计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(7):1−13 [doi: 10.15888/j.cnki.csa.009580]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

全局搜索和多实例判别特征的长时跟踪方法①

肖诗逢[1], 程　旭[2]

[1](南京信息工程大学 软件学院, 南京 210044)
[2](南京信息工程大学 计算机学院、网络空间安全学院, 南京 210044)
通信作者: 肖诗逢, E-mail: 1485003598@qq.com

摘　要: 长时目标跟踪相对于短时目标跟踪仍然是一个巨大的挑战. 然而现有的长时跟踪算法通常在面对目标频繁出现消失、目标外观发生剧变等挑战中表现不佳. 本文提出了一种基于局部搜索模块和全局搜索跟踪模块的全新、鲁棒且实时的长时跟踪框架. 局部搜索模块利用 TransT 短时跟踪器生成一系列候选框, 并通过置信度评分确定最佳候选框. 针对全局重新检测开发了一个新颖的全局搜索跟踪模块, 以 Faster R-CNN 为基础模型, 在 RPN 阶段与 R-CNN 阶段引入非局部操作和多级实例特征融合模块, 以充分挖掘目标实例级特征. 为了改进全局搜索跟踪模块的性能, 设计了双模板更新策略来提升跟踪器的鲁棒能力. 通过使用不同时间点上更新的模板能够更好地适应目标的变化. 根据局部或全局置信度分数判断目标是否存在, 并在下一帧中选择局部或全局搜索跟踪策略. 同时能够为局部搜索模块估计目标的位置和大小. 此外还为全局搜索跟踪器引入了排名损失函数, 隐式学习了区域提议与原始查询目标的相似度. 通过在多个跟踪数据集上进行大量实验对提出的跟踪框架进行了广泛评估. 结果一致表明, 本文提出的跟踪框架实现了令人满意的性能.

关键词: 视觉目标跟踪; 长时跟踪; 全局搜索跟踪; 模板更新

引用格式: 肖诗逢,程旭.全局搜索和多实例判别特征的长时跟踪方法.计算机系统应用,2024,33(7):1–13. http://www.c-s-a.org.cn/1003-3254/9580.html

Long-term Tracking Method with Global Search and Multiple Instance Discriminative Features

XIAO Shi-Feng[1], CHENG Xu[2]

[1](School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China)
[2](School of Computer Science & School of Cyber Science and Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China)

Abstract: Long-term object tracking remains a formidable challenge compared to short-term object tracking. However, existing long-term tracking algorithms often perform poorly when faced with challenges such as targets frequently appearing and disappearing, and drastic changes in target appearance. This study proposes a novel, robust, and real-time long-term tracking framework based on local search modules and global search tracking modules. The local search module utilizes the TransT short-term tracker to generate a series of candidate boxes, and the best candidate box is determined through confidence scoring. A novel global search tracking module is developed for global re-detection, based on the Faster R-CNN model, with the introduction of Non-Local operations and multi-level instance feature fusion modules in the RPN and R-CNN stages, aiming to fully exploit target instance-level features. To improve the performance of the global search tracking module, a dual-template update strategy is designed to enhance the robustness of the tracker. By utilizing templates updated at different time points, the tracker can better adapt to target changes. The target presence is determined based on local or global confidence scores, and the local or global search tracking strategy is selected in the

① 基金项目: 国家自然科学基金 (61802058, 61911530397); 中国博士后科学基金 (2019M651650)
收稿时间: 2024-01-02; 修改时间: 2024-02-26; 采用时间: 2024-03-18; csa 在线出版时间: 2024-06-05
CNKI 网络首发时间: 2024-06-08

next frame. Additionally, the local search module is capable of estimating the position and size of the target. Moreover, a ranking loss function is introduced for the global search tracker, implicitly learning the similarity between region proposals and the original query target. A large number of experiments are conducted on multiple tracking datasets to comprehensively assess the proposed tracking framework. The results consistently demonstrate that the proposed tracking framework achieves satisfactory performance.

Key words: visual object tracking; long-term tracking; global search tracking; template update

目标跟踪作为计算机视觉领域的关键技术[1], 取得了显著的突破和广泛的应用. 目标跟踪在智能监控系统和增强现实[2]的应用中都扮演着至关重要的角色. 目标跟踪技术[3−10]的持续突破给各行业带来了重大变化, 为该技术的未来发展开辟了更多的机遇. 现有的短时跟踪器[3−5,11]在短时跟踪基准测试中也得到了令人满意的结果. 然而在长时跟踪场景中, 现有的短时跟踪器经常面临一些无法解决的挑战. 在长时间跟踪中, 目标物体可能会经历遮挡、光照变化、背景干扰等情况, 导致频繁丢失或重新出现. 这些因素会影响跟踪器的连续性和稳定性, 使得跟踪目标变得困难. 此外, 目标在长时间跟踪中可能会发生外观变化, 如姿态变化、形状变化等. 这些复杂的形变使得现有的短时跟踪器难以准确捕捉目标的外观特征, 从而导致跟踪性能下降. 另一个挑战是长时间跟踪需要在大量的视频帧上进行处理, 而现有的短时跟踪器在处理复杂场景时可能需要消耗较长的时间. 这导致跟踪器无法满足实时性的要求, 可能会存在延迟问题. 最后, 由于累积误差的存在, 长时间跟踪器可能会发生漂移, 即跟踪框架与目标的真实位置之间的偏差逐渐增大. 这是因为短时跟踪器无法很好地处理长时间的运动变化和背景干扰, 导致误差逐渐累积. 为了实现实时性能, SPLT 方法[7]引入了 Skimming 模块, 通过离线训练快速提取候选区域. 针对目标频繁出现和消失的挑战, ELGLT 方法[8]引入了重新检测模块, 通过离线训练直接估计整个图像中目标的状态. 这种方式可以降低非目标区域的影响, 提高跟踪的连续性. DMTrack 方法 [12]结合了 re-id 嵌入空间和全局重新检测框架, 可以在相邻帧之间关联检测结果, 应对目标干扰物的问题. 针对目标遮挡的挑战, Li 等人[13]引入了“检测先于跟踪”的概念, 并使用基于卷积神经网络的轨迹预测模块. 该模块利用目标的时间运动信息, 减小遮挡物体的影响, 提高跟踪的鲁棒性.

但目前无论是已有的局部-全局跟踪方法还是全局跟踪方法, 都存在各自的局限性. 为了更好地应对目标频繁消失与出现、达到实时性以及处理目标外观变化等挑战, 本文提出了一种高效的局部-全局跟踪框架. 针对目标频繁消失与出现的挑战, 使用两阶段目标检测模型 Faster R-CNN[14]作为全局搜索跟踪器的基础模型, 该方法将全局检测问题视为目标检测的一种特殊形式. 全局搜索跟踪器包括 3 个主要模块: 负责生成初级候选框的非局部区域建议网络 (Non-Local RPN), 新颖的多级特征融合模块, 以及用于精确分类和回归网络. 全局搜索跟踪器专注于准确高效地重新找回丢失的目标, 并为局部跟踪器提供目标位置. 与其他重检测器不同的是, 当目标的置信度分数较低时, 全局搜索跟踪模块本身也能够提供确定的目标位置和大小, 而不会将不确定的目标区域传递给局部跟踪模块. 换句话说, 本文提出的跟踪框架能够有效地防止跟踪漂移, 使其能够与大多数短时跟踪器无缝集成, 实现高效的长期跟踪. 针对目标外观变化剧烈问题, 采用了一种可学习的模板更新方法, 将动态模板特征与初始模板特征相结合. 混合特征融合允许全局搜索跟踪模块有效地捕获和适应在整个推理过程中对象外观的变化. 针对干扰物的问题, 还引入了排名损失, 以增强全局搜索跟踪器对实例级干扰物的判别能力.

1 长时目标跟踪概述

在本节中根据长时跟踪方法的总体构造, 将其分为两类: 局部-全局跟踪方法和全局跟踪方法.

1.1 局部-全局跟踪方法

局部-全局跟踪器可被视为对局部跟踪器的增强, 在这种跟踪器一般由局部跟踪器、重检测器和验证模块组成. 在早期工作[15,16]当中通过局部-全局切换的策略来解决长时目标跟踪任务. 将全局重新检测器与局部跟踪器结合是一种非常有效的方法, 特别是在局部

跟踪器遇到失败或限制的情况下, 这种方法能够提升算法的跟踪性能. 在 2015 年, 基于检测的跟踪器 TLD[14] 首次被提出, 它将基于光流的局部跟踪器与使用弱分类器集合的全局检测结合起来, 同时引入在线更新机制, 通过不断更新局部跟踪模块的“显著特征点”和检测模块的目标模型及相关参数来抵抗目标外观变化. 在此范式的基础上, Ma 等人[16]将目标跟踪问题分解为对目标的平移估计和尺度估计两个步骤, 前者采用上下文相关性负责预测目标位置变化, 后者采用相关滤波负责目标尺度变化, 同时运用随机蕨类分类器作为重检测器来处理目标丢失. 但这种基于传统特征的方法是具有局限性的, 即传统手工特征无法应对更为复杂的跟踪场景[17,18].

与此同时, 深度学习在长时跟踪领域的应用也取得了令人鼓舞的成果. 为了解决目标丢失问题, Bertinetto 等人提出了一种将重检测器纳入 SiamFC [11]短时跟踪器的长时跟踪方法, 通过使用一种简单的重新检测策略, 考虑每一帧中随机位置的搜索区域, 若区域分数超过预定义的阈值则送入 SiamFC 当中进行局部跟踪. 为了获得更优的候选框, Zhu 等人[19]基于 EdgeBox 边缘分割方法获得特定实例的目标候选框, 并采用 SVM[20] 分类器进行验证. MBMD[6]引入了一种创新的跟踪框架, 它由边界框回归网络与验证网络两部分构成. 前者由 RPN 网络与类似 SiamFC 的特征融合模块组成, RPN 会输出一系列候选框及相似度分数, 全部送入在线更新的验证网络当中去判别. 如果两个网络都无法找到一个既与目标相似又被分类为前景的候选框, 那么将启动全图搜索机制. 在这种机制下, 跟踪器将使用滑窗方式在整个图像中进行检测. 跟踪器会根据置信度得分在局部搜索和全局搜索之间动态切换. 然而, 上述方法的复杂性使得速度无法达到实时性 (仅达到 2.7 f/s). 为了实现速度和精度之间的平衡, ELGLT[8]则通过提取高质量的目标候选模块和目标验证器的组合. 这种方法能够最大可能的排除干扰物来确定最优的候选框. 同时为了抵抗目标外观变化还设计了一个长短时更新策略 (使用不同时间更新的多个模板进行特征融合) 来提升目标验证器的性能, 根据置信度分数就可以判断跟踪对象是否存在在画面中, 由此在下一帧中分别选择局部或全局的跟踪策略. 然而, 仍有一些因素阻碍了这些局部-全局跟踪器的性能. 一个因素是在复杂的情况下, 如干扰物以及物体部分以及完全遮挡, 重新检测器易向干扰物漂移. 另一个因素是验证模块本身的识别能力有限, 无法识别出最优候选框.

1.2 全局跟踪方法

还有一些研究将长时目标跟踪视为对于特定目标的检测任务. 这种类型的跟踪器基于检测框架生成, 它独立分析每一帧并充当专门的目标跟踪器. 在扫描整个图像时, 它能够提供可靠的目标候选框. Dave 等人[21] 引入了一种轻量级跟踪策略, 利用 Mask R-CNN[22]架构将特定类别的目标检测器转换为目标特定的检测器, 将目标特征与搜索特征进行融合得到注意力权重, 再利用得到的注意力权重对搜索特征进行加权, 以检测跟踪目标. 同时并提出了一种端到端计算鉴别目标模板的线性分类器, 以有效地处理干扰物. 与其相似的是, GlobalTrack[9]利用 Faster R-CNN 检测框架作为基准模型, 在 RPN 与 R-CNN 阶段同时利用目标信息来引导网络搜索特定目标信息. 它对视频每一帧的跟踪完全不依赖相邻帧, 没有累计跟踪误差, 这就使得它在长时跟踪场景下跟踪稳定, 但也导致对目标外观变化非常敏感. 与 GlobalTrack 的方法类似, Zhang 等人[12]通过纳入重识别 (re-id) 关联, 扩展了特定目标拼接检测网络. 为了融入时间上下文信息, DMTrack[12]受多目标跟踪算法[23]的启发, 将判别性嵌入空间纳入全局检测方法中. 这样可以捕获相邻帧之间的检测结果, 并利用上下文信息进一步提升跟踪性能, 利用先前的检测结果来提高精度. 与此同时该算法能够达到 31 f/s 的速度. 美中不足的是, 使用的 re-id 方法与判别性嵌入空间仅在行人数据集上起到较为明显的作用. 总体而言, 这种基于全局的跟踪策略在实时性与鲁棒性等方面存在不同程度的缺陷.

此外, 基于 Transformer 的长时跟踪器也已经有了显著的进展. STARK [24]受 DETR[25]的启发, 采用传统的 Transformer 架构, 将时间信息与空间信息看作一个整体, 沿空间维度连接第 1 帧模板、当前帧搜索区域以及动态模板的骨干网络特征. 这会产生一个结合了空间和时间信息的特征序列, 然后将其用作 Transformer 编码器的输入, 以学习强大的时空表示. Song 等人[26] 采用了多尺度循环变换窗口关注机制, 从像素到窗口转移注意力. 该方法能够在保持目标完整性的同时找到目标对象的最佳匹配. 然而, 它对相互关系的依赖限制了它的能力. 另一方面, TransT [27]则提出使用 Transformer 代替互相关. 它生成融合特征, 包含更全面的语

义信息, 而不仅依赖响应得分. 因此, 这些方法与先前的 Siamese 跟踪器相比, 实现了显著提高的跟踪准确性. 但是需要注意的是, 这些基于 Transformer 的跟踪器依赖于训练集, 导致模型可能在学习数据集分布和任务目标方面不一致. 为了解决这个问题, SLT [28]在现有方法的基础上引入了基于强化学习的序列级训练策略, 以提高 Transformer 跟踪算法的准确性和鲁棒性.

2 实例级重检测目标跟踪方法

在第 1 节各自分析了目前两种跟踪框架的各自局限性. 已有的局部-全局跟踪方法因重检测器、验证模块等局限性而难以应对目标频繁出现与消失、干扰物等挑战. 而全局跟踪方法大都难以利用上下文信息, 达不到实时性的要求. 本文算法基于局部-全局跟踪框架, 提出了一种利用多实例判别特征的全局搜索跟踪模块, 以 Faster R-CNN 检测模型为基础模型, 分别在 RPN 阶段与 R-CNN 阶段引入非局部操作和多级实例特征融合模块, 以充分挖掘目标实例级特征. 为了改进全局搜索跟踪模块的性能, 设计了双模板更新策略来提升跟踪器的鲁棒能力. 通过使用不同时间点上更新的模板能够更好地适应目标的变化. 根据局部或全局置信度分数判断目标是否存在, 并在下一帧中选择局部或全局搜索策略. 同时能够为局部跟踪模块提供目标的位置和大小. 此外还为全局搜索跟踪器引入了排名损失函数, 隐式学习了区域提议与原始查询目标的相似度.

2.1 整体跟踪流程

为了便于解释本文提出的跟踪方法, 假设跟踪序列中有 N 个帧, 取 I_t 为当前帧, 当前预测目标的坐标和置信值分别为 B_t 和 S_t. 单对象跟踪的任务可以描述为在所有的帧中引用静态模板和动态模板来搜索每一帧下的目标位置$\{B_t, t\in 1, 2, \cdots, n\}$的过程. 如图 1 所示, 整体框架结构包括一个负责初始定位和边界盒回归的局部跟踪模块, 以及一个具有模板更新机制的全局搜索跟踪模块. 前者负责在局部区域内进行精确跟踪与验证, 而后者侧重于在全局搜索状态下对整个图像进行全面搜索和验证, 以重新定位目标. 利用局部跟踪模块, 该跟踪器将在预定义的区域内启动对目标的搜索. 搜索状态是根据对象的置信度来确定的. 如果局部分数 $S_t^{(\text{local})}$ 超过预定义值阈值$\theta^{(\text{local})}$, 跟踪器继续进行局部跟踪, 确保在局部区域内的连续跟踪. 如果局部跟踪器认为目标已经消失, 那么它将触发全局搜索跟踪模式. 在全局搜索过程中, 首先应用模板更新方法来获得动态模板. 然后, 与搜索模板一起将其输入全局搜索跟踪模块中, 得到相应预测得分的预测结果, 并选择得分最高的一个作为当前帧的全局预测目标. 为了确定下一帧的搜索状态, 跟踪器将评估对象的置信度. 当置信度分数$S_t^{(\text{global})}$超过预定义的阈值$\theta^{(\text{global})}$时, 基于全局坐标$B_t^{(\text{global})}$执行局部跟踪. 否则, 全局搜索跟踪将持续在后续的帧中.

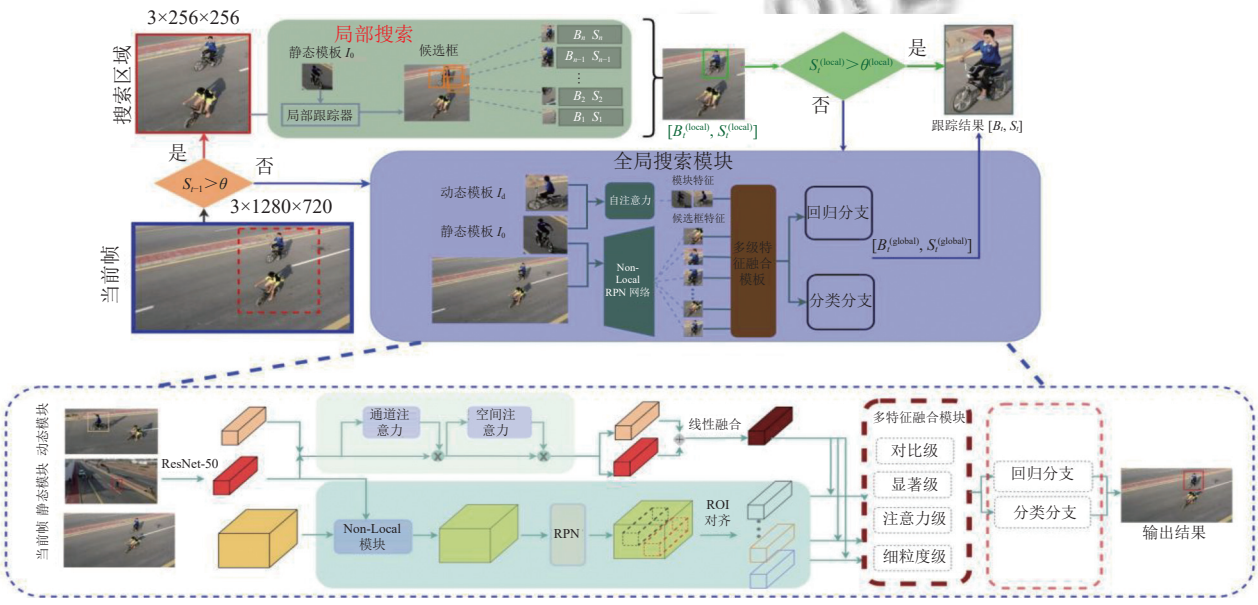


图 1 跟踪框架流程图

2.2 局部跟踪模块

本文的局部搜索模块使用的是 TransT 跟踪器, 它提出了一种基于 Transformer 的特征融合模型 (图 2 中红色框), 主要包含基于自注意力机制的 ECA 模块和基于互注意力机制的 CFA 模块. ECA 模块进行特征的上下文信息增强, CFA 模块对两个支路的特征进行融合. 如图 3 所示, ECA 模块通过自注意力机制增强特征上下文信息. 自注意力机制允许每个特征点在特征空间中考虑到其他所有特征点, 建立起长距离依赖关系. 这样, 即使部分特征由于遮挡等原因丢失, 跟踪器也能通过其他特征点重建出完整的对象表示. CFA 模块采用互注意力机制对两个不同支路的特征进行融合. 这种融合不是简单的线性组合, 而是通过全局、非线性的方式实现的, 这有助于跟踪器能够同时考虑两个特征集合之间的相互作用, 这样即使目标物体在尺度或角度上发生变化, 跟踪器也能自适应地调整特征表示, 保持跟踪的稳定性.

2.3 全局搜索跟踪模块

本文设计了一个新的全局搜索跟踪模块, 其灵感来自于专门用来跟踪目标的检测任务. 重新检测模块 包括一个 Non-Local RPN 模块, 它生成针对特定目标的自适应建议框, 以及 R-CNN 阶段的多级特征融合模块, 当局部跟踪不可信时, 该模块能全局上重新寻找目标, 能够独立地生成目标初始候选框, 并对其进行精确分类与回归. 分类是为了确定这些候选框中哪些可能包含目标, 而回归则是为了进一步精确定位跟踪目标的位置. 下面详细介绍了 Non-Local RPN 模块和多级特征融合模块, 以及应用在全局搜索跟踪模块中的 Ranking-Loss.

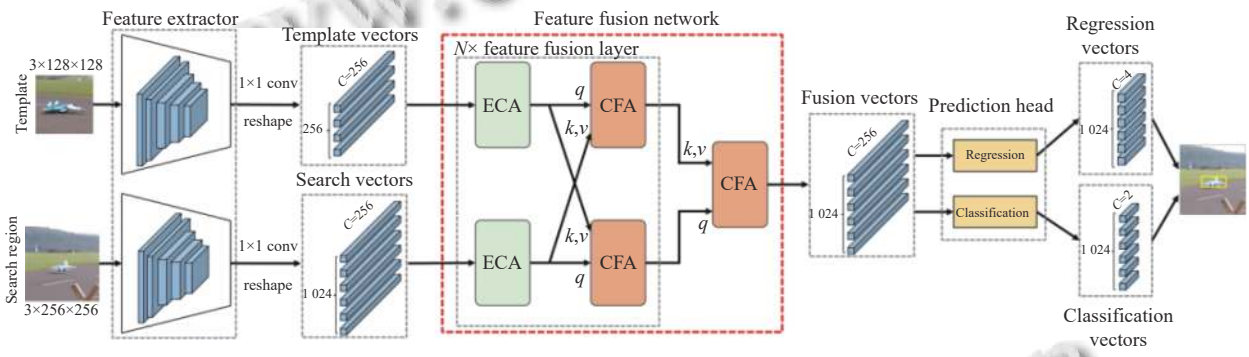


图 2 TransT 局部跟踪器流程图

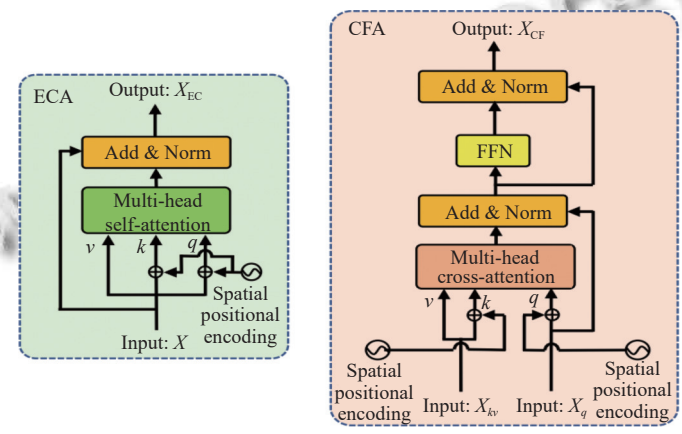


图 3 ECA 模块与 CFA 模块

2.3.1 Non-Local RPN 网络

当使用 Faster R-CNN 模型进行全局检测时, 主要目标是确保检测组件在目标跟踪中的适用性, 即保证 RPN 生成的候选框适合单目标跟踪. 回想一下, 传统 RPN 训练使用了每个图像中所有对象类上存在边界框的信息. 然而, 当 RPN 应用于单目标跟踪时, 只需要得到特定单一目标对应的标签和位置信息. 这就意味着, 如果 RPN 产生的候选框包含许多干扰物甚至背景, 那

么就会严重影响到后续候选框的进一步筛选. 为了解决这个问题, 通过非局部操作[29]来丰富搜索特征信息. 通过引入 Non-Local 模块 (如图 4)[30], 目的是在目标特征中获得广泛的依赖关系和上下文信息.

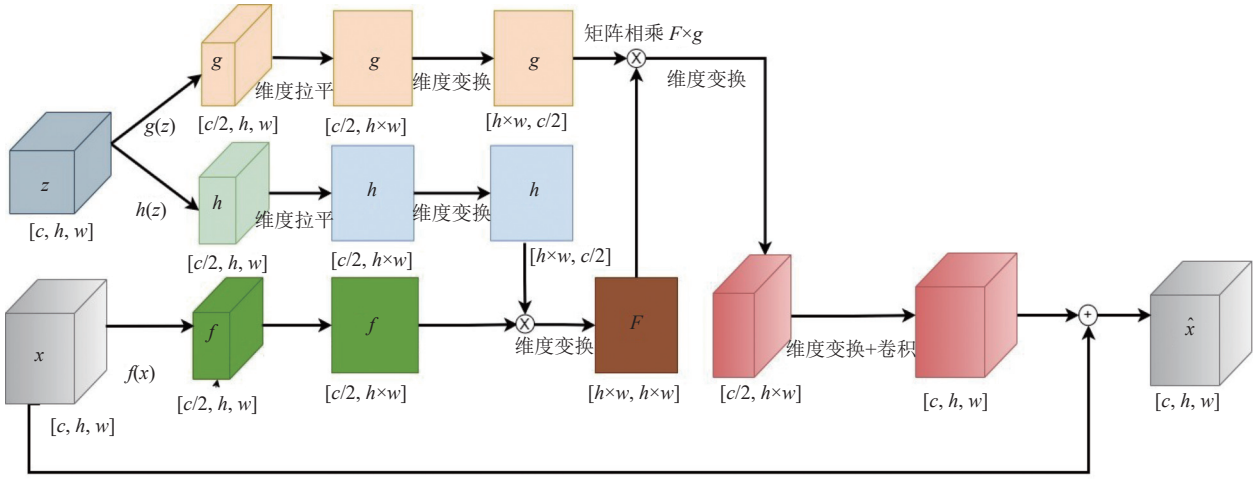


图 4 Non-Local 模块

通过目标信息来引导 RPN 能够更好地理解跟踪对象的外观和特征. 在该方法中, 将 $z \in R^{k\times k\times c}$ 表示查询对象的 ROI 特征, 而 $x \in R^{h\times w\times c}$ 表示搜索目标特征, 其中 h、w 和 k 表示特征大小. 为了将模板特征更好地融合到搜索图像特征中, 构建了一个特征嵌入过程. 具体来说模板特征 z 经过两个不同的卷积层得到不同特征映射 g 和 h, 搜索特征经过卷积得到特征映射 f. 然后将维度拉平与变换后的特征映射 h 与 f 进行相乘得到注意力权重矩阵 F. 再将权重矩阵与目标特征映射 g 相乘, 从而能够捕获目标特征中的实例信息. 再将这些信息与原搜索特征相加得到嵌入目标信息的搜索特征 $\hat{x} \in R^{h\times w\times c}$, 如式 (1) 所示:

$$\hat{x} = g_{\text{non-local}}(x,z) = x \oplus \psi(x,z) \tag{1}$$

其中, $\oplus$表示所有元素的求和, ψ 表示非局部运算. 基于模板特征设计的 Non-Local RPN 将学习从模板特征 z 中挖掘更多信息, 并生成特定目标的区域建议. 保持$\hat{x}$和 x 在相同分辨率大小的优点是, 传统 Faster R-CNN 模型可以直接调用 RPN 模块来生成候选框. 与其相同的是也使用相同的 RPN 损失来训练 Non-Local RPN 网络, 其中 L_{cls} 和 L_{loc} 被实现为 Binary Cross Entropy Loss 和 Smooth L1 Loss. Non-Local RPN 网络的联合 Loss 为:

$$\begin{aligned} &L_{\text{non-local}}(x,z) = L_{\text{rpn}}(x) \\ &= \frac{1}{N_{\text{cls}}}\sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda \frac{1}{N_{\text{loc}}} p_i^* \sum_i L_{\text{loc}}(s_i, s_i^*) \end{aligned} \tag{2}$$

其中, p_i 和 s_i 代表第 i 个候选结果的对应分数和预测的位置, 而 p_i^* 和 s_i^* 代表真实框. 损失权重系数值 λ 为 1.

2.3.2 多级特征融合模块

仅使用 Non-Local RPN 生成的建议框是不够的. 对于单目标跟踪来说, 这些生成后建议框仍然存在着干扰物. 当面对无法区分的干扰物时, 产生的最终候选框可能是无效的. Fan 等人[31]提出了一种多关系检测器来建模不同的关系. 然而, 这种多关系检测器有两个缺点. 1) 对于每个关系头, 它得到了关系特征的全局表示, 从而导致了关系特征的空间信息丢失. 2) 它只是添加由不同关系头生成的分类分数, 而不是融合关系特征. 因此, 该方法对分类任务是有用的, 但不适用于定位任务. 与之不同的是, 本文所提出的多级特征融合模块消除了上述缺点, 并采用了分层的方式来全面描述目标与搜索特征的语义关系. 需要注意的是, 这个过程是特定于被跟踪的目标的. 该模块的架构如图 5 所示, 同时编码了在对比级、显著级、注意力级和细粒度级上的特征. 然后将这些分支特征进行特别融合, 以增强目标特征的识别能力.

对比级关系: 为了计算全局特征和局部特征之间的关系, 首先加入了一个对比级的关系分支. 这个分支侧重于比较模板映射向量与候选特征的每个元素. 使用减法操作将查询向量与目标特征的每个位置进行比较, 这种直接的方法清楚地显示了两者之间的对比性关系. 产生的对比特征如下:

$$x_c^i = Conv_{c/2}(|R(p(x_i)) - z|) \tag{3}$$

其中, |·|是一个绝对值的运算符. $R(\cdot)$ 是一个广播操作. $p(\cdot)$ 表示平均池化操作, $Conv_{c/2}$ 为 1×1 卷积层, 输出通道数为输入通道数的一半, $z\in R^{k\times k\times c}$ 表示模板特征, $x_i\in R^{k\times k\times c}$ 表示第 i 个候选特征.

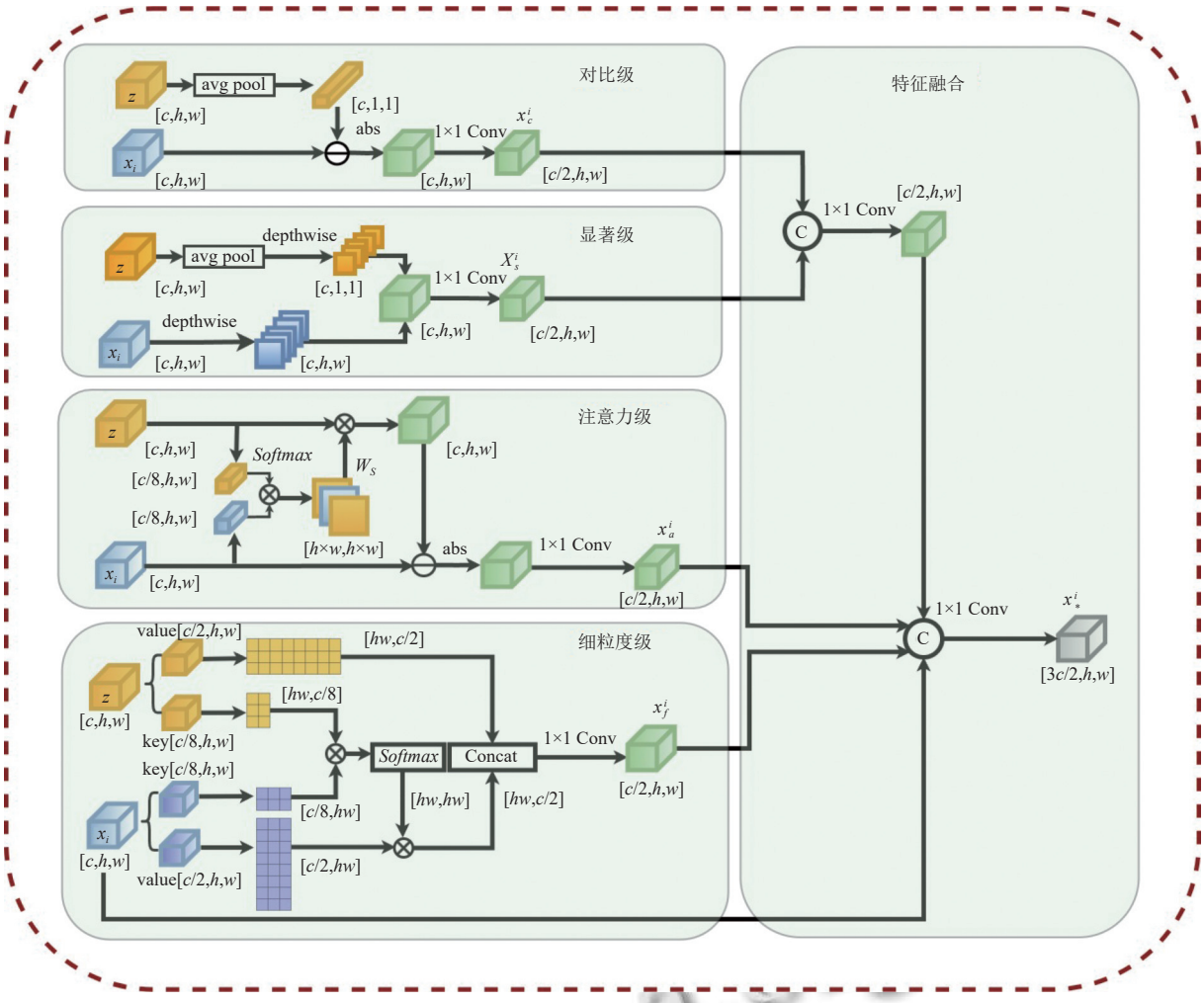


图 5 多级特征融合模块示意图

显著级关系: 为了在实例级上更好地捕获显著性关系[32], 该模块加入一个显著级的关系分支. 该方法将模板向量视为卷积核, 并从目标特征中深入提取关系特征. 模板向量与目标特征进行深度卷积, 并将其与目标映射特征结合起来, 得到全局模板特征. 这种方法有效地保存了丰富的目标特定的信息. 由此得到的显著级特征可以表示如下:

$$x_s^i = Conv_{c/2}(\phi(p(x_i),z)) \tag{4}$$

其中, $\phi(\cdot)$ 表示深度级卷积运算. $p(\cdot)$ 表示平均池化操作, $Conv_{c/2}$ 为 1×1 卷积层, 输出通道数为输入通道数的一半, $z\in R^{k\times k\times c}$ 表示模板特征, $x_i\in R^{k\times k\times c}$ 表示第 i 个候选特征.

注意级关系: 本文提出了一个注意级的分支来学习更具体的局部信息. 但查询特征与目标特征之间的局部比较可能会产生空间错位的问题. 为了缓解这个问题, 这里采用了交叉注意生成了空间感知的模板特征. 具体来说, 卷积层 $Conv_{c/8}$ 生成两组嵌入特征, 一组用于模板特征, 另一组用于候选目标特征. 它们产生的空间相似性是通过使用这些嵌入特征的矩阵乘积来计算的. 然后, 对相似度矩阵进行 *Softmax* 函数处理. 然后将得到的权重矩阵与模板特征按元素方向相乘, 然后从目标特征中减去, 得到它们的局部信息关系. 输出关系可以表示如下:

$$x_a^i = Conv_{c/2}(|(Softmax((Conv_{c/8}(x_i))^{\mathrm{T}} Conv_{c/8}(z)))\times z - x_i|) \tag{5}$$

其中, $Conv_{c/2}$ 为 1×1 卷积层, 输出通道数为输入通道数的一半, $z\in R^{k\times k\times c}$ 表示模板特征, $x_i\in R^{k\times k\times c}$ 表示第 i 个候选特征, $Conv_{c/8}$ 的输出通道数为输入通道数的 1/8.

细粒度级关系: 本文还通过细粒度级别分支从目标特征中提取到像素级的相关信息. 具体来说, 它使用两个并行的 3×3 卷积层来降低输入特征的维数. 对于目标特征, 键值对是输出特征: $K_q\in R^{c/8\times w\times h}$, $V_q\in R^{c/2\times k\times k}$. 对于模板特征, 输出为 $K_t\in R^{c/8\times w\times h}$, $V_t\in R^{c/2\times k\times k}$, K 表示帮助确定目标位置的相似度矩阵. 而 V 则用于目标细节识别. 基于 key-value 映射上, 以 Non-Local 的方式逐像素计算相似度. 然后应用 *Softmax* 函数得到最终的权值 W_f:

$$W_f = Softmax(\varphi(K_q)^{\mathrm{T}}\varphi'(K_t)) \tag{6}$$

其中, φ、φ' 是不同的映射空间. 然后, 利用得到的权值矩阵 W_f 对模板特征值 V_t 进行加权, 然后将其与目标特征值 V_q 进行连接. 因此, 最终的输出为:

$$x_f^i = Conv_{c/2}(Concat(V_q, W_f\times V_s)) \tag{7}$$

在获得这 4 个关系特征后, 第 1 步是合并全局特征 features x_s^i 是和 x_c^i, 使用 $Conv_{c/2}$ 来降低计算量. 然后将合并后的特征连接到局部特征 x_a^i 和 x_f^i 上, 以获得通道 $3c/2$ 的目标实例特征. 最后, 通过将实例特征与候选特征 x^i 沿通道维度连接, 得到一个尺寸为 $5c/2$ 的映射特征, 最后通过 $Conv_{3c/2}$ 得到嵌入特征 x_*^i.

$$x_*^i = Conv_{3c/2}([Conv_{c/2}([x_s^i, x_c^i], x_a^i, x_f^i, x^i]) \tag{8}$$

通过提取嵌入目标多级实例特征的候选框 x_*^i, 模型继续对其进行分类和定位, 最终得到预测对象. 针对模型的分类与回归训练仍然使用 Binary Cross Entropy Loss 和 Smooth L1 Loss. 整个模块损失计算如下:

$$L_{\text{relation}}(z,x) = \frac{1}{N_p}L_{\text{rcnn}}(x^i) \tag{9}$$

其中, N_p 是候选框数量.

$$L_{\text{rcnn}}(x^i) = L_{\text{cls}}(p_i, p_i^*) + \lambda p_i^* L_{\text{loc}}(s_i, s_i^*) \tag{10}$$

其中, p_i 和 s_i 是候选目标的置信度分数和位置, λ 被设置为 1, p_i^* 和 s_i^* 表示真实框.

2.3.3 排名损失 (Ranking-Loss)

本文通过在模型学习过程中加入 Ranking-Loss 来提高全局搜索跟踪器区分目标和干扰的能力. 在 Non-Local RPN 阶段, 选择 K 个区域作为候选框 (实验中的 K=256). 与此同时设计了一个两层的 MLP 网络 M, 其最后一层是一个双向 *Softmax*. 在训练阶段, 首先根据它们与目标框真实值的 IoU 值是否大于 0.5, 将 K 个候选提议标记为前景 (标签 1) 或背景 (标签 0). 然后考虑基于边界的排名损失, 以隐式地学习理想的度量, 使得与查询 p 最相关的提议出现在排名列表的顶部. 为此将每个候选框的特征向量 x^i 与目标特征向量 z 连接起来, 得到一个组合向量, 记为 $p=[(x^i)^{\mathrm{T}}\mid z^{\mathrm{T}}]^{\mathrm{T}}\in R^{2C}$, 其中 y 表示其标签, 如果 x^i 对应于前景提议, 则 y 为 1, 否则为 0. 本文选择以 $2c\to 8\to 2$ 的层维度分布来构建 M. 现在让 $s=M(p)$ 表示 M 相对于查询 q 预测的前景概率. 定义此排名损失为:

$$L_{\text{MR}}(\{x_i\}) = \sum_{i=1}^{K} y_i\times\max\{m^+ - s_i, 0\} + (1-y_i) \times\max\{s_i - m^-\} + \Delta_i \tag{11}$$

$$\Delta_i = \sum_{i=1}^{K}[y_i = y_j]\times\max\{|s_i - s_j| - m^-, 0\} + [y_i\neq y_j] \times\max\{m^+ - |s_i - s_j|, 0\} \tag{12}$$

其中, $[\cdot]$为艾弗森括号, s 为 R-CNN 阶段的置信分数, m^+是预测目标的期望概率下界, m^-是预测背景与干扰物的期望概率上界, 分别设置为 0.7 和 0.3. 全局搜索跟踪模型的联合损失可以表示为:

$$L = L_{\text{non-local}} + L_{\text{relation}} + \lambda L_{\text{MR}} \tag{13}$$

其中, 损失权重系数 λ 设为 3. $L_{\text{non-local}}$ 和 L_{relation} 分别代表 Non-Local RPN 损失与多级实例特征的 R-CNN 损失.

2.4 模板更新

困扰大多数长时追踪器的主要挑战是, 目标仍在不断变化. 为了解决这一挑战, 算法通过实现双模板策略来增强全局搜索跟踪模块. 图 6 显示了模板更新架构的方案. 除了静态模板 I_0 外, 算法还在推理过程中选择每个阶段的动态帧, 并在 R-CNN 阶段获得它们的置信度分数. 选择分数最高的作为动态模板, 以便在下一阶段更新动态模板. 将动态模板图像裁剪输入网络, 得到相应的动态模板特征, 通过给定的参数 ω 将所得特征 F_d 与初始模板特征 F_0 进行插值, 每段时间后, 融合的模板特征 F_T 将被更新:

$$F_T = (1-\omega)F_0 + \omega F_d \tag{14}$$

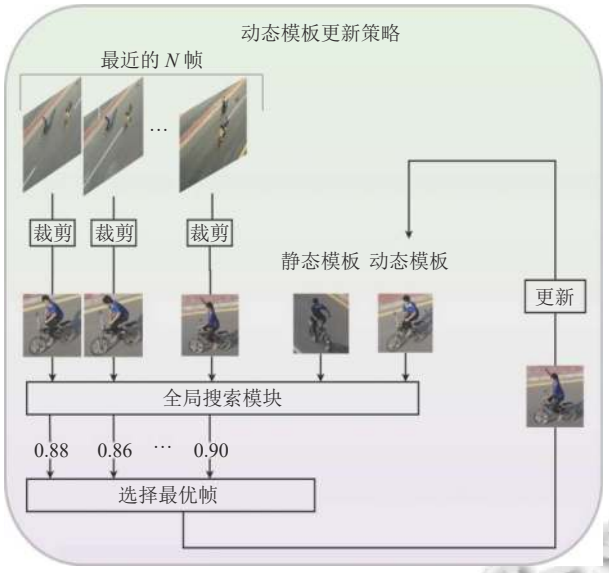


图 6 动态模板更新流程图

3 实验分析

3.1 实验训练数据

在本文中, 局部跟踪器使用了 4 个数据集进行训练, 分别是 TrackingNet[33]、COCO[34]、LaSOT[35]和 GOT-10k[36]. 全局搜索跟踪模块则以 COCO、GOT-10k 和 LaSOT 数据集为训练集, 采样率分别为 0.4、0.4 和 0.2. COCO 是一个包含超过 11.8 万张图像, 属于 80 个对象类. GOT-10k 和 LaSOT 是视觉跟踪数据集, 其中 GOT-10k 由 563 个对象类的 10 000 个视频组成, 而 LaSOT 由 70 个对象类的 1 400 个视频组成. 在训练过程中从 COCO 数据集当中随机抽取一对图像, 并对其进行数据增强, 生成一个图像对; 在 GOT-10k 和 LaSOT 数据集当中, 对视频进行采样帧对, 通过随机角度翻转和噪声干扰来增强上述图像对从而得到训练数据.

3.2 模型训练

对于局部搜索模块和全局搜索跟踪模块, 前者采用 AdamW 优化算法[29]进行训练, 训练具体细节请参考文献[27], 本文使用 SGD 优化算法训练后者, 并为动量和权重衰减设置 0.9 和 1×10^{-4}. 提出的全局搜索跟踪模型主干网络由 COCO 数据集上预先训练的 Faster R-CNN 初始化, 虽然所有其他参数都可以学习, 但保持批处理归一化参数不变. 该模型共有 12 个阶段的训练. 初始学习率定义为 0.01, 在第 8 阶段和第 11 阶段衰减系数为 0.1. 跟踪器是在一个标准的 PC 配置上使用 PyTorch 实现的, 具有一个 Inter i9 CPU (64 GB RAM) 和一个 NVIDIA GeForce GTX 2080 Ti GPU (11 GB RAM). 模型具备实时性, 处理速度大约为 35 帧每秒 (f/s).

3.3 实验结果与分析

3.3.1 VOT-LT 数据集实验结果

VOT-LT 系列数据集包含 35 个长期视频, 总共有 146 847 帧, 跟踪不同种类的对象. VOT2018-LT[37]数据集中的所有视频都涉及对象消失, 并且平均每个视频存在 40.6 帧的中断长度. 与许多其他长期数据集相比, 该数据集展示了更高频率的目标消失事件. 为了进一步评估性能, 还使用了 VOT2019-LT[17]数据集. 它引入了 15 个额外具有挑战性的视频, 并且评估标准与 VOT2018-LT 保持一致. 文献[37]还提出了 3 个衡量长期跟踪器性能的指标: 跟踪精度 (*TP*)、跟踪召回率 (*TR*) 和跟踪分数 *F*1-score (*F*1). *TP* 和 *TR* 用于确定综合评估指标 *F*1.

$$F(\tau_\theta)=\frac{2TP(\tau_\theta)TR(\tau_\theta)}{TP(\tau_\theta)+TR(\tau_\theta)} \tag{15}$$

本文围绕目标跟踪的精度、目标跟踪的召回率以及检测跟踪速度 3 个方面, 将本文提出的方法与其他长时跟踪器在多个数据集上进行比较. VOT2018-LT、VOT2019-LT 数据集中使用多种算法进行测试, 结果如表 1 和表 2 所示.

表 1 不同跟踪方法在 VOT2018-LT 长时跟踪数据集上的性能对比

跟踪算法	综合分数	准确率*Pr*	召回率*Re*	速度 (f/s)
Ours	0.724	0.742	0.706	35
KeepTrack	0.713	0.727	0.703	18
LTMU	0.690	0.710	0.672	13
Siam R-CNN	0.671	0.667	0.675	5
SiamRPN++	0.626	0.644	0.608	21

表 2 不同跟踪方法在 VOT2019-LT 长时跟踪数据集上的性能对比

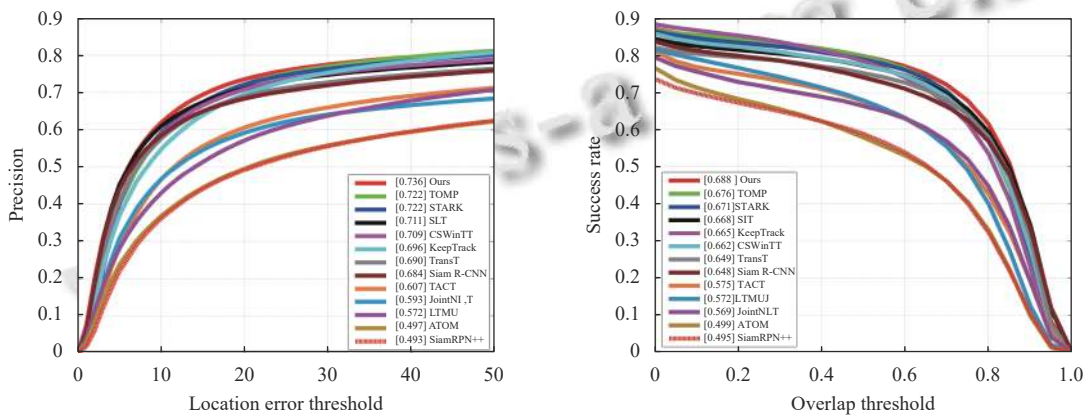
跟踪算法	综合分数	准确率*Pr*	召回率*Re*
Ours	0.714	0.731	0.697
KeepTrack_LT	0.712	0.725	0.700
LTMU	0.697	0.721	0.674
Siam R-CNN	0.664	0.654	0.673
MBMD	0.575	0.623	0.534

如表 1 所示, 说明了不同长时跟踪方法的比较结果, 包括 KeepTrack[38]、LTMU[39]、Siam R-CNN[40]、SiamRPN++[4]. 与目前性能非常出色的 KeepTrack 相比, 本文提出的算法跟踪指标 *F*1 高出 1.1%, 并且在速

度上几乎是 KeepTrack 的两倍左右. 无论是各方面性能还是实时性方面, 都明显优于 LTMU 与 Siam R-CNN 算法. 同样, VOT2019-LT 的可比性结果见表 2. 因此无论是在准确率、召回率还是跟踪速度性能上, 本文算法都具备出色的竞争力.

3.3.2 LaSOT 数据集实验结果

同时本文也在 LaSOT 数据集上进行对比, LaSOT 是一个重要的、被广泛认可的单对象跟踪数据集. 该测试集包括 280 个视频, 每个视频的平均帧长为 2 500. 这就对被测试的跟踪器提出了更高的要求, 例如具有很强的处理被遮挡和被遮挡物体的能力. 评估结果被报告在测试集上. 如图 7 所示, 与没有配备的局部跟踪器 TransT 相比, 本文算法跟踪成功率提高了 3.8%, 精度也提高了 4.4%. 此外, 本文算法性能上也优于 STARK[24], 成功率提高了 1.6%, 精度提高了 1.3%. 与其他 9 个跟踪器相比, 本文算法在成功和准确性方面都取得了最好的性能. 为了更好地展现本文提出的跟踪器性能, 下面也对跟踪结果进行可视化, 如图 8 所示.



(a) 准确率 (b) 成功率

图 7 不同方法在 LaSOT 数据集上的准确率和成功率

图 8 在 LaSOT 数据集上可视化跟踪结果

3.4 消融实验

本节对提出的跟踪框架进行全面分析, 探索其各种关键组成部分. 本文跟踪器包括 3 个基本组件: 局部跟踪器、全局搜索跟踪器和双模板更新策略. 为了评估各种组件的性能, 本文实现了以下变体. 表 3 显示, 模板更新策略和全局搜索跟踪器对总体性能都有显著影响. 在 VOT2018-LT 数据集上的性能表明, 每增加一个模块, 算法的性能就能得到提升. 实验证明了全局搜索跟踪器与模板更新模块的有效性.

表 3 在 VOT2018-LT 数据集上不同跟踪组件的性能对比

局部跟踪	全局搜索跟踪	模板更新	准确率*Pr*	召回率*Re*	综合分数*F*1
√	—	—	0.720	0.642	0.679
√	√	—	0.740	0.665	0.700
√	√	√	0.742	0.706	0.724

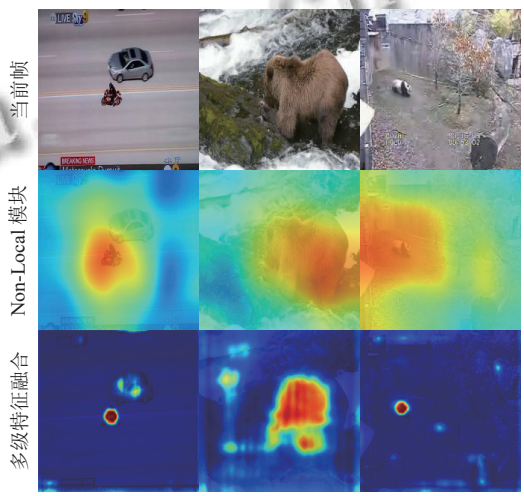
其次对全局搜索跟踪器进行一系列的消融实验, 为了评估它们的表现, 本文进行了一系列的实验. 首先采用了全局搜索跟踪器作为单独的跟踪器, 并在 OTB-2015 数据集上评估了它的性能. 在表 4 中的结果表明, 当全局搜索跟踪器单独配备 Non-Local RPN 模块时, 其准确率只有 27.5%, 而成功率也仅为 35.6%. 同样, 当仅配备多级特征融合模块时, 准确率提高到 52.5%, 成功率提高到 70.7%. 然而, 这两个模块的组合可以进一步提高性能, 成功率为 64.95%, 准确率为 81.8%. 这些实验结果强调了 Non-Local RPN 模块和多级特征融合模块在本文算法中所起的关键作用. 此外 Ranking-Loss 损失对整体性能的提高也有一定的贡献.

表 4 针对全局搜索跟踪器的消融实验 (%)

Non-Local RPN 模块	多级特征融合模块	Ranking-Loss损失	成功率	准确率
√	—	—	27.5	35.6
√	—	√	30.2	38.9
—	√	—	52.5	70.7
√	√	—	64.9	81.8
√	√	√	65.9	82.8

为了更好地展示 Non-Local RPN 模块与多级特征融合模块在特征提取方面的效果, 在图 9 中展示了一些由非局部 RPN 和多级特征融合模块学习的特征映射. 这些特征映射来自算法模型的中间层. 可以看到, 互相非局部的特征使得 RPN 能够生成更多优先考虑目标和模板感兴趣区域的建议, 从而提供共同关注的效果. 融合关系特征可以充分利用目标的独特特征, 因此能够全面地提供由关系模块生成的具有区分性的语义线索.

为了验证多级特征融合模块中所提出的 4 种分支特征的有效性, 本文在 OTB-100 短时跟踪数据集上进行了消融实验. 把全局搜索跟踪器当作成一个单独的全局跟踪器并且分析了各个子模块对跟踪性能的影响. 在表 5 中, 前 5 行显示了单独使用分层模块可以产生有效但有限的性能. 虽然采用单一分层模块会带来一些好处, 但它也可能会引入相似性偏差. 表 5 中的第 6–8 行表明, 随着包含了更多的关系特性, 跟踪变得更加健壮和准确. 当使用所有关系模块时, 可以获得最佳的性能, 这展示了多级特征融合模块的有效性.



(a) 摩托车 (b) 北极熊 (c) 熊猫

图 9 目标各模块特征图可视化

表 5 不同子模块对跟踪性能产生的影响 (%)

对比模块	显著模块	注意模块	细粒模块	成功率	准确率
—	—	—	—	30.2	38.9
√	—	—	—	61.3	75.6
—	√	—	—	62.0	76.6
—	—	√	—	61.5	76.3
—	—	—	√	61.7	77.0
√	√	—	—	64.3	81.0
√	√	√	—	65.0	81.8
√	√	√	√	65.9	82.8

同时为了能够给全局搜索跟踪模块提供一个全面的评价, 本文战略性地将其集成到其他局部跟踪器中, 并取代了一些检测跟踪方法的重检测模块. 然后继续使用 VOT2018-LT 数据集重新评估这些跟踪器的性能. 评估的重点是评估改进后的跟踪器 (即 SiamMask[5]、SiamRPN++ [4]、Ocean[41]) 在长时跟踪场景中的跟踪能力. 可以从表 6 看出, 传统的短时跟踪算法在长时跟踪场景中表现不佳, 当分别配备本文提出的全局检测器后, 性能分别得到了巨大的提升, 如 SiamMask, *F*1 指标从之前的 49.9% 提升到 64.8%, SiamRPN++的

*F*1 从 49.5% 增加到 64.3%. 在已有的检测跟踪方法 ELGLT 中将原有的重检测器替换成本文提出的全局搜索跟踪模块, 从结果可以看出召回率增加了 2.4% 的性能提升. 实验结果再一次证实了所提出的全局搜索跟踪方法在提高评估跟踪器的性能方面的有效性 ("_R"代表将原有的检测器替换成本文提出的全局搜索跟踪模块).

表 6 集成其他跟踪器所产生的性能对比

跟踪算法	综合分数*F*1	准确率*Pr*	召回率*Re*
SiamRPN++	0.495	0.622	0.411
SiamMask	0.499	0.633	0.412
Ocean	0.521	0.529	0.513
ELGLT	0.638	0.669	0.610
SiamRPN++_R	0.643	0.653	0.633
SiamMask_R	0.648	0.636	0.642
Ocean_R	0.665	0.667	0.663
ELGLT_R	0.651	0.670	0.634

4 结论与展望

本文以目标检测框架为基础开发了一种新的实例级全局搜索跟踪器, 分别在 RPN 阶段和 R-CNN 阶段加入 Non-Local 模块和多级特征融合模块, 借此对目标进行不同实例级特征的挖掘, 以此获得质量极高的候选目标, 整个跟踪框架在最大程度上利用了上下文信息的同时也可以有效地避免跟踪漂移. 该全局搜索跟踪器也可以灵活地与其他短时跟踪器集成. 这种创新的跟踪组件不仅准确地确定了丢失目标的位置和大小, 而且可以实现跟踪策略的合理切换. 此外, 本文还设计了高效的双模板更新策略和引入了排名损失来增强其全局搜索跟踪能力. 为了评估本文提出长时跟踪框架的性能, 在多个广泛使用的跟踪基准上进行了实验. 实验结果不可否认地证明了该跟踪框架对于成功实现长时目标跟踪的有效性. 可以将其用于解决目标追踪困难、目标频繁消失以及外观严重变化等问题. 但该跟踪框架仍然具有一定局限性, 即对于目标完全被遮挡的情况并不能很好应对. 因此接下来会尝试加入目标轨迹预测模块来一定程度缓解目标完全丢失的状况.

参考文献

1 卢湖川, 李佩霞, 王栋. 目标跟踪算法综述. 模式识别与人工智能, 2018, 31(1): 61–76. [doi: 10.16451/j.cnki.issn1003-6059.201801006]

2 Grishchenko I, Ablavatski A, Kartynnik Y, *et al*. Attention mesh: High-fidelity face mesh prediction in real-time. arXiv:2006.10962, 2020.

3 Li B, Yan JJ, Wu W, *et al*. High performance visual tracking with Siamese region proposal network. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8971–8980.

4 Li B, Wu W, Wang Q, *et al*. SiamRPN++: Evolution of Siamese visual tracking with very deep networks. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 4277–4286.

5 Wang Q, Zhang L, Bertinetto L, *et al*. Fast online object tracking and segmentation: A unifying approach. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 1328–1338.

6 Zhang YH, Wang D, Wang LJ, *et al*. Learning regression and verification networks for long-term visual tracking. arXiv:1809.04320, 2018.

7 Yan B, Zhao HJ, Wang D, *et al*. 'Skimming-perusal' tracking: A framework for real-time and robust long-term tracking. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 2385–2393.

8 Zhao HJ, Yan B, Wang D, *et al*. Effective local and global search for fast long-term tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(1): 460–474. [doi: 10.1109/TPAMI.2022.3153645]

9 Huang LH, Zhao X, Huang KQ. GlobalTrack: A simple and strong baseline for long-term tracking. Proceedings of the 34th AAAI Conference on Artificial Intelligence. New York: AAAI, 2020. 11037–11044.

10 Zhou L, Zhou ZK, Mao KG, *et al*. Joint visual grounding and tracking with natural language specification. Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver: IEEE, 2023. 23151–23160.

11 Bertinetto L, Valmadre J, Henriques JF, *et al*. Fully-convolutional Siamese networks for object tracking. Proceedings of the 2016 European Conference on Computer Vision. Amsterdam: Springer, 2016. 850–865.

12 Zhang ZK, Zhong BN, Zhang SP, *et al*. Distractor-aware fast tracking via dynamic convolutions and MOT philosophy. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 1024–1033.

13 Li ZB, Wang Q, Gao J, *et al*. Globally spatial-temporal perception: A long-term tracking system. Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP). Abu Dhabi: IEEE, 2020. 2066–2070.

14 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 91–99.

15 Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-

detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409–1422. [doi: 10.1109/TPAMI.2011.239]
16 Ma C, Yang XK, Zhang CY, *et al*. Long-term correlation tracking. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2015. 5388–5396.
17 Kristan M, Matas J, Leonardis A, *et al*. The seventh visual object tracking VOT2019 challenge results. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshops. Seoul: IEEE, 2019. 2206–2241.
18 Valmadre J, Bertinetto L, Henriques JF, *et al*. Long-term tracking in the wild: A benchmark. Proceedings of the 15th European Conference on Computer Vision (ECCV). Munich: Springer, 2018. 692–707.
19 Zhu G, Porikli F, Li HD. Beyond local search: Tracking objects everywhere with instance-specific proposals. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 943–951.
20 Platt JC. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report, MSR-TR-98-14, Microsoft Research, 1998.
21 Dave A, Tokmakov P, Schmid C, *et al*. Learning to track any object. arXiv:1910.11844, 2019.
22 He KM, Gkioxari G, Dollár P, *et al*. Mask R-CNN. Proceedings of the 2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017. 2980–2988.
23 Xu YH, Ban YT, Alameda-Pineda X, *et al*. DeepMOT: A differentiable framework for training multiple object trackers. arXiv:1906.06618, 2019.
24 Yan B, Peng HW, Fu JL, *et al*. Learning spatio-temporal transformer for visual tracking. Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision. Montreal: IEEE, 2021. 10428–10437.
25 Carion N, Massa F, Synnaeve G, *et al*. End-to-end object detection with Transformers. Proceedings of the 16th European Conference on Computer Vision. Glasgow: Springer, 2020. 213–229.
26 Song ZK, Yu JQ, Chen YPP, *et al*. Transformer tracking with cyclic shifting window attention. Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 8781–8790.
27 Wang N, Zhou WG, Wang J, *et al*. Transformer meets tracker: Exploiting temporal context for robust visual tracking. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 1571–1580.
28 Kim M, Lee S, Ok J, *et al*. Towards sequence-level training for visual tracking. Proceedings of the 17th European Conference on Computer Vision. Tel Aviv: Springer, 2022. 534–551.
29 Loshchilov I, Hutter F. Decoupled weight decay regularization. Proceedings of the 7th International Conference on Learning Representations. New Orleans: OpenReview.net, 2019.
30 Wang XL, Girshick R, Gupta A, *et al*. Non-local neural networks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 7794–7803.
31 Fan Q, Zhuo W, Tang CK, *et al*. Few-shot object detection with attention-RPN and multi-relation detector. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 4012–4021.
32 Liu Y, Zhang Q, Zhang DW, *et al*. Employing deep part-object relationships for salient object detection. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 1232–1241.
33 Müller M, Bibi A, Giancola S, *et al*. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. Proceedings of the 15th European Conference on Computer Vision (ECCV). Munich: Springer, 2018. 310–327.
34 Lin TY, Maire M, Belongie S, *et al*. Microsoft COCO: Common objects in context. Proceedings of the 13th European Conference on Computer Vision. Zurich: Springer, 2014. 740–755.
35 Fan H, Lin LT, Yang F, *et al*. LaSOT: A high-quality benchmark for large-scale single object tracking. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 5369–5378.
36 Huang LH, Zhao X, Huang KQ. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(5): 1562–1577. [doi: 10.1109/TPAMI.2019.2957464]
37 Lukežič A, Zajc LČ, Vojíř T, *et al*. Now you see me: Evaluating performance in long-term visual tracking. arXiv: 1804.07056, 2018.
38 Mayer C, Danelljan M, Paudel DP, *et al*. Learning target candidate association to keep track of what not to track. Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision. Montreal: IEEE, 2021. 13424–13434.
39 Dai KN, Zhang YH, Wang D, *et al*. High-performance long-term tracking with meta-updater. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 6297–6306.
40 Voigtlaender P, Luiten J, Torr PHS, *et al*. Siam R-CNN: Visual tracking by re-detection. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 6577–6587.
41 Zhang ZP, Peng HW, Fu JL, *et al*. Ocean: Object-aware anchor-free tracking. Proceedings of the 16th European Conference on Computer Vision. Glasgow: Springer, 2020. 771–787.

(校对责编: 孙君艳)