中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION

基于并行多外观特征的孪生网络目标跟踪算法

陈志旺, 孙泽兵, 吕昌昊, 曹索航, 彭勇

引用本文:
陈志旺,孙泽兵,吕昌昊,曹索航,彭勇. 基于并行多外观特征的孪生网络目标跟踪算法[J]. 控制与决策, 2024, 39(11): 3628–3636.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2023.0851

基于并行多外观特征的孪生网络目标跟踪算法

陈志旺[1,2†], 孙泽兵[1,2], 吕昌昊[3], 曹索航[1,2], 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. 燕山大学 电力电子节能与传动控制河北省重点实验室，河北 秦皇岛 066004；4. 燕山大学 电气工程学院，河北 秦皇岛 066004)

摘 要: 目标跟踪通常只能使用视频第1帧的外观信息,在线学习目标的外观特征,并预测后续帧中该目标的位置和大小. 然而,跟踪过程中目标外观时刻变化,仅通过第1帧并不能准确描述后续目标的外观. 针对上述问题,提出一种基于并行多外观特征的孪生网络目标跟踪算法. 首先,引入包含目标近期外观信息的动态模板帧,同时提出3种方法: 多外观特征、并行外观特征、并行多外观特征,利用动态模板帧进行目标跟踪. 与简单地使用动态模板帧替换初始模板帧不同,所提出的方法可解决由动态模板帧中目标容易漂移导致的跟踪算法性能下降的问题. 其次引入评价模块,使用基于信息熵的评价方法或基于IOU-Net的评价方法,对得到的多个预测结果分别进行打分,选择得分最高的预测结果作为最终的预测结果. 最后提出更新模块,对评价模块得到的得分进行分析,当得分满足更新模块设立的更新条件时,用最终的预测结果更新动态模板帧,使用新的外观信息指导下一帧跟踪. 实验结果显示,该算法在GOT-10k、OTB100等标准数据集上取得较好效果,验证了所提算法的有效性.

关键词: 目标跟踪；孪生网路；外观特征；信息熵

中图分类号: TP391.4 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2023.0851

引用格式: 陈志旺, 孙泽兵, 吕昌昊, 等. 基于并行多外观特征的孪生网络目标跟踪算法 [J]. 控制与决策, 2024, 39(11): 3628-3636.

Tracking algorithm of Siamese network based on parallel multiple appearance features

CHEN Zhi-wang[1,2†], *SUN Ze-bing*[1,2], *LV Chang-hao*[3], *CAO Suo-hang*[1,2], *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. Key Lab of Power Electronics for Energy Conservation and Motor Drive of Hebei Province，Yanshan University，Qinhuangdao 066004，China；4. School of Electrical Engineering，Yanshan University，Qinhuangdao 066004，China)

Abstract: Target tracking usually only uses the appearance information of the first frame of the video to obtain the appearance characteristics of the target online, and predict the position and size of the target in subsequent frames. However, the appearance of the target changes all the time during the tracking process, and the appearance of subsequent targets is not be accurately described by the first frame alone. Focusing on the above problems, this paper proposes a Siamese network target tracking algorithm based on parallel multi-appearance features. First, dynamic template frames containing information about the recent appearance of the target are introduced. At the same time, three methods of multi-appearance, parallel appearance and parallel multi-appearance are proposed, which make dynamic template frames for target tracking. Second, either the evaluation strategy of information entropy or the evaluation method of the neural network in the evaluation module is applied to score the obtained multiple predictions separately, and the prediction result with the highest score is selected as the final prediction result. Finally, an update module is proposed to analyse the scores obtained from the evaluation module. If the score meets the update conditions set by the update module, the final prediction result is used to update the dynamic template frame, and the new appearance information is used to guide the tracking of the next frame. The experimental results show that the algorithm achieves good results on standard datasets such as GOT-10k, OTB100, which verif the effectiveness of the proposed algorithm.

Keywords: object tracking；Siamese network；appearance features；information entropy

收稿日期: 2023-06-16；**录用日期:** 2023-10-09.

基金项目: 国家自然科学基金项目(61573305)；河北省自然科学基金项目(F2022203038, F2019203511).

责任编委: 李少远.

†通讯作者. E-mail: czwaaron@ysu.edu.cn.

0　引　言

目标跟踪需要基于最少的监督(通常是视频的第 1 帧),在线学习目标的外观特征,连续定位视频中运动的目标,它有两个重要任务: 1) 需要识别出外观变化较大的目标,实现跟踪这一任务; 2) 需要过滤掉背景中与目标非常相似的干扰因素,防止目标框在目标与相似物之间漂移,实现跟踪好这一任务.

对于任务 1),一项具有里程碑意义的工作 SiamFC (fully-convolutional Siamese networks)[1] 将目标跟踪问题描述为相似性学习问题,解决了跟踪外观变化较大的目标这一难题. SiamRPN (Siamese region proposal network)[2] 在 SiamFC 的基础上,引入了区域候选网络 (region proposal network, RPN)[3],进一步提高了跟踪精度. 但 SiamFC 和 SiamRPN 只能使用诸如 AlexNet (Alex's convolutional neural networks)[4] 等较浅的卷积神经网络提取特征,限制了深度神经网络在目标跟踪领域的应用. 当背景驳杂、目标由于快速运动变得模糊或背景中存在相似物时, SiamFC 和 SiamRPN 往往会失效[5-6].

对于任务 2),为了区分外观变化较大的目标和与目标非常相似的干扰物,通常从两个方面努力:

1) 通过强大的特征提取网络学习更具表现力的特征,从而区分目标和相似干扰物[7]. SiamRPN++ (Siamese region proposal network plus plus)[8] 采用空间感知采样策略,使目标跟踪可以用 ResNet[9] 等深网络作为特征提取网络,提高了跟踪算法的精度. 接下来的许多工作都是在 SiamRPN++ 的研究成果上开展的,如 SiamFC++ (Siamese fully convolutional networks plus plus)[10]、Ocean (object-aware anchor-free tracking)[11] 等,都是借鉴了 SiamRPN++ 的空间感知采样策略,使用 GoogleNet[12] 等深层网络提取更具表现力的特征,这类算法还指出了引入区域候选网络 (RPN) 来提高跟踪精度的一些缺点,提出了无预定锚框 (anchor-free) 的方法对预测结果进行回归,进一步提高了跟踪精度. SBT (single branch transformer)[13] 基于 Transformer[14] 为目标跟踪任务设计了一个特征提取网络,并将其应用到现有的多个跟踪算法上,显著提高了原跟踪算法的性能.

2) 设计更优秀的互相关模块,充分利用特征提取网络从初始帧提取的模板特征和从当前帧提取的检测特征,更有效地对比两者的异同,从而区分目标与相似干扰物. DiMP (discriminative model prediction)[15] 和 PrDiMP (prediction-refinement for discriminative model prediction)[16] 采用一种基于在线学习的互相关模块,几次迭代后可得到一个具有判别能力的模板特征; TransT (transformer tracking)[17] 和 STARK (sparse spatio-temporal transformer for visual tracking)[18] 将 Transformer 引入目标跟踪,取代原有的互相关模块,获得了巨大成功; ToMP (transforming model prediction for tracking)[19] 对特征提取网络提取的初始模板特征与初始检测特征进行融合,获得了更具判别能力的增强模板特征和更丰富的增强检测特征,有利于目标跟踪.

虽然以上工作都取得了很好的效果,但仅使用视频第 1 帧提供的少量外观信息,使目标跟踪任务存在模板信息不充分的局限. 本文通过对以上算法的研究,提出一种基于并行多外观特征的孪生网络目标跟踪算法. 主要工作为: 1) 提出并行多外观特征策略,在跟踪过程中引入不断变化的外观信息; 2) 引入评价模块和更新模块,选择最佳的跟踪结果并控制外观信息的更新; 3) 引入一个基于 Kullback-Leibler (KL) 散度的辅助训练头,避免因热力图上多处有较高概率值而造成的目标框发散.

1　算法描述

本文的整体算法框图如图 1 所示.

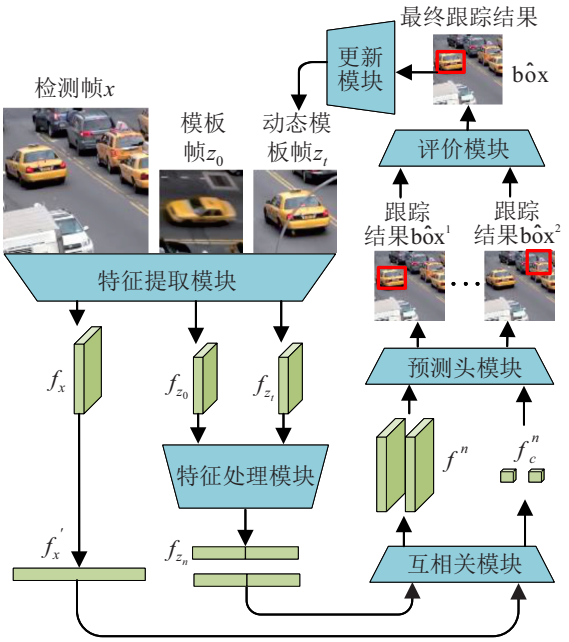


图 1　基于并行多外观特征的孪生网络目标跟踪算法

图 1 的输入有 3 张图片: 初始目标对应的模板帧 z_0、近期目标对应的动态模板帧 z_t 和当前检测帧 x. 首先将 3 张图片输入特征提取模块 (移除最后一级卷积层和全连接层的 ResNet50),得到包含初始目标外观信息的初始外观特征 f_{z_0}、近期目标外观信息的动态外观特征 f_{z_t} 和当前目标外观和位置信息的检测特征 f_x; 其次将初始外观特征 f_{z_0} 和动态外观特征

f_{z_t}输入特征处理模块,组合成外观特征更加丰富的并行多外观特征f_{z_n};然后将检测特征f_x和并行多外观特征f_{z_n}送入互相关模块进行特征匹配和融合,随后将融合后的特征输入预测头模块得到两个跟踪结果;最后将两个跟踪结果送入评价模块打分,得到最终的预测结果,并由更新模块判断最终的预测结果能否更新动态模板帧z_t,以引入最新的外观信息.

2 特征处理模块

特征处理模块的作用是:组合初始外观特征f_{z_0}和动态外观特征f_{z_t},为用动态模板帧z_t替换模板帧z_0、更新目标的外观特征提供了3种新思路:多外观特征f_z、并行外观特征f_n和并行多外观特征f_{z_n},它们的结构如图2所示.

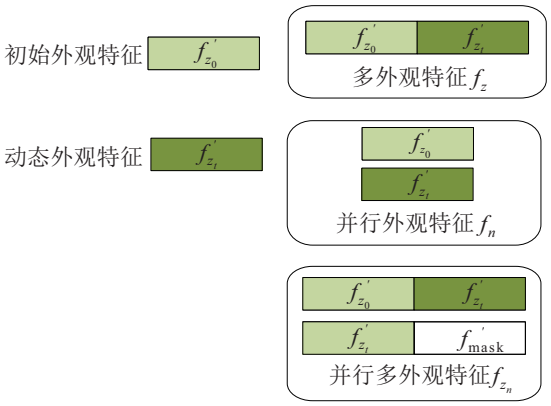


图2 f_z、f_n和f_{z_n}的结构

1) 多外观特征:动态模板帧f_{z_t}包含很重要的外观信息,但简单的用动态模板帧f_{z_t}替换模板帧f_{z_0}去定位目标会导致性能下降,本文采用多外观特征策略解决这一问题. 多外观特征f_z中既包含初始外观信息,又包含动态外观信息. 使用多外观特征f_z对检测特征f_x中的目标定位,相当于掌握了目标多个角度的外观信息后再对目标进行跟踪. 多外观特征的计算如下所示:

$$f'_{z_0} = \text{Flatten}(f_{z_0}), \tag{1}$$

$$f'_{z_t} = \text{Flatten}(f_{z_t}), \tag{2}$$

$$f_z = \text{Cat}(f'_{z_0}, f'_{z_t}). \tag{3}$$

其中:Flatten$(\cdot)$表示将三维张量$R^{c\times h\times w}$变形为二维张量$R^{c\times(hw)}$;Cat$(\cdot)$表示将两个张量拼接.

2) 并行外观特征:在多外观特征f_z的基础上,本文提出并行外观特征f_n. 使用并行外观特征f_n对检测特征f_x中的目标定位,相当于拥有目标两个角度的外观信息,每次使用一个外观信息对目标进行跟踪,跟踪两次得到两个跟踪结果,最后综合考虑这些跟踪结果,其计算如下所示:

$$f_n = \text{Stack}(f'_{z_0}, f'_{z_t}), \tag{4}$$

其中Stack$(\cdot)$表示将多个张量沿新维度拼接.

3) 并行多外观特征:综合考虑多外观特征f_z和并行外观特征f_n,本文提出并行多外观特征f_{z_n}. 使用并行多外观特征f_{z_n}对检测特征f_x中的目标定位,相当于掌握多个外观信息后对目标进行跟踪,跟踪两次得到两个跟踪结果,最后综合考虑这些跟踪结果,其计算如下所示:

$$f_{z_n} = \text{Stack}(\text{Cat}(f'_{z_0}, f'_{z_t}), \text{Cat}(f'_{z_0}, f'_{\text{mask}})). \tag{5}$$

其中:f'_{mask}表示形状与f'_{z_0}相同、元素全为0的填充特征.

3 互相关模块和预测头模块

3.1 互相关模块

在目标跟踪任务中,常见的互相关模块有:朴素互相关模块(naive correlation)、深度互相关模块(depth-wise correlation)[20]等,它们都可以用下式进行归纳:

$$f_{xz_0} = f_x * f_{z_0}. \tag{6}$$

其中:$*$表示卷积运算;f_{xz_0}表示互相关后得到的融合特征. 很明显,使用上述互相关模块计算检测特征f_x和初始外观特征f_{z_0}之间的相似性是一种线性运算. 受Transformer的启发,还可以用注意力模块(attention)做互相关模块,其公式如下:

$$\text{Att}(Q,K,V) = \text{softmax}\Big(\frac{QK^{\text{T}}}{\sqrt{d_k}}\Big)V. \tag{7}$$

因为矩阵乘法在数学上等价于卷积核为1的卷积运算,所以QK^{T}相当于一次卷积运算. 同理$\Big[\text{softmax}\Big(\frac{QK^{\text{T}}}{\sqrt{d_k}}\Big)\Big]$也相当于一次卷积运算. 故一次注意力运算相当于两次卷积运算,注意力模块做互相关模块比其他常用的互相关模块更加有效. 在目标跟踪过程中,跟踪目标的外观处于时刻变化的状态,比较两个不断变化的目标外观特征的相似性绝不是一个简单的线性运算. 由于激活函数softmax$(\cdot)$的存在,注意力运算是一个非线性运算,更适合目标跟踪任务. 为了在保证算法的简洁性的基础上,使用注意力模块做互相关模块,更好地比较不断变化的目标外观特征的相似性,本文借鉴了Transformer的结构,互相关模块的网络结构如图3所示,可用如下公式描述:

$$(f^n, f^n_c) = \text{Correlation}(\text{repeat}(f_x), f_{z_n}). \tag{8}$$

其中:Correlation$(\cdot)$表示互相关模块,repeat$(\cdot)$表示将张量沿新维度复制一次,f^n表示融合特征,f^n_c表示通道特征.

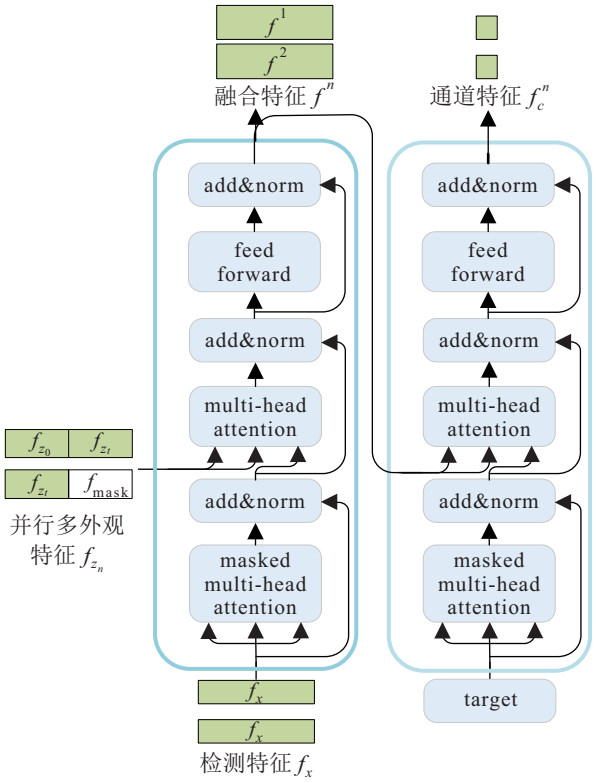


图 3　互相关模块网络结构

3.2　预测头模块

本文借鉴了 STARK 的预测头模块, 另外添加一个辅助训练头用于辅助训练, 其网络结构如图 4 所示.

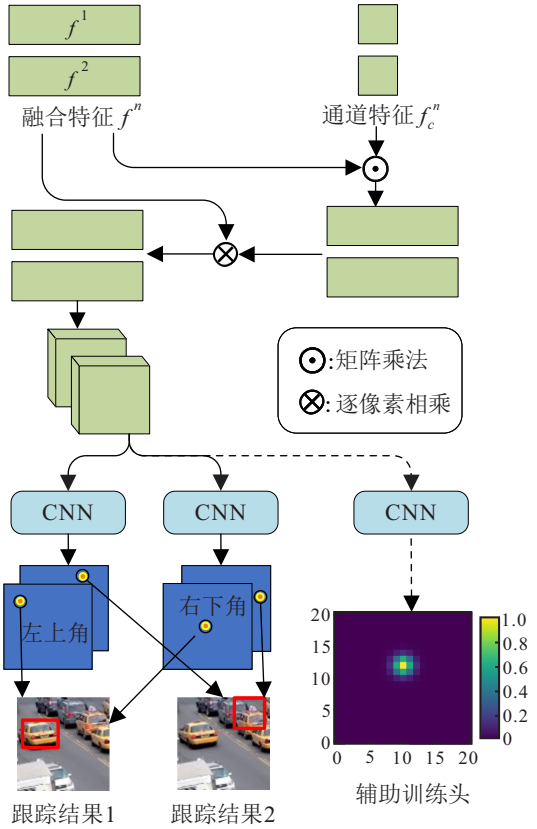


图 4　预测头模块网络结构

预测头模块可以描述为

$$\hat{\text{box}}^n = (\hat{x}_{tl}^n, \hat{y}_{tl}^n, \hat{x}_{br}^n, \hat{y}_{br}^n) = \text{head}(f^n, f_c^n). \quad (9)$$

其中: $\hat{\text{box}}^n = (\hat{x}_{tl}^n, \hat{y}_{tl}^n, \hat{x}_{br}^n, \hat{y}_{br}^n)$ 表示第 n 个预测结果; $\text{head}(\cdot)$ 表示预测头模块.

首先使用矩阵乘法计算融合特征 f^n 与通道特征 f_c^n 之间的相似性, 得到相似性分数图; 然后将融合特征 f^n 与相似性分数图逐元素相乘, 以增强重要区域的特征表示, 削弱其他区域的特征表示; 最后将增强后的融合特征 f^n 送入 3 个并行的卷积网络 (CNN) 中, 其中前两个卷积网络分别输出跟踪目标的预测框的左上角热力图和右下角热力图, 最后一个卷积网络是辅助训练头, 仅在训练阶段发挥作用. 实际上, 左上角热力图上每个特征点的数值表示为: 目标框的左上角为该特征点的概率, 故可以使用概率分布 P_{tl} 表示左上角热力图; 同理右下角热力图可以表示为 P_{br}. 可以使用概率分布 P_{tl}(或 P_{br}) 的均值表示跟踪目标的预测框的左上角 (或右下角) 的坐标, 即

$$\begin{cases} (X, Y) = (16x + 8, 16y + 8), \\ \hat{x}_{tl} = \sum\limits_{x=0}^{w}\sum\limits_{y=0}^{h} X \cdot P_{tl}(x, y), \\ \hat{y}_{tl} = \sum\limits_{x=0}^{w}\sum\limits_{y=0}^{h} Y \cdot P_{tl}(x, y), \\ \hat{x}_{br} = \sum\limits_{x=0}^{w}\sum\limits_{y=0}^{h} X \cdot P_{br}(x, y), \\ \hat{y}_{br} = \sum\limits_{x=0}^{w}\sum\limits_{y=0}^{h} Y \cdot P_{br}(x, y). \end{cases} \quad (10)$$

其中: (x, y) 表示热力图上的位置坐标, (X, Y) 表示热力图映射回原图片的位置坐标, (w, h) 表示热力图的宽和长, $(\hat{x}_{tl}, \hat{y}_{tl}, \hat{x}_{br}, \hat{y}_{br})$ 表示预测结果.

3.3　损失函数

本文借鉴了 DETR (detection transformer)[21] 中使用的损失函数进行训练, 损失函数的计算如下所示:

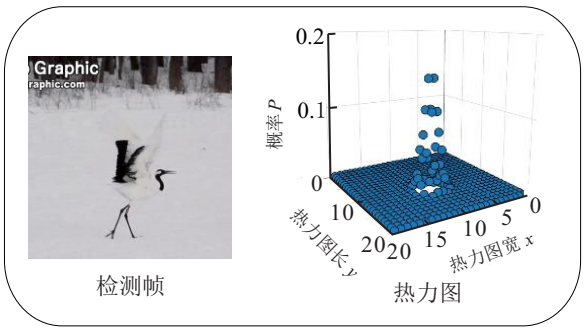
$$L_1 = \lambda_{\text{giou}} L_{\text{giou}}(\text{box}, \hat{\text{box}}) + \lambda_{L_1} L_{L_1}(\text{box}, \hat{\text{box}}). \quad (11)$$

其中: $L_{\text{giou}}(\cdot)$ 表示 giou 损失函数, $L_1(\cdot)$ 表示 L_1 损失函数, box 表示真实框, $\hat{\text{box}}$ 表示预测框, λ_{giou} 和 λ_{L_1} 表示超参数.

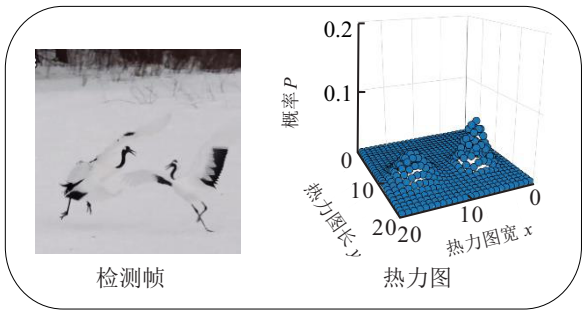
如图 5 所示, 当跟踪的场景中存在多个与目标非常相似的干扰物时, 预测头输出的热力图可能会出现多个极值点. 为解决上述问题, 本文在训练时引入一个基于 KL 散度的辅助训练头. KL 散度是用来度量两个概率分布相似度的指标, 适用于上述场景, 其计算如下所示:

$$D_{\text{KL}} = \sum_{i=0}^{n} P \log \frac{P}{Q}. \quad (12)$$

其中: P 表示数据的真实分布, Q 表示预测分布.



(a) 无干扰物



(b) 存在干扰物

图 5 有无干扰物对热力图的影响

基于KL散度的辅助训练头的损失函数的标签(真实分布 $P(x,y)$)为: 以目标的真实框box为中心的高斯分布, 其计算公式为

$$\begin{cases} \text{box} = (\text{cx}, \text{cy}, \text{cw}, \text{ch}), \\ \sigma_w = \dfrac{\text{cw}}{8}, \\ \sigma_h = \dfrac{\text{ch}}{8}, \\ P(x,y) = \dfrac{1}{\sqrt{2\pi\sigma_w\sigma_h}} \mathrm{e}^{-\frac{1}{2}\left[\frac{(x-\text{cx})^2}{\sigma_w^2} + \frac{(y-\text{cy})^2}{\sigma_h^2}\right]}. \end{cases} \tag{13}$$

其中: (cx, cy) 表示真实框box的中心点; cw, ch 表示真实框box的宽和长.

辅助训练头的输出是得分图 $s(x,y)$, 表示目标在检测帧 (x,y) 处的分数. 可以使用下述公式将得分图 $s(x,y)$ 转换为数据的预测分布 $Q(x,y)$, 即

$$Q(x,y) = \frac{\mathrm{e}^{s(x,y)}}{\sum\limits_{x=0}^{w}\sum\limits_{y=0}^{h}\mathrm{e}^{s(x,y)}}. \tag{14}$$

基于KL散度的辅助训练头的损失函数 L_2 可以表示为

$$L_2 = \log\sum_{x=0}^{w}\sum_{y=0}^{h}\mathrm{e}^{s(x,y)} - \sum_{x=0}^{w}\sum_{y=0}^{h}P(x,y)s(x,y). \tag{15}$$

该辅助训练头的作用是强迫网络学习到每次只对单一位置有强响应能力, 避免热力图对多处有较高概率值而造成目标框发散. 辅助训练头仅在训练阶段起作用, 训练完成后删掉. 最终得到第一阶段的总损失函数

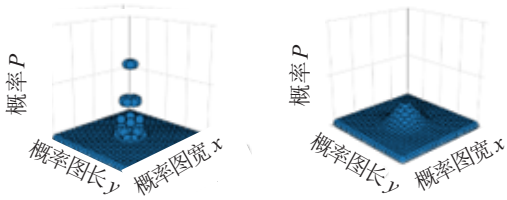
$$L = L_1 + L_2. \tag{16}$$

4 评价模块和更新模块

如图4所示, 当采用并行多外观特征策略时, 会得到两个预测结果. 必须从这两个预测结果中筛选出最佳的一个作为最终的预测结果, 并决定是否用最终的预测结果更新动态模板帧. 本节提出两种方法解决这一问题: 1) 基于信息熵的方法, 该方法速度较快; 2) 基于IOU-Net (intersection over union network) 模块的方法, 该方法精度较高.

4.1 基于信息熵的方法

1) 基于信息熵的评价模块: 已知左上角(右下角)热力图表示左上角(右下角)的概率分布. 如图6(a)所示, 当概率分布比较集中、呈瘦高状时, 可以认为左上角(右下角)的预测更确定; 如图6(b)所示, 当概率分布比较分散、呈扁平状时, 可以认为左上角(右下角)的预测非常不确定, 得到的跟踪结果可靠性低.



(a) 确定的热力图 (b) 不确定的热力图

图 6 热力图可视化

信息熵可以描述信息源各可能事件发生的不确定性, 适用于上述场景, 其计算如下所示:

$$H = -\sum_{i=1}^{n} p(x_i)\log p(x_i). \tag{17}$$

计算每一个预测结果 $\hat{\text{box}}^n$ 左上角热力图的信息熵 H_{tl} 和右下角热力图的信息熵 H_{br}, 当信息熵之和 $H_{tl}+H_{br}$ 最小时, 认为该预测结果最佳. 预测结果的热力图信息熵之和的计算公式如下所示:

$$\begin{cases} H_{tl} = \sum\limits_{x=0}^{w}\sum\limits_{y=0}^{h}P_{tl}(x,y)\cdot\log P_{tl}(x,y), \\ H_{br} = \sum\limits_{x=0}^{w}\sum\limits_{y=0}^{h}P_{br}(x,y)\cdot\log P_{br}(x,y), \\ H = H_{tl} + H_{br}. \end{cases} \tag{18}$$

2) 基于信息熵的更新模块: 更新模块的作用是判断当前得到的最好预测结果是否可以成为新的动态模帧指导下一帧的跟踪. 对于基于信息熵的评价模块得到的预测结果, 本文记录训练集视频每一帧

的左上角热力图信息熵 H_{tl} 和右下角热力图信息熵 H_{br}, 如图 7 信息熵曲线所示. 然后用 4 次多项式分别拟合 H_{tl} 和 H_{br}, 如图 7 拟合曲线所示. 最后分别求出左上角热力图信息熵拟合曲线和右下角热力图信息熵拟合曲线的拐点. 当信息熵大于拐点值时, 说明信息熵处于加速变大区域, 即处于不确定区域; 当信息熵小于拐点值时, 说明信息熵处于加速变小区域, 即处于确定区域. 因此可以将拐点值作为判断信息熵是否处于确定区域的阈值. 通过计算求得左上角热力图信息熵 H_{tl} 和右下角热力图信息熵 H_{br} 的阈值分别为: 左上角阈值 $H_{tl}^{\text{thr}} = 1.46$; 右下角阈值 $H_{br}^{\text{thr}} = 1.50$.

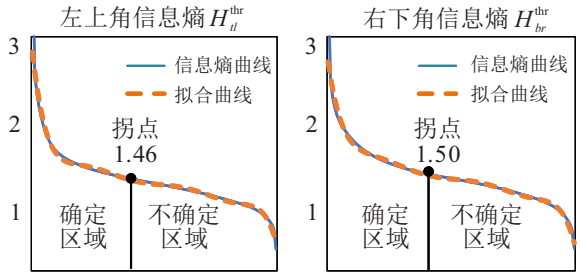


图 7　热力图信息熵及拟合曲线

为防止出现信息熵很低但预测错误, 导致错误的预测结果成为动态模板帧指导后续的跟踪, 需增加更严格的更新条件: 只有预测头模块得到的两个预测结果指向同一个目标才会考虑更新动态模板帧. 综上, 对于使用基于信息熵的评价模块得到的预测结果, 当其左上角热力图信息熵和右下角热力图信息熵满足如下关系时, 对动态模板帧进行更新:

$$\begin{cases} \text{IOU}(\hat{\text{box}}^{1}, \hat{\text{box}}^{2}) > 0.75, \\ H_{tl} < 1.46, \\ H_{br} < 1.50. \end{cases} \tag{19}$$

4.2　基于 IOU-Net 模块的方法

1) 基于 IOU-Net 模块的评价模块: 基于信息熵的评价模块虽然能简单高效地筛选出最确定的预测结果, 但该预测结果不是最准确的预测结果, 可能出现信息熵很低, 但预测结果不准确的情况. 为此, 本文使用 ATOM (accurate tracking by overlap maximization) 的 IOU-Net 模块[22] 作为评价模块. 该模块可以预测第 n 个预测结果 $\hat{\text{box}}^{n}$ 与真实框之间的 IOU, IOU 越大, 表示预测结果 $\hat{\text{box}}^{n}$ 越准确. 相较于基于信息熵的评价模块, 基于 IOU-Net 的评价模块效果更好, 但速度较慢, 可以根据实际情况选择合适的方法. 本文算法利用 ATOM 中的损失函数对 IOU-Net 模块进行训练, 损失函数的计算如下所示:

$$L_3 = \sum_{i=1}^{n} (\text{iou}_p - \text{iou})^2. \tag{20}$$

其中: iou_p 表示评价模块预测的 IOU 值, iou 表示真实的 IOU 值.

2) 基于 IOU-Net 模块的更新模块: 对于使用基于 IOU-Net 模块的评价模块得到的预测结果 $\hat{\text{box}}$, 当 IOU-Net 模块输出的预测 iou_p 大于阈值 iou_{thr} 时, 则考虑更新动态模板帧. 通过实验发现, 该方法对超参 iou_{thr} 不敏感, 当 $\text{iou}_{\text{thr}} \in (0.5, 0.95)$ 时都会取得较好的实验结果, 具有较强的泛化能力. 综上, 对于使用基于 IOU-Net 模块的评价模块得到的预测结果 $\hat{\text{box}}$, 当满足如下关系时, 对动态模板帧进行更新:

$$\begin{cases} \text{IOU}(\hat{\text{box}}^{1}, \hat{\text{box}}^{2}) > 0.75, \\ \text{iou}_p > \text{iou}_{\text{thr}}. \end{cases} \tag{21}$$

在下一帧的跟踪过程中, 将用新得到的动态模板帧替换以前的动态模板帧, 用于指导下一帧的跟踪.

5　实　验

5.1　训练细节

本文算法在 CPU 为 Intel core (TM) i7-8700 K、主频为 3.70 GHz、 内存为 32 G、 GPU 为 Nvidia GTX 1080Ti、操作系统为 64 位 Ubuntu16.04 的台式机上进行, 使用 Python 3.6 和 PyTorch 1.5.1 实现. 训练数据集包括 Lasot[23]、GOT-10k[24] 和 COCO2017[25]. 模板帧和检测帧的大小分别为 128×128 像素和 320×320 像素, 对应于目标框的 2^2 倍和 5^2 倍. 数据增强包括水平翻转和模糊. 整个训练包括 2 个阶段: 第 1 阶段对特征提取模块、外观特征处理模块、互相关模块、预测头模块训练 500 个 epoch, 训练数据的最小单元由 2 张模板帧和 1 张检测帧组成; 第 2 个阶段对基于 IOU-Net 模块的评价模块训练 50 个 epoch, 训练数据的最小单元由 1 张模板帧与 1 张检测帧组成. 每个批次采样 16 组最小单元, 每个 epoch 采样 60 000 组最小单元. 使用 AdamW 优化器对网络进行优化, 权值衰减系数为 10^{-4}. 特征提取模块使用 ImageNet (image data set for networking) 上预先训练的参数初始化, 其他模块随机初始化. 超参数 λ_{giou} 和 λ_{L_1} 分别取 5 和 2. 特征提取模块和其他模块的初始学习率分别为 10^{-5} 和 10^{-4}, 学习率在第 1 阶段的 400 个 epoch 和第 2 个阶段的 40 个 epoch 之后下降 10 倍[18].

5.2　评价指标

使用的评价指标为: 平均重合度 (average overlap, AC)[24]、成功率 (success rate, SR)、AUC (area under curve)、归一化精确度 (normalized precision, P_{norm})、

精确度(precision, p)、跟踪速率(frames per second, FPS)[23]. $SR_{0.5}$ 和 $SR_{0.75}$ 分别表示AO阈值为0.5和0.75时成功跟踪的比率[24].

5.3 对比实验

本文算法在标准数据集GOT-10k、OTB100[26]、UAV123[27]、Lasot上与SiamFC、ATOM、SiamFC++、DiMP50、SiamRCNN (Siamese regional convolutional neural networks)[28]、TrDimp (tracking by discriminative model prediction)[29]、Ocean、TransT这些经典算法做对比实验.

1) GOT-10k: 如表1所示, 在使用相同的ResNet-50作为特征提取模块的情况下, 本文提出的算法(ours)比Ocean的AO高7.4%, 比使用ResNet-101的SiamRCNN的AO高3.6%; 与同样使用Transformer的TrDimp和TransT相比, 本文算法的AO比其高1.4%.

表1 GOT-10k数据集上的对比实验

算法	AO	$SR_{0.5}$	$SR_{0.75}$
SiamFC	34.8	35.3	9.8
ATOM	55.6	63.4	40.2
SiamFC++	59.5	69.5	47.9
DiMP50	61.1	71.7	49.2
Ocean	61.1	72.1	47.3
SiamRCNN	64.9	72.8	59.7
TrDiMP	67.1	77.7	58.3
TransT	67.1	76.8	60.9
ours	**68.5**	**77.5**	**61.8**

2) Lasot: 如表2所示, 在使用相同的ResNet-50作为特征提取模块时, 本文算法(ours)比Ocean的AUC高8.4%, 在同等速度下比ATOM的AUC高12.9%, 比DiMP50的AUC高7.5%. 虽然SiamRCNN在Lasot数据集上取得了最好的成绩, 但是本文算法比SiamRCNN (5 fps)快7倍以上, 能够以实时速度运行. 可见本文算法在性能和速度上有很好的平衡, 能够以较高性能实时运行.

表2 Lasot数据集上的对比实验

算法	AUC	P_{norm}	p	speed (FPS)
SiamFC	33.6	42.0	33.9	86
ATOM	51.5	57.6	50.5	35
SiamFC++	54.4	62.3	54.7	30
DiMP50	56.9	65.0	56.7	35
Ocean	56.0	65.1	56.6	25
SiamRCNN	64.8	72.2	68.4	5
Ours	**64.4**	**72.0**	**68.2**	**37**

3) UAV123: 如表3所示, 本文提出的算法(ours)在UAV123数据集上表现优异, 比DiMP50的AUC高2.5%.

表3 UAV123和OTB数据集上AUC的对比

数据集	**Ours**	DiMP50	SiamRPN++	ATOM
UAV123	**67.8**	65.3	61.3	64.2
OTB	**68.1**	68.4	69.6	66.9

4) OTB: 其包含100个具有挑战性的视频, 包括25%的灰度视频. 如表3所示, 虽然本文算法(ours)的AUC比DiMP50低0.3%, 比SiamRPN++低1.5%, 但是也取得了较高的成绩, AUC达到了68.1%, 具有很强的竞争力. 本文提出的算法在该数据集上表现较差的原因是: 没有对训练数据使用灰度增强, 在灰度视频上表现较差.

由上述对比实验可以看出, 本文提出的算法(ours)在多个数据集上都展现出优秀的性能, 表明其具有很强的泛化能力, 是一个极具竞争力的算法.

5.4 消融实验

为了探究基于IOU-Net的更新模块的IOU阈值 iou_{thr} 对算法性能的影响, 设计如下的实验: 使用基于IOU-Net的评价模块和更新模块, 用并行多外观特征策略在测试集UAV123上进行实验. 其中 $\text{iou}_{\text{thr}} \in [0.5, 0.95]$, 取步长为0.5. 实验结果如图8所示, 当IOU阈值 iou_{thr} 取不同值时, AUC的变化不明显, 说明本文提出的基于IOU-Net的更新模块是一个超参不敏感的方法, 具有很强的泛化性. 在本文中 iou_{thr} 取0.75.

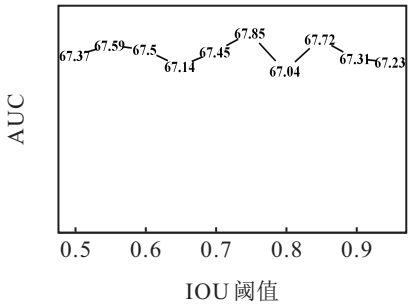


图8 IOU阈值 iou_{thr} 对跟踪性能的影响

为分析简单地使用动态模板帧替换初始模板帧(表4中用 a 表示)、本文提出的多外观特征(表4中用 b 表示)、并行外观特征(表4中用 c 表示)、并行多外观特征(表4中用 d 表示)对算法性能的影响, 在GOT-10k上进行了实验, 实验效果如表4所示.

表4 不同的利用动态模板帧的方法对跟踪性能的影响

#	a	b	c	d	AO
√					66.9
	√				66.4
		√			67.7
			√		68.0
				√	68.5

在表4中: #表示仅使用初始模板帧进行跟踪, a、

b、c、d均使用基于IOU-Net的评价模块和更新模块去更新动态模板帧. 可以看到,简单地使用动态模板帧替换初始模板帧a会使算法的AO略微下降(下降0.5 %);本文提出的3种方法都可以提高算法的性能,其中多外观特征b提升了0.8 %,并行外观特征c提升了1.1 %,并行多外观特征d提升了1.6 %. 值得注意的是:多外观特征b与并行外观特征c的计算量几乎相同,但并行外观特征c比多外观特征b提高了0.3 %,说明将多个外观特征并行输入到网络中确实会提高算法的性能.

另外,还可以使用基于信息熵的评价模块和更新模块更新动态模板帧. 如表5所示,使用信息熵比使用IOU-Net的性能略低,但却更快,计算量更小. 因此可以根据具体的需求选择不同的方法更新动态模板帧.

表 5　IOU-Net与信息熵对比

数据集	方法	AUC/AO	FPS
GOT-10K	IOU-Net	68.50	37
Lasot		64.42	
UAV123		67.85	
OTB100		68.09	
GOT-10k	信息熵	68.20	40
Lasot		64.21	
UAV123		67.47	
OTB100		67.95	

为验证互相关模块、预测头模块,以及基于KL散度的损失函数对本文算法的影响,在GOT-10k上设计以下消融实验:1) 为了验证互相关模块对算法的影响,用Ocean的深度互相关(depth-wise correlation, DW)模块代替本文算法的互相关模块;2) 为了验证预测头模块对算法的影响,分别用DETR (detection transformer)的基于多层感知机(multilayer perceptron, MLP)的预测头模块和跟踪中常用的分类-回归(categorical-regression, CR)预测头模块代替本文算法的预测头模块;3) 去掉基于KL散度的损失函数(-KL). 为了实验简单起见,仅使用初始外观特征进行跟踪(base).

如表6所示:1) 基于Transformer的互相关模块在本文算法中起非常重要的作用,在AO上比DW高9.5 %. 2) 本文算法使用的预测头模块比MLP和CR更加准确,比MLP高3.7 %,比CR高2.5 %. CR生成了大量带有置信度分数的预测框,通常会使用尺度/比例惩罚、边界框平滑等操作提高算法的精度,实验CR_PLUS遵循了这一传统,其在AO上与本文算法性能相当,但是这些操作涉及大量超参数,为发挥算法的最佳性能,需要针对不同的数据集设计不同的超参数,不方便后续算法的部署. 3) 使用基于KL散度的损失函数辅助训练比不使用(-KL)高0.8 %,说明本文算法使用的基于KL散度的损失函数的辅助训练头可以提高算法的性能.

表 6　不同的利用动态模板帧的方法对跟踪性能的影响

base	DW	MLP	CR	CR_PLUS	-KL	AO
√						66.9
	√					57.4
		√				63.2
			√			64.4
				√		66.7
					√	66.1

6　结　论

本文针对目标跟踪通常仅使用视频的第一帧在线学习目标的外观特征,而目标的外观信息却是不断变化的这一问题,提出了一种新的基于并行多外观特征的孪生网络跟踪算法,具有很强的泛化能力. 其中,并行外观特征和并行多外观特征方法虽然增加了外观特征的参数量和算法的计算量,但却很好地发挥了现代GPU的并行计算能力,并不会降低算法的跟踪速度. 并行外观特征方法和基于信息熵的评价模块、更新模块不需要太多额外的训练就可以提升算法的性能. 并行多外观特征方法和基于IOU-Net模块的评价模块、更新模块可以最大程度地发挥算法的性能,但需要重新训练. 实验结果表明,本文提出的方法在多个标准数据集上展现出了较高的性能,验证了所提出方法的有效性.

参考文献(References)

[1] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional Siamese networks for object tracking[C]. European Conference on Computer Vision. Cham, 2016: 850-865.

[2] 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.

[3] 李辉, 董燕, 刘祥, 等. 基于两阶段深度网络的输电线路异常目标检测方法[J]. 控制与决策, 2022, 37(7): 1873-1882.
(Li H, Dong Y, Liu X, et al. Transmission line abnormal object detection method based on deep network of two-stage[J]. Control and Decision, 2022, 37(7): 1873-1882.)

[4] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[5] Zhang Z P, Peng H W. Deeper and wider Siamese networks for real-time visual tracking[C]. 2019

IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, 2019: 4591-4600.

[6] 宋建辉, 孙晓南, 刘晓阳, 等. 融合HOG特征和注意力模型的孪生目标跟踪算法[J]. 控制与决策, 2023, 38(2): 327-334.
(Song J H, Sun X N, Liu X Y, et al. Twin target tracking network combining HOG features and attention model[J]. Control and Decision, 2023, 38(2): 327-334.)

[7] 刘如浩, 张家想, 金辰曦, 等. 基于可变形卷积的孪生网络目标跟踪算法[J]. 控制与决策, 2022, 37(8): 2049-2055.
(Liu R H, Zhang J X, Jin C X, et al. Target tracking based on deformable convolution Siamese network[J]. Control and Decision, 2022, 37(8): 2049-2055.)

[8] 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. Long Beach, 2019: 4282-4291.

[9] 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. Las Vegas, 2016: 770-778.

[10] Xu Y D, Wang Z Y, Li Z X, et al. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 12549-12556.

[11] Zhang Z P, Peng H W, Fu J L, et al. Ocean: Object-aware anchor-free tracking[C]. European Conference on Computer Vision. Cham, 2020: 771-787.

[12] Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions[C]. 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, 2015: 1-9.

[13] Xie F, Wang C Y, Wang G T, et al. Correlation-aware deep tracking[C]. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans, 2022: 8751-8760.

[14] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30: 5998-6008.

[15] Bhat G, Danelljan M, Van Gool L, et al. Learning discriminative model prediction for tracking[C]. 2019 IEEE/CVF International Conference on Computer Vision. Seoul, 2019: 6182-6191.

[16] Danelljan M, Van Gool L, Timofte R. Probabilistic regression for visual tracking[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, 2020: 7183-7192.

[17] Chen X, Yan B, Zhu J W, et al. Transformer tracking[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, 2021: 8126-8135.

[18] Yan B, Peng H W, Fu J L, et al. Learning spatio-temporal transformer for visual tracking[C]. 2021 IEEE/CVF International Conference on Computer Vision. Montreal, 2021: 10448-10457.

[19] Mayer C, Danelljan M, Bhat G, et al. Transforming model prediction for tracking[C]. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans, 2022: 8731-8740.

[20] Kristan M, Leonardis A, Matas J, et al. The eighth visual object tracking VOT2020 challenge results[C]. Computer Vision — ECCV 2020 Workshops. Glasgow, 2020: 547-601.

[21] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]. Computer Vision — ECCV 2020. Cham: Springer International Publishing, 2020: 213-229.

[22] 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. Long Beach, 2019: 4660-4669.

[23] Fan H, Lin L T, Yang F, et al. LaSOT: A high-quality benchmark for large-scale single object tracking[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, 2019: 5374-5383.

[24] Huang L H, Zhao X, Huang K Q. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(5): 1562-1577.

[25] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common objects in context[C]. European Conference on Computer Vision. Cham, 2014: 740-755.

[26] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark[C]. 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland, 2013: 2411-2418.

[27] Mueller M, Smith N, Ghanem B. A benchmark and simulator for UAV tracking[C]. European Conference on Computer Vision. Cham, 2016: 445-461.

[28] 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. Seattle, 2020: 6578-6588.

[29] Wang N, Zhou W G, Wang J, et al. Transformer meets tracker: Exploiting temporal context for robust visual tracking[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, 2021: 1571-1580.

作者简介

陈志旺 (1978–), 男, 副教授, 博士, 硕士生导师, 从事运动物体目标检测与跟踪等研究, E-mail: czwaaron@ysu.edu.cn;

孙泽兵 (1998–), 男, 硕士, 从事计算机视觉中目标跟踪的研究, E-mail: 893130323@qq.com;

吕昌昊 (1996–), 男, 硕士, 从事智能电网的优化和控制等研究, E-mail: 316998054@qq.com;

曹索航 (1999–), 男, 硕士, 从事计算机视觉中目标跟踪的研究, E-mail: 1127001852@qq.com;

彭勇 (1963–), 男, 教授, 博士生导师, 从事生物机器人控制等研究, E-mail: PY81@sina.com.