近决定,参数p用于控制在优化过程中t趋近于t^*的速度. $(W_s^2+W_d^2)$是由参数μ控制的W的惩罚项. 为了使$S_f(t^*)$为全局最大值,为式(7)增加约束条件: $S_f(t^*)-\beta_{t^*}(S_f)(1-\mathrm{e}^{-\frac{p}{2}|t-t^*|})\geqslant S_f(t)$,此时令$\beta=\beta_{t^*}(S_f)$,引入可学习变量$\beta$,让其与权重$W$一起进行在线学习.

综上,为达到目标浅层、深层特征响应自适应加权的目的,需优化以下带有约束条件的问题:

$$\min\ L_{t^*}(\beta, W) = -\beta + \mu(W_s^2 + W_d^2).$$
$$\text{s.t. } W_s \geqslant 0,\ W_d \geqslant 0,\ W_s + W_d = 1;$$
$$S_f(t^*) - \beta(1 - \mathrm{e}^{-\frac{p}{2}|t-t^*|}) \geqslant S_f(t). \quad (8)$$

式(8)是一个含有二次型目标函数和约束条件的二次规划问题,可以直接使用Python中的CVXOPT工具包完成优化.

最后,将融合响应$S_f(x)$中最大响应值所在处作为目标位置,由该位置、当前所在尺度以及上一帧的目标边界框一起计算得到粗略的目标边界框实现目标的初始估计.

1.2.2 最大IoU预测

以1.2.1节为基础,考虑到在跟踪任务中,算法对目标缺乏先验知识,不知被跟踪的目标具体是什么,属于何种类型,且使用1.2.1节的方法得到的目标边界框缺乏尺度估计,无法有效应对目标在跟踪过程中的旋转、遮挡等问题. 本文参考文献[9]中的IoU调制-预测网络,该网络主要由IoU调制器和IoU预测器两部分组成,具体网络结构框架见图3.

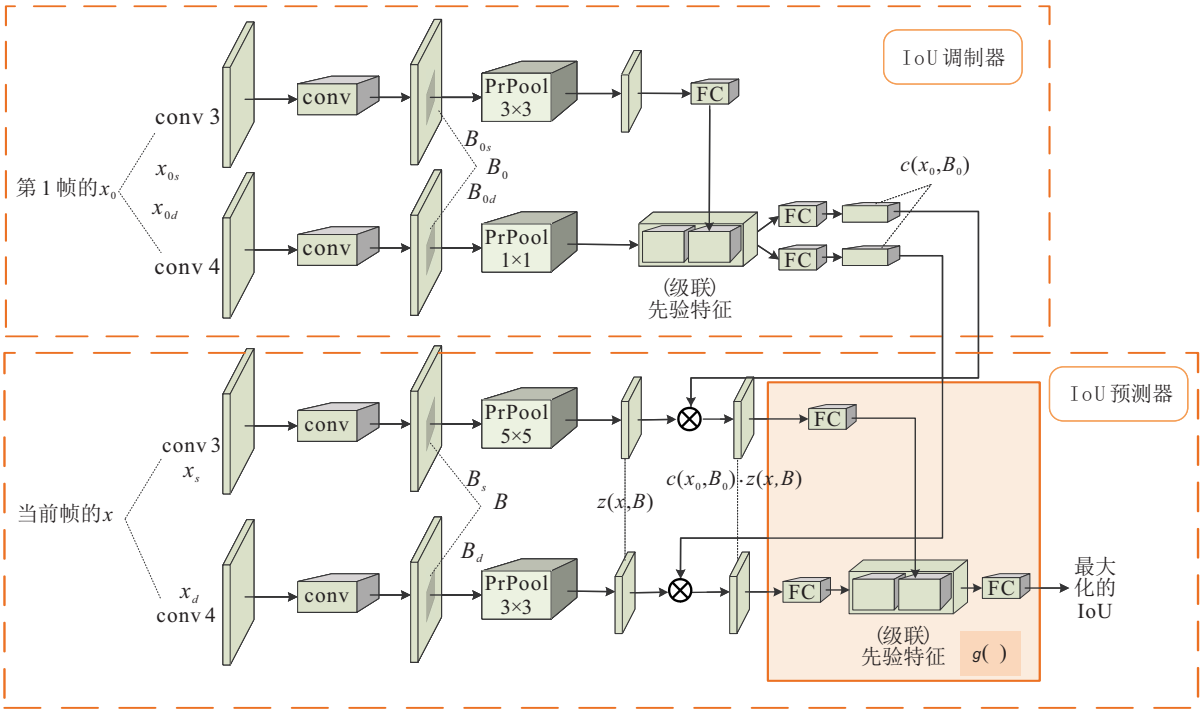

图3 调制-预测网络结构

由图3可知,IoU调制-预测网络与文献[10]中的IoUNet(单网络结构)有所不同,它是一个伪孪生网络结构,此结构便于在目标未知的情况下,通过第1帧获取准确的目标信息,并很方便利用此信息对IoU预测进行有效的调制和约束. 此外,仅使用单帧视频图像信息在线训练更新网络以应对目标在跟踪过程中的不同尺度、状态是不可靠的,所以IoU调制-预测网络是针对大量的数据样本,使用梯度上升方法进行线下训练得到的(具体训练细节见文献[9]),此策略最大限度地将目标先验信息注入了目标估计过程,可进一步提升目标边界框估计的准确性. 以下将通过IoU调制器和IoU预测器的介绍,对IoU调制-预测网络进行详细论述.

1) IoU调制器.

结合图1与图3的上半部分可知,IoU调制器仅在算法初始阶段使用1次,它以第1帧图像的搜索区域特征conv 3x_{0s}、conv 4x_{0d}(统记为x_0)和真实目标边界框B_{0s}、B_{0d}(统记为B_0)为输入,整体由2个卷积层conv、2个池化层PrPool(precise RoI pooling)[10]和3个全连接层FC组成. x_0先通过conv做特征提取,然后对B_{0s}、B_{0d}区域分别按照3×3、1×1的单位池化力度划分成多个区域,对每一区域进行PrPool,即对区域先进行插值计算,使用插值后区域内的像素和除以区域面积获得统一固定大小的局部浅层、深层目标特征. 由于PrPool中的插值操作,使B_0坐标连续可导,这为IoU调制-预测网络通过梯度上升的方法进行线下训练提供了条件. 之后,为了突出目标的几何轮廓与本质,舍弃常用的特征通道融合策略,采用特征级联的方式保持局部深、浅特征的特性,生成如图3中IoU调制器部分所示的先验特征. 然后,由FC规范统一生成大小为$1\times1\times D$(D为先验特征的通道数)的调制向量$c(x_0, B_0)$,将调制向量输入IoU预测器对当前帧特征PrPool池化的结果进行调制约束. 具体计算见下节IoU预测器内容.

2) IoU预测器.

结合图1和图3的下半部分可知,IoU预测器用于视频序列中除第1帧外的所有帧的目标状态估计. 它以当前帧图像搜索区域的conv 3x_s、conv 4x_d(统记为x)、IoU调制器生成的调制向量$c(x_0, B_0)$和1.2.1节获得的粗略目标边界框为基础,通过添加扰动一共生成的H个候选目标边界框为输入,记边界框通向为$B^h = \{B_s^h, B_d^h\}$. 其中: H为正整数,且$h \in [1, H]$. 其整体结构与IoU调制器相似,它们最大不同是IoU预测器使用5×5和3×3大单位的PrPool池化力度,削弱了背景特征信息的干扰,保留了目标的显著特征,生成了大小为$N \times N \times D$的局部浅层、深层目标特征,N是特征分辨率大小. 本文将上述获得的局部特征称为待调制项,记为$z(x, B)$. 而IoU调制器之所以使用较小的3×3和1×1的单位池化力度,是为了尽可能多地保留第1帧中准确的目标特征信息,这样产生的调制向量$c(x_0, B_0)$可靠性强,使用其对调制项$z(x, B)$进行加权调制所得的结果更加准确,调制计算为$c(x_0, B_0) \cdot z(x, B)$,可理解为使用$c(x_0, B_0)$中的$D$个参数对$z(x, B)$进行对应通道加权. 将调制结果输入到由3个FC组成的IoU预测模块$g(\ )$,可得到对应当前候选目标边界框B^h的IoU值,具体表示为

$$\mathrm{IoU}(B^h) = g(c(x_0, B_0) \cdot z(x, B^h)). \tag{9}$$

由于IoU调制-预测网络使用梯度上升方法训练而成,式(9)中的$\mathrm{IoU}(B^h)$就是对应第h个候选目标边界框的最大化IoU值. 当利用式(9)遍历当前帧的H个候选目标边界框后,对获得的IoU值中前H^*($H^* < H$,且为正整数)个较大值对应的候选边界框取平均可得预测边界框

$$B^* = \mathrm{mean}\Big(\sum_{h^*=1}^{H^*} B^{h^*}\Big). \tag{10}$$

其中: $h^* \in [1, H^*]$; B^*为最终预测的目标边界框,其中心位置为目标的新位置.

1.3　模型更新模块

由图1可知,模型更新模块主要由样本集生成与滤波模型更新两部分组成. 主要流程是在目标估计结果上根据高斯标签的宽度(与标准差有关)剪裁出目标区域作为训练样本,使该区域经ResNet50提取浅层、深层特征进入样本集用于滤波模型在线学习. 以下将对样本集的在线生成以及滤波模型的在线更新过程进行详细论述.

1.3.1　样本集的在线生成

为不影响样本集多样性以及避免样本空间的浪费,本文通过计算训练样本在空间中的位置判断其相似性,两样本位置越近相似性越大,对相似性大的样本舍弃其中学习权重较低的一方更新样本集. 设: 样本集可容纳M个训练样本,x_m是样本集中第$m(1 \leqslant m \leqslant M)$个训练样本,$\omega_m$是其对应学习权重; $d_{i1}, d_{i2}, \ldots, d_{iM}$分别表示新训练样本$x_i$与样本集中的$M$个训练样本间的距离. 具体样本集生成如下.

首先,设置一学习率$\eta \in (0, 1)$用于训练样本先验权重. 当样本集未填满时,每帧产生的训练样本逐次进入样本集直至达到M个,此过程通过以下公式计算更新训练样本权重:

$$\omega_m = \begin{cases} \omega_m/(1-\eta), & m = 1, 2, \ldots, i-1; \\ \eta, & m = i. \end{cases} \tag{11}$$

当样本集含有M个训练样本后,设置一小于m的参数K,用于计算最近K个(包含第i个最新的训练样本)进入样本集的训练样本的先验权重,在$i-K$个之前进入样本集的训练样本的先验权重统一设置为常数ξ,此时更新权重公式如下:

$$\omega_m = \begin{cases} \xi, & m = 1, 2, \ldots, i-K-1; \\ \xi(1-\eta)^{i-K-m}, & m = i-K, i-K+1, \ldots, i. \end{cases} \tag{12}$$

其中ω_m为样本集中已有训练样本的权重通项. 由$\sum\limits_m \omega_m = 1$可得$\xi = \Big(i - K + \dfrac{(1-\eta)^{-K} - 1}{\eta}\Big)^{-1}$.

然后,按照下述步骤在线更新样本集:

step 1: 训练样本数未达M时,新训练样本直接进入样本集,之后使用式(11)更新所有样本权重.

step 2: 训练样本数达到M后,实行新样本替换旧样本策略. 设一权重阈值ε^ω,每当一个新训练样本x_i进入样本集时,比较样本集中所有样本的先验权重,找到最小权重$\omega_{\min}$对应的训练样本.

step 3: 若$\omega_{\min} < \varepsilon^\omega$,则最小权重训练样本由新训练样本代替; 否则,计算比较新训练样本与样本集中所有样本的间距$d_{i1}, d_{i2}, \ldots, d_{iM}$,假设距离最近的训练样本为$x_m$,对应距离为$d_{im}$.

step 4: 比较计算d_{im}与样本集现有样本的间距$d_{12}, \ldots, d_{kq}, \ldots, d_{(M-1)M}$($k \in [1, M-1], q \in [2, M]$,且$k \neq q$). 找到最小距离设为$d_{kq}$,对应样本对为$(x_k, x_q)$. 若$d_{im} < d_{kq}$,则表明与样本集中最相似的样本对相比,新训练样本$x_i$与$x_m$更为相似,则由$x_i$替换$x_m$进入样本集; 否则,表明$x_k$与$x_q$更相似,由$x_i$替换$(x_k, x_q)$中权重较小的一方进入样本集.

step 5: 新训练样本按照step 2～step 4进入样本

集后,使用式(12)重新计算样本集中所有样本的先验权重,从而实现样本集的更新.

1.3.2 滤波模型的在线更新

目标在跟踪过程中是一直变化着的,在线更新滤波模型可以使算法适应实际运动情况更加稳定.考虑计算量问题,采用稀疏策略每隔N_s帧图像更新一次滤波模型,具体实现更新内容如下.

首先,由式(6)得到训练样本$x_m=\{x_{ms},x_{md}\}$的融合响应$S_f(x_m)$,其中x_{ms}、x_{md}分别有C_s、D_s个特征通道.

$$S_f(x_m)=W_s\sum_{c_s=1}^{C_s}f_s^{c_s}\otimes J_s^{c_s}\{x_{ms}^{c_s}\}+W_d\sum_{c_d=1}^{C_d}f_d^{c_d}\otimes J_d^{c_d}\{x_{md}^{c_d}\}. \tag{13}$$

其中:$f_s^{c_s}\otimes J_s^{c_s}\{x_{ms}^{c_s}\}$为$x_{ms}$的$c_s$通道的插值结果与其对应滤波模型$f_s$的$c_s$通道做卷积运算;$W_s$、$W_d$为浅层、深层响应权重,其具体解释见式(6);$f_d^{c_d}\otimes J_d^{c_d}\{x_{md}^{c_d}\}$解释同上.

然后,构建目标损失函数

$$E(f)=\sum_{m=1}^{M}\omega_m\|S_f(x_m)-y_m\|_{L^2}^2+\sum_{j_1=1}^{C_s}\|\lambda f_s^{j_1}\|_{L^2}^2+\sum_{j_2=1}^{C_d}\|\lambda f_d^{j_2}\|_{L^2}^2. \tag{14}$$

其中:$f=\{f_s,f_d\}$,$f_s=(f_s^1,f_s^2,\dots,f_s^{C_s})$,$f_d=(f_d^1,f_d^2,\dots,f_d^{C_d})$;$M$为训练样本个数;$\omega_m$为第$m$个训练样本的权重;$\|S_f(x_m)-y_m\|_{L^2}^2$为融合响应与高斯标签$y_m$误差的$L^2$范数;$\lambda$为$f_s$、$f_d$的惩罚项.

最后,文献[13]提到优化此类问题,同时使用高斯牛顿算法与共轭梯度算法可解决在线学习收敛速度慢的问题.据此,令f残差$r_m(f)=\sqrt{\omega_m}(S_f(x_m)-y_m)$,$r_{m(c_s+c_d)}(f)=\lambda f^j(m\in[1,M],j\in[1,C_s+C_d])$,则式(14)可写为$E(f)=\|r(f)\|_{L^2}^2$.对$r(f+\Delta f)$进行一阶泰勒展开,即$r(f+\Delta f)\approx r_f+J_f\Delta f$,可得$E(f)$的二阶高斯牛顿近似如下:

$$\tilde{E}(f)\approx E(f+\Delta f)=\Delta f^{\mathrm{T}}J_f^{\mathrm{T}}J_f\Delta f+2\Delta f^{\mathrm{T}}J_f^{\mathrm{T}}r_f+r_f^{\mathrm{T}}r_f. \tag{15}$$

其中:$r_f=r(f)$,$J_f=\dfrac{\partial r}{\partial f}$是在当前$f$下的雅可比式;$\Delta f$为$f$的增量;T为转置.用共轭梯度算法优化式(15)可得Δf,计算$f=f+\Delta f$可更新滤波模型.

2 算法步骤

IT-AWCR跟踪算法具体过程如下.

step 1:初始化算法参数:初始化搜索尺度因子α、响应加权权重W_s、W_d和滤波模型f;设置训练样本数M、模型更新频率N_s等参数.

step 2:读取视频序列第1帧图像,使用其提供的目标信息裁剪出训练样本,使用ResNet50提取目标浅层、深层特征$x=\{x_s,x_d\}$初始化样本集,利用式(5)对x做插值处理,通过式(6)得到深、浅响应加权融合得分$S_f(x)$.使用高斯牛顿和共轭梯度算法优化式(14)初始更新滤波模型.

step 3:读取下一帧图像,利用式(1)获得搜索区域,对该区域进行与step 2一致的特征提取、插值以及特征响应加权融合操作以得到$S_f(x)$,找到其中最大响应值所在位置和尺度,从而获得粗略目标边界框,然后添加扰动生成H个候选边界框.

step 4:将第1帧目标信息对应的特征x_0和人工标定的边界框B_0、当前帧特征x和由step 3生成的候选目标框B^h输入IoU调制-预测网络,通过式(9)、(10)得到最终预测目标边界框,取此边界框中心为目标新位置.记录保存当前目标信息.

step 5:使用式(2)、(3)计算距离D和速度v,若帧数允许,则使用式(4)计算v_{av}.根据1.1.1节判断条件是否重置搜索尺度因子α.若需要,则令$\alpha=\alpha^*$用于计算下帧搜索区域;否则,$\alpha=\alpha$.

step 6:根据step 4的目标估计结果剪裁出新训练样本,根据1.3.1节的样本集在线更新步骤更新样本集,包含使用式(11)或(12)更新样本集中所有样本的先验权重.

step 7:判断是否要更新滤波模型.如果需要,则先优化式(14)获得增量Δf,然后计算$f=f+\Delta f$更新滤波模型.

step 8:判断是否跟踪完视频序列,如果没有,则跳转至step 3;若跟踪完毕,则输出视频所有帧的目标信息保存结果.

3 实验结果与分析

3.1 实验平台

本文所做实验均在一台装有1张Nvidia GTX1080tiGPU、处理器为Intelcore(TM)i7-8700 K、主频为3.70 GHz、内存为32 GB的计算机上进行.操作系统为64位Ubuntu16.04,编程环境为python3.7,深度学习框架为PyTorch.

3.2 实验参数设置

由于本文算法在OTB2015和VOT2018两个数据集上进行测试评估,经大量实验结果发现某些参数应根据不同数据集设置相应的数值,以便使算法性能获得相对较优的效果.具体参数值设置见表1.此外,对于目标搜索区域和每帧产生的新训练样本用ResNet50提取的conv 3、conv 4作为目标特征分别用

于目标估计和滤波模型学习;设置初始化搜索尺度因子$\alpha = 4.5$;深、浅层响应自适应初始融合权重$W_s = W_d = 0.5$. 为公平对比,此处未提及的其他需要初始化的参数均沿用文献[9]中的设置.

表1　超参数值

数据集	α^*	K	η	N_s	ε^v	ε^D	$\varepsilon^{v_{av}}$	μ	p	ε^ω	M
OTB2015	5.5	80	0.009	6	0.25	0.35	0.225	2×10^{-6}	0.15	0.003 6	200
VOT2018	5	100	0.007 5	5							

3.3　参数选取对比分析

本节用到的性能评价指标有:OTB2015数据集中的success,即预测目标框与真实目标框交并比大于阈值0.6的视频帧的百分比;精确度precision是预测位置到真实目标位置阈值小于20个像素点的视频帧的百分比. VOT2018数据集中的鲁棒性robustness,其值越小,表示算法越稳定.

对于本文1.1.1节中ε^v、$\varepsilon^{v_{av}}$、ε^D阈值的确定,其中ε^v值是根据参考文献[14]中计算两个特征在同一空间的位置距离阈值0.25设置的. 为增加判断目标快速运动的准确性,针对ε^v增加了一个平均速度阈值$\varepsilon^{v_{av}}$. 此外,还设置了根据单帧判断目标是否快速运动的距离阈值ε^D. 为了体现参数$\varepsilon^{v_{av}}$和ε^D对算法性能的影响,以ε^v值0.25为参考,在其周围以0.025的步长对$\varepsilon^{v_{av}}$和ε^D不断取值测试,得到表2的实验结果.

表2　不同$\varepsilon^{v_{av}}$和ε^D值测试结果

$\varepsilon^{v_{av}}(\varepsilon^v=0.25,\varepsilon^D=0.35)$	OTB2015 FM属性 success
0.200	0.696
0.225	0.702
0.250	0.699
0.275	0.689
0.300	0.675
$\varepsilon^D(\varepsilon^v=0.25,\varepsilon^{v_{av}}=0.225)$	OTB2015 FM属性 success
0.200	0.674
0.225	0.688
0.250	0.690
0.275	0.691
0.300	0.698
0.325	0.701
0.350	0.702
0.375	0.701

表2数据的评价指标是OTB2015中的快速移动FM属性的成功率success. 由表2实验数据可知,只有$\varepsilon^{v_{av}} = 0.225$时,快速移动属性对应的跟踪成功率最大,但是,除取0.275、0.300时成功率较最优值差距比较大外,其他取值效果相差不大,与最优值最高相差0.6%. 从0.200到0.225成功率提升了,说明目标速度变快,调高搜索尺度因子做法是正确的;此外,当$\varepsilon^{v_{av}} = 0.300$时,阈值过大,此时再调高搜索尺度因子已经错过了最佳时机,可能在$\varepsilon^{v_{av}} = 0.250$等其他值时,因搜索尺度因子保持不变,所得搜索区域已经出现目标包含不全的现象,加上不断跟踪帧间的叠加效果,极大可能出现跟丢目标的情况,造成成功率明显下降;另外,随着ε^D取值从0.200到0.350,success逐渐递增,说明目标速度比较快时,调高搜索尺度因子是有效的. 但$\varepsilon^D = 0.200$时,success最低,可能是当时目标速度还不“足够”快,调高尺度因子引入了过多背景信息,降低了目标在搜索区域的占比,影响了跟踪结果. 当$\varepsilon^D = 0.325$和0.375时,success均比最优值仅低了0.1%,可以把[0.325, 0.375]当作取值的一个饱和区域,测试结果为$\varepsilon^D = 0.350$时效果最佳,则取0.350作为最终算法的ε^D值.

表3是根据文献[9]给出的以α的先验值4.5为基准,以0.5为间隔形成的不同搜索尺度因子分别在OTB2015和VOT2018上测得的成功率success和鲁棒性robustness.

表3　不同搜索尺度因子在OTB2015上的测试结果

搜索尺度因子α^*	OTB2015 success	VOT2018 robustness
4.0	0.660	0.208
4.5	0.676	0.201
5.0	0.688	0.173
5.5	0.698	0.206
6.0	0.674	0.212

由表3中针对OTB2015的success数据可知,当$\alpha^* = 5.5$时,成功率最高为0.698,比参考的先验值4.5性能提升了2.2%. 当α^*取值从4.0到5.5时,算法的成功率呈不断上升趋势;从5.5到6.0时,性能表现下降. 观察VOT2018的robustness数据可知,当$\alpha^* = 5.0$时,算法鲁棒性最好,比α^*为4.5和5.5时性能分别提升了2.8%和3.3%,其他取值获得的robustness差距都在1%以内. 以上实验结果可以说明当目标运动情况符合1.1.1节的搜索尺度因子重置条件时,目标运动速度相对较快,调高α^*值扩大目标搜索区域是正确的,但过度调高会使算法遭受较强干扰,出现性能下降的情况. 所以,设置合理的α^*能使搜索区域既不会丢失目标,也不会增加过多背景,这样更有利于后续的目标估计和滤波模型学习.

为了验证1.1.2节分析采用conv 3、conv 4作为目标的浅层和深层特征的有效性,首先,根据对深度卷积网络的先验认知将conv 2和conv 3划分为浅层特征,conv 4和conv 5划分为深层特征,得到如表4所示的不同特征组合在OTB2015上的测试精确度和成功率结果. 由于conv 1包含太多背景空间信息,且几乎不含目标语义信息,一旦出现目标丢失的情况,其所示特征对目标准确定位干扰巨大,此处conv 1不参与测试分析. 又考虑到算法的计算量问题,深、浅特征各取一种.

表 4 不同特征组合在OTB2015上的测试结果

特征组合	precision	success
conv 2+conv 4	0.891	0.667
conv 2+conv 5	0.898	0.688
conv 3+conv 4	0.920	0.698
conv 3+conv 5	0.926	0.701

由表4中conv 2+conv 4和conv 3+conv 4的数据对比可知:后者较前者分别提升了2.9%和3.1%的精度和成功率,表明了conv 3较conv 2的辨识度以及应对目标表观变化的能力更强,这也对应了conv 3在1.1.2节的可视化展示,拥有较好的目标空间位置信息以及包含部分语义信息,有利于目标估计和滤波模型的学习,所以,采用conv 3作为目标的浅层特征. 以conv 3作浅层特征为基础,对比表4中conv 3+conv 4和conv 3+conv 5两组数据可知,后者仅比前者提高了0.6%和0.3%的精度和成功率,这是因为conv 5包含极为丰富的目标语义信息,对目标尺度、形变等具有不变性,可以很好地表征目标本质,进而较conv 4提升了算法性能. 但因conv 5过度抽象、分辨率极低,导致性能提升不大,又conv 5的通道数是conv 4的2倍,考虑后续的卷积计算量,最终采用conv 4作为目标的深层特征.

图4是某帧图像搜索区域的浅层、深层特征响应(使用的是未做任何加权处理学习得到的滤波模型)以不同加权方式得到的融合响应展示.

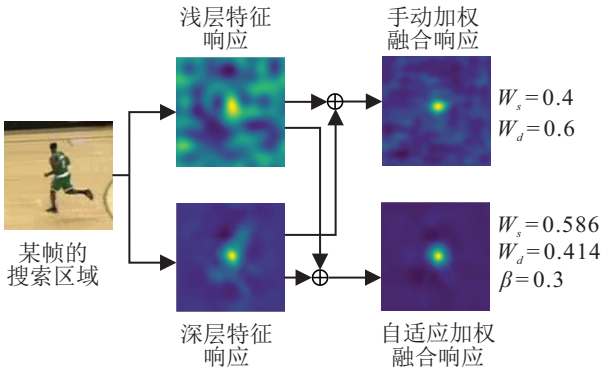

图 4 特征响应融合方式对比

从图4中的浅层特征响应可以看出,除目标响应所在的主峰外,还有很多背景产生的次峰,且次峰峰值与主峰差距不明显,表明滤波模型中和浅层特征进行卷积的部分对目标的判别力度较小. 深层特征响应中次峰较少,但获得的主峰偏向于大标准差的高斯形状,不满足形状越尖定位越准确的要求. 手动加权融合响应是在浅层响应权重W_s和深层响应权重W_d分别为0.4和0.6时获得的,其次峰峰值与主峰峰值相差较远,增加了主峰峰值处(即目标实际位置)的概率,但其主峰(目标)附近存在密集的次峰,说明其滤波模型的稳定性有待进一步提高. 自适应加权融合响应是在得到自适应值0.586、0.414、0.3后通过式(6)得到的,由图4可知,中间明亮部分较其他响应范围更小、更亮,说明:主峰形状高而尖则目标定位精度较高;离主峰较远处才有微弱的干扰响应则此滤波模型抗干扰能力强.以上结果分析可以验证本文对特征响应做自适应加权策略的有效性.

3.4 跟踪算法性能对比分析

为了验证本文提出的IT-AWCR跟踪算法的有效性,将其与近几年一些主流跟踪算法一起分别在OTB2015、VOT2018数据集上进行对比测试,涉及的相关性能指标AUC是某跟踪算法生成的成功率曲线与横、纵坐标轴之间的面积;帧率表示每秒算法跟踪的帧数;EAO是所有视频帧预测的边界框与其对应的人工标定的边界框之间的交并比的平均取值;accuracy精确度数值越大,算法准确度越高;robustness、success含义同3.3节.

表5是本文算法与基于相关滤波跟踪算法ECO、UPDT(unveiling the power of deep tracking)、基于孪生网络的跟踪算法SPM、MLT(deep meta learning for real-time target-aware visual tracking)、SiamRPN、DaSiamRPN(distractor-aware siamese networks for visual object tracking)、SiamRPN++和依靠其他方式获取高性能的跟踪算法MDNet、ATOM在OTB2015数据集上的AUC和帧率测试对比结果.

表 5 OTB2015测试对比结果

跟踪算法	AUC	帧率(FPS)
ECO[1]	0.687	7
UPDT[12]	0.689	0.4
SPM[15]	0.687	120
MLT[16]	0.611	48
SiamRPN[6]	0.643	71
DaSiamRPN[17]	0.665	160
SiamRPN++[7]	0.696	35
MDNet[18]	0.678	1
ATOM[9]	0.671	30
本文算法	0.698	8

由表5中AUC数据可知: 本文算法的AUC较相关滤波中最优算法UPDT高出了0.9%, 较基于孪生网络的最优算法SiamRPN++提高了0.2%, 较基础算法ATOM提高了2.7%. 这是因为与相关滤波算法相比, 本文通过使用IoU调制-预测网络进一步精确了目标边界框的预测; 与基于孪生网络的算法相比, 本文通过在线更新机制使滤波模型尽可能地适应目标和背景变化; 与其他算法相比, 本文特征响应自适应加权以及样本集的生成和权重更新策略, 为稳定跟踪提供了保障, 提高了成功率, 增大了AUC值. 此外, 本文较SiamRPN等孪生网络算法添加了样本集和滤波模型的在线更新, 并使用梯度下降法实现训练更新, 导致帧率很低, 但整体跟踪效果AUC相对较优; 与AUC靠前的基于相关滤波框架的ECO和UPDT相比, 本文算法每秒分别多跟踪1帧和7.6帧. 这得益于本文采取conv 4表示目标深层特征, 以及相似样本替换、稀疏更新滤波模型等策略.

表6是本文算法与UPDT、DaSiamRPN、LADCF、ATOM、 SiamR-CNN算法在VOT2018数据集上的EAO、accuracy、robustness的测试对比结果.

表 6　VOT2018测试对比结果

跟踪算法	EAO	accuracy	robustness
UPDT[12]	0.378	0.536	0.184
DaSiamRPN[17]	0.383	0.586	0.276
LADCF[3]	0.389	0.503	0.159
ATOM[9]	0.401	0.590	0.204
SiamR-CNN[19]	0.408	0.609	0.220
本文算法	0.424	0.592	0.173

由表6可知, 本文算法以0.424的EAO成绩排名第1, 这得益于特征响应自适应加权策略和类似孪生网络的IoU调制-预测网络的使用, 根据目标实际状况在线更新深、浅响应权重, 使目标深、浅特征在目标定位时以响应的形式充分发挥自身优势. 之后, 将第1帧的目标信息作用于后续帧的目标估计, 有效应对了目标的尺度变化, 进而提升了EAO. 此外, 本文以0.592的成绩accuracy指标排名第2, 仅次于SiamR-CNN的1.7%, 比基础算法ATOM提升了0.2%; robustness仅次于LADCF算法, 比EAO性能排名靠前的ATOM、SiamR-CNN分别降低了3.1%、4.7%. 以上结论说明本文算法可以获得有效、强稳定性的滤波模型, 也可进一步说明采取的特征响应自适应加权等策略的有效性.

图5是本文算法与基础算法ATOM在VOT2018数据集的basketball、iceskater 2、soccer 2、soldier序列中的部分帧的实际跟踪结果. 算法ATOM用细线表示, 本文算法用粗线表示.

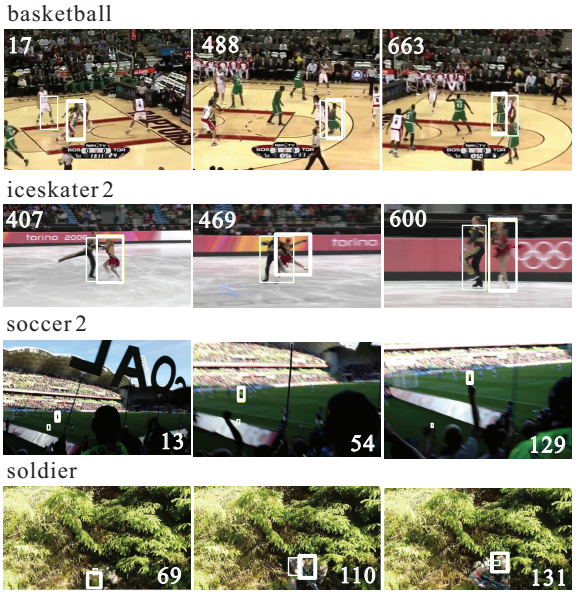

图 5　VOT2018部分序列跟踪结果对比

在basketball序列的第17帧中, 跟踪目标被相似伪目标遮挡了将近90%, ATOM细框跟丢了目标, 而本文算法粗框能实现有效跟踪. 在basketball的488帧和663帧中, 本文算法预测的目标边界框较ATOM更加贴合目标大小, 这样不会导致下一帧的目标搜索区域过大, 增加不必要的特征提取时间和伪目标的干扰力度, 并且488帧涉及目标的快速移动, 本文跟踪结果也得益于确定的合理的目标搜索区域; 从iceskater 2序列的407帧到469帧再到600帧, ATOM的跟踪结果逐渐偏向于跟踪对象旁边的干扰(男士), 随着目标(女方)的快速移动、大的非刚性形变和运动模糊, 最终在600帧中丢失了目标. 可知本文算法相对ATOM具有较好的应对快速移动、非刚性形变、运动模糊等跟踪挑战的能力; soccer 2序列的跟踪对象是一个小目标, ATOM在3帧中均跟踪失败. 第13帧拍摄镜头较远, 目标太小不易捕捉; 第54帧镜头拉近, 目标尺度发生一定变化, 相机抖动模糊了目标; 第129帧存在部分遮挡. 本文跟踪结果良好可以说明本文方法具有较好的应对目标尺度变化的能力, 并验证了特征选取策略、特征响应自适应加权协同IoU调制-预测网络进行目标估计策略的有效性和学习的滤波模型的强稳定性. soldier序列主要涉及跟踪过程中光照、不同程度的遮挡以及相似物干扰等挑战. 在所展示的3帧跟踪结果中, ATOM均未锁定目标(士兵头部), 且跟踪目标框忽大忽小, 说明其在线学习的卷积核在soldier上的稳定性较差. 本文算法在110帧也跟丢了目标, 但与ATOM的结果相比, 本文结果更靠近实际目标, 这有利于在后续帧中重新找回目标. 在131帧中, 本文算法虽未能实现很好的跟踪效果, 但比ATOM预测的目标框包含的目标部分多了很多.

4 结 论

本文提出的IT-AWCR跟踪算法由特征提取模块、目标估计模块、模型更新模块3部分组成. 通过特征可视化图确定使用ResNet50输出的conv 3、conv 4分别作为目标的浅层和深层特征表示,并对特征响应设计自适应加权处理策略,从加权融合响应获取目标的大致位置和粗略边界框. 然后,为有效应对目标尺度变化,使用IoU调制-预测网络进一步精确了目标边界框. 最后,将不断更新的样本集用于以稀疏方式优化损失函数的在线更新滤波模型. 本文的特征响应自适应加权策略为使用IoU调制-预测网络预测高精度目标边界框进一步提供了保障;高准确度的目标边界框为学习到抗干扰能力强的滤波模型提供了保障;而强抗干扰力的滤波模型有益于特征响应的自适应加权. 所有主要策略环环相扣,致力于提升算法的精度和稳定性. 实验结果也表明了本文跟踪算法具有较好的应对目标尺度变化、非刚性形变、运动模糊、快速移动等能力,以及拥有较高的跟踪精度和较为稳健的滤波模型.

参考文献(References)

[1] Danelljan M, Bhat G, Khan F S, et al. ECO: Efficient convolution operators for tracking[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, 2017: 6931-6939.

[2] Li F, Tian C, Zuo W M, et al. Learning spatial-temporal regularized correlation filters for visual tracking[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, 2018: 4904-4913.

[3] Xu T Y, Feng Z H, Wu X J, et al. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking[C]. IEEE Transactions on Image Processing. Piscataway: IEEE, 2019: 5596-5609.

[4] Yang T, Xu P F, Hu R B, et al. ROAM: Recurrently optimizing tracking model[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, 2020: 6717-6726.

[5] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[6] Li B, Yan J J, Wu W, et al. High performance visual tracking with Siamese region proposal network[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, 2018: 8971-8980.

[7] Li B, Wu W, Wang Q, et al. SiamRPN++: Evolution of siamese visual tracking with very deep networks[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, 2019: 4277-4286.

[8] Du F, Liu P, Zhao W, et al. Correlation-guided attention for corner detection based visual tracking[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, 2020: 6835-6844.

[9] Danelljan M, Bhat G, Khan F S, et al. ATOM: Accurate tracking by overlap maximization[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, 2019: 4655-4664.

[10] Jiang B R, Luo R X, Mao J Y, et al. Acquisition of localization confidence for accurate object detection[C]. Computer Vision-ECCV 2018. Cham: Springer, 2018: 816-832.

[11] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, 2016: 770-778.

[12] Bhat G, Johnander J, Danelljan M, et al. Unveiling the power of deep tracking[M]. Computer Vision-ECCV 2018. Cham: Springer, 2018: 493-509.

[13] Chen Z W, Wang Y, Song J, et al. The application of LTRNet convolution features in the improvement of ECO[J]. Control Theory & Applications, 2020, 37(12): 2601-2610.

[14] Liu Q Y. Research on adaptive visual target tracking method based on correlation filter[D]. Changchun: Northeast Normal University, 2019: 50-51.

[15] Wang G T, Luo C, Xiong Z W, et al. SPM-tracker: Series-parallel matching for real-time visual object tracking[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, 2019: 3638-3647.

[16] Choi J, Kwon J, Lee K M. Deep meta learning for real-time target-aware visual tracking[C]. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, 2019: 911-920.

[17] Zhu Z, Wang Q, Li B, et al. Distractor-aware siamese networks for visual object tracking[M]. Computer Vision-ECCV 2018. Cham: Springer, 2018: 103-119.

[18] Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, 2016: 4293-4302.

[19] Voigtlaender P, Luiten J, Torr P H S, et al. Siam R-CNN: Visual tracking by re-detection[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, 2020: 6577-6587.

作者简介

陈志旺(1978–), 男, 副教授, 博士, 从事运动物体目标检测与跟踪、多旋翼飞行器控制等研究, E-mail:czwaaron@ysu.edu.cn;

王莹(1995–), 女, 硕士生, 从事单目标跟踪的研究, E-mail:1718819591@qq.com;

宋娟(1978–), 女, 工程师, 从事无人机电力系统巡线的研究, E-mail: 1138812341@qq.com;

刁华康(1996–), 男, 硕士生, 从事目标跟踪的研究, E-mail: 1319586335@qq.com;

彭勇(1963–), 男, 教授, 博士生导师, 从事生物调控、脑电应用科学等研究, E-mail: PY81@sina.com.

(责任编辑: 闫 妍)