基于特征自适应选取的视觉目标跟踪算法

王彩霞[1,2],安 琪[1*],周鸿策[1],李义鹏[1]
(1. 长春理工大学电子信息工程学院,吉林长春 130022;2. 吉林省光电检测与智能信息处理工程技术研究中心,吉林长春 130022)

摘 要: 针对现有视觉目标跟踪算法始终选择所有的历史模板与全部搜索区域进行交互,导致有强背景干扰或者目标发生形变时产生的跟踪失败问题,提出一种基于特征自适应选取的视觉目标跟踪算法. 首先,通过模板特征过滤器将传统图像级模板更新优化为特征级动态更新,筛选当前帧的强相关模板特征并压缩弱相关特征,减少冗余信息干扰;其次,采用搜索特征鉴别器自适应划分搜索区域中潜在的目标特征与噪声特征,抑制无关区域的交互;最后,引入时空信息传播令牌,跨帧传递目标外观与位置信息,逐帧修正跟踪响应;设计基于分离注意力机制的特征交互编码器,将自注意力与交叉注意力分离,适配上述模块并增强判别能力. 在多种大规模公开数据集上的实验取得了鲁棒结果,在OTB100、LaSOT和UAV123数据集上的精度分别达到93.0%、79.6%和91.2%,且算法能够实现跟踪成功率与跟踪速度的良好平衡,提升了跟踪器在复杂场景下的准确性和鲁棒性.

关键词: 计算机视觉;目标跟踪;特征自适应选取;模板特征过滤器;搜索特征鉴别器;时空信息传播令牌
基金项目: 国家重点研发计划项目(No.2018YFB1107600);吉林省科技厅项目(No.20210201021GX)
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112(2025)08-2879-20
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20250046

Visual Object Tracking Algorithm Based on Adaptive Feature Selection

WANG Cai-xia[1,2], AN Qi[1*], ZHOU Hong-ce[1], LI Yi-peng[1]
(1. *School of Electronic Information and Engineering*, *Changchun University of Science and Technology*, *Changchun*, *Jilin* 130022, *China*;
2. *Jilin Province Technology Research Center of Photoelectric Detection and Intelligent Information Processing*, *Changchun*, *Jilin* 130022, *China*)

Abstract: To address the persistent tracking failures caused by strong background interference or target deformation in existing visual object tracking algorithms that indiscriminately utilize all historical templates and interact with entire search regions, this paper proposes a feature-adaptive selection based visual object tracking algorithm. First, a template feature filter is introduced to optimize traditional image-level template updating into feature-level dynamic updating, which selectively preserves strongly correlated template features while compressing weakly relevant features to reduce redundant information interference. Second, a search feature discriminator is employed to autonomously distinguish potential target features from noise features in search regions, thereby suppressing interactions with irrelevant areas. Furthermore, spatio-temporal information propagation tokens are incorporated to transmit target appearance and positional information across frames for progressive response refinement. A feature interaction encoder based on decoupled attention mechanisms is designed, which separates self-attention and cross-attention operations to better adapt to the proposed modules while enhancing discriminative capabilities. Comprehensive experiments on multiple large-scale public datasets demonstrate robust performance, achieving precision scores of 93.0%, 79.6%, and 91.2% on OTB100, LaSOT, and UAV123 datasets respectively. The algorithm maintains an optimal balance between tracking success rate and operational efficiency, significantly improving tracking accuracy and robustness in complex scenarios.

Key words: computer vision; object tracking; adaptive feature selection; template feature filter; search feature discriminator; spatio-temporal information propagation tokens

Foundation Item(s): National Key Research and Development Program of China (No.2018YFB1107600); Jilin Provincial Department of Science and Technology Project (No.20210201021GX)

收稿日期:2025-01-11;录用日期:2025-06-10;责任编辑:覃怀银
*通讯作者:安琪

1 引言

视觉目标跟踪任务在人机交互[1]、航空制导[2]、自动驾驶[3]等实际场景下具有广泛应用,近年来成为计算机视觉领域的热点研究方向. 根据待跟踪目标的个数,视觉目标跟踪可以划分为多目标跟踪和单目标跟踪,本文聚焦于单目标跟踪领域展开研究.

目前的单目标跟踪[4-6]大都采用孪生神经网络对初始帧框选的待跟踪目标及后续搜索帧进行特征提取,再应用深度互相关或Transformer进行特征交互,得到后续分类前背景与回归位置的相关响应. 其中,SiamFC(Siamese Fully-Convolutional)[7]算法作为基于深度孪生网络跟踪算法的开山之作,将跟踪转换为模板与搜索区域之间的相似性匹配问题,实现了准确率与速度的较好平衡;随后的SiamRPN++(Siamese Region Proposal Network++)[8]进一步提出了深度互相关策略,极大提升了孪生子网的深度;此外,SiamCAR(Siamese fully convolutional Classification And Regression for visual tracking)[9]、SiamBAN(Siamese Box Adaptive Network)[10]等大量无锚跟踪器也是在此框架下的代表工作. 尽管深度特征的利用显著提升了跟踪性能,但始终选择初始帧的模板特征与搜索区域的所有部分进行交互,当发生遮挡、形变、相似性干扰等实际挑战时会引入错误的噪声区域交互,导致剧烈的跟踪漂移和目标失跟.

考虑到跟踪的时域连续性,解决这一问题的一种思路是高效利用时间上下文信息进行动态建模来弥补初始帧模板特征的局限. ATOM(Accurate Tracking by Overlap Maximization)[11]能够根据新的观测数据在线更新模型参数,使跟踪器能够适应目标外观的变化;AutoMatch[12]通过在线学习和反馈机制自动调整关键参数,以适应跟踪的动态变化;TransT(Transformer Tracking)[13]通过引入Transformer架构,高效利用帧序列间的连续时域特征;ODTrack(OnLine Dense temporal token learning for visual Tracking)[14]以在线令牌传播的方式利用历史特征指导特征建模;GlobalTrack[15]使用全序列作为搜索区域,有效整合目标的历史信息;STARK(Spatio-Temporal trAnsfoRmer network for visual tracKing)[16]和Mixformer[17]设计了一个分数头,通过选取高置信度的模板结果进行在线实时更新;AIATrack(Attention In Attention for transformer visual Tracking)[18]引入了前背景嵌入,分别针对长时和短时信息进行目标建模. 然而,上述算法忽略了不同时刻搜索区域所需的历史信息差异,使用大量不相干时域特征会不可避免在特征交互中引入错误响应;此外,受限于模板池的内存限制,传统的更新机制都是以先入先出的方式抛弃早期模板,在遮挡、形变、视角旋转等挑战下会因信息缺失而导致表现不佳.

通常搜索区域背景相对占比较大,解决这一问题的另一种思路是稀疏背景的特征表示,减轻噪声区域的交互. SiamGAT(Siamese Graph Attention Tracking)[19]采用图注意力网络来处理模板图和搜索图之间的特征,更准确地聚焦目标特征并抑制背景噪声;OSTrack(One Stream Tracking)[20]通过多层编解码网络逐步消除噪声令牌,减轻背景干扰;SparseTT(visual Tracking with Sparse Transformers)[21]缓解了自注意力机制中上下文信息过度聚焦,提升搜索帧潜在区域的关注度;GRM(Generalized Relation Modeling for transformer tracking)[22]通过可学习区域令牌分离机制,减轻搜索区域错误的交叉关系建模;BANDT(Border-Aware Network with Deformable Transformers for visual tracking)[23]采用可变形Transformer,通过计算目标周围的特定位置注意力,减轻了噪声区域交互;CSWinTT(Transformer Tracking with Cyclic Shifting Window attention)[24]通过多尺度循环移位窗口机制,更好地捕捉了搜索帧目标前景的完整性. 尽管上述算法取得了一定进展,但这些方法大多在静态背景的场景下表现较好,当面临不同时刻复杂背景时,稀疏的背景表示依然不够精确;并且在上述算法均是在特征交互时逐层进行背景稀疏,容易建立错误的特征关联,导致跟踪器混淆前景与背景.

针对上述问题,本文提出一种基于特征自适应选取的视觉目标跟踪算法(Feature-Adaptative Selection for visual object Tracking,FAST),提出模板特征过滤器,筛选历史帧信息,迭代出当前时刻的最佳模板特征;设计搜索特征划分器,动态划分搜索特征为相关特征与独立特征,削弱不恰当的特征关联;设计了适配上述模块的分离注意力机制编码器进行特征交互,引入时空信息令牌,在跨帧传播信息的同时,弥补了深度互相关局部最优的缺陷. FAST通过自适应选取搜索区域并构建动态模板池,有效减轻了噪声区域的交互,时空信息逐帧修正物体的跟踪轨迹. 在OTB100[25]、LaSOT[26]和UAV123[27]等多个大规模公开数据集的各项挑战中取得了鲁棒结果,有效克服了现有跟踪器的局限.

2 方法架构

2.1 方法整体框架

本文算法的整体框架如图1所示,主要由骨干网络、特征自适应选取模块、分离注意力特征交互编码器和预测子网组成. 骨干网络将输入图像进行特征嵌入;特征自适应选取模块包含模板特征过滤器和搜索特征鉴别器,模板特征过滤器接收历史帧模板特征与当前帧搜索特征进行交互,输出适合当前搜索区域的模板过滤特征,搜索特征鉴别器感知模板过滤特征的信息

表示,将搜索帧划分为相关特征和独立特征;在此基础上利用分离注意力机制编码器进行特征交互,同时以迭代的方式更新初始令牌,跨帧传播时空信息;编码器输出的搜索区域响应送至预测子网,通过角点预测头回归目标位置,并经分数头判定,选取高置信度的跟踪结果进行模板更新.

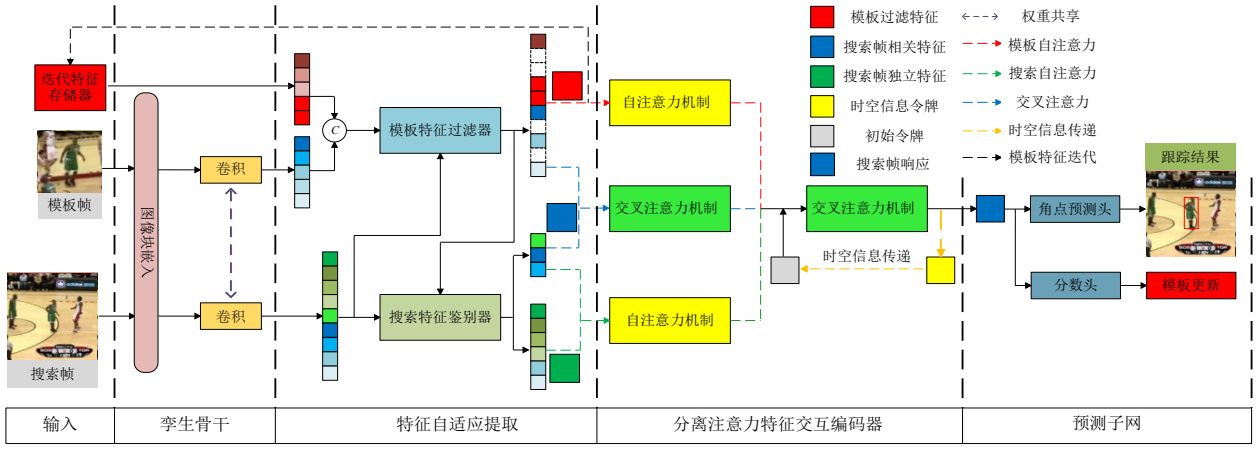


图1　FAST模型整体框架

2.2　模板特征过滤器

孪生子网接收模板帧图像 $\boldsymbol{z}\in\mathbf{R}^{3\times H_z\times W_z}$ 与搜索帧图像 $\boldsymbol{x}\in\mathbf{R}^{3\times H_x\times W_x}$ 作为输入,首先通过Patch Embedding层将模板帧 $\boldsymbol{z}$ 和搜索帧 $\boldsymbol{x}$ 分别划分成数量为 $N_z=H_zW_z/S^2$ 和 $N_x=H_xW_x/S^2$ 的非重叠图像块 $\boldsymbol{Z}_p$ 和 $\boldsymbol{X}_p$. 其中,S^2 表示每个图像块大小. 其次,通过卷积层将这些图像块线性映射为模板特征 $\boldsymbol{T}^{z_0}=\left\{\boldsymbol{T}_1^{z_0},\boldsymbol{T}_2^{z_0},\cdots,\boldsymbol{T}_{N_z}^{z_0}\right\}$ 和搜索特征 $\boldsymbol{T}^{x}=\left\{\boldsymbol{T}_1^{x},\boldsymbol{T}_2^{x},\cdots,\boldsymbol{T}_{N_x}^{x},\right\}$. 其中 $\boldsymbol{T}\in\mathbf{R}^{1\times C}$,$C$ 为每个特征块的维度. 最后将 $\boldsymbol{T}_x$ 和 $\boldsymbol{T}_z$ 拼接在一起,生成长度为 N_z+N_x 维度为 C 的特征 $\boldsymbol{T}_{\boldsymbol{m}}\in\mathbf{R}^{(HW\times C)}$ 输入到模板特征过滤器中.

传统的模板更新方法[28~30]一般以固定的时间间隔将模板存储在预先定义的模板池中,在模板池容量达到上限时,会以先入先出的方式丢弃早期模板,更新过程可以表示如下.

$$\boldsymbol{M}_t=\left\{\boldsymbol{z}_0,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_{t-1}\right\}\cup\boldsymbol{z}_t,\boldsymbol{M}\in\mathbf{R}^{n\times 3\times H_z\times W_z}\tag{1}$$

其中,$\boldsymbol{M}_t$ 表示 t^{th} 帧的模板池;$\boldsymbol{z}_0$ 为初始帧框选的模板,它会一直存储在模板池中;t 表示时间步长;n 表示模板个数.

图2(a)中展示了传统的模板更新方式,这种直接将整个跟踪结果反馈至模板池中的更新方式会不可避免地引入干扰信息,且直接丢弃早期的模板很容易丢失代表性特征,为后续判别响应带来麻烦. 为解决这一问题,本文提出一种新的特征级模板更新方式,整体流程如图2(b)所示. 通过模板特征过滤器、跟踪器动态地从模板池 $\boldsymbol{M}_{t-1}$ 和待更新的模板中自适应选取当前搜索区域的相关模板帧特征序列 $\boldsymbol{m}_{\mathrm{R}}$. 为防止信息丢失,对弱相关模板区域 $\boldsymbol{m}_{\mathrm{I}}$,本文并没有选择直接滤除而是采用特征压缩方式聚合为当前时刻的模板特征 $\boldsymbol{T}_z^{\mathrm{F}}$ 用于后续特征交互,模板特征过滤器的架构如图2(c)所示.

首先,将迭代特征存储器初始化为 $\boldsymbol{M}_0=\left\{\boldsymbol{T}^{z_0}\right\}$,并在内存未满时向其添加新的模板 $\boldsymbol{T}^{z_t}$,当模板池达到最大值 $N_{\max}$ 时,将模板池与当前时刻的搜索特征 $\boldsymbol{T}^{x_t}$ 送入模板特征过滤器中,计算存储的全部模板特征与当前时刻搜索区域的相关性,该过程可表述如下:

$$\begin{aligned}&\boldsymbol{T}_t=\boldsymbol{M}_{t-1}\cup\boldsymbol{T}^{z_t}\cup\boldsymbol{T}^{z_t}\\&\boldsymbol{Q}=\boldsymbol{T}_t,\ \boldsymbol{KV}=\boldsymbol{T}_t\\&\boldsymbol{T}_t'=\mathrm{LayerNorm}\left(\boldsymbol{T}_t+\mathrm{MHA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})\right)\\&\boldsymbol{T}_{\boldsymbol{m}}=\mathrm{LayerNorm}\left(\boldsymbol{T}_t'+\mathrm{FFN}(\boldsymbol{T}_t')\right)\end{aligned}\tag{2}$$

其中,$\boldsymbol{T}_t$ 表示当前时刻模板与搜索区域的特征集合,$\boldsymbol{T}_t'$ 表示多头自注意力的中间结果;$\boldsymbol{T}_{\boldsymbol{m}}$ 表示多头自注意力的运算结果;LayerNorm表示层归一化;FFN(Feed Forward Network)表示前馈层.

如式(3)所示,将得到的注意力权重输入到一个轻量的多层感知机中,通过非线性变换获得模板池特征与当前搜索区域特征的相关性得分:

$$\begin{aligned}&\boldsymbol{w}^*=\frac{\sum_{j=0}^{N_x}\boldsymbol{w}_j}{N_x}\\&S=\mathrm{SoftMax}\left(\mathrm{MLP}(\boldsymbol{w}^{*\mathrm{T}})\right)\end{aligned}\tag{3}$$

其中,$\boldsymbol{w}^*$ 表示当前时刻模板特征与搜索区域特征的注意力权重;$\boldsymbol{w}_j$ 表示第 j 个搜索特征对应的注意力权重;S 表示模板池特征与当前搜索区域特征的相关性得分集合.

随后,对相关性得分结果进行降序排列并按照预先定义的模板特征保留比例 k,将需要保留的高分特征块划分为相关特征序列 $\boldsymbol{m}_{\mathrm{R}}$,忽视的低分模板特征划分

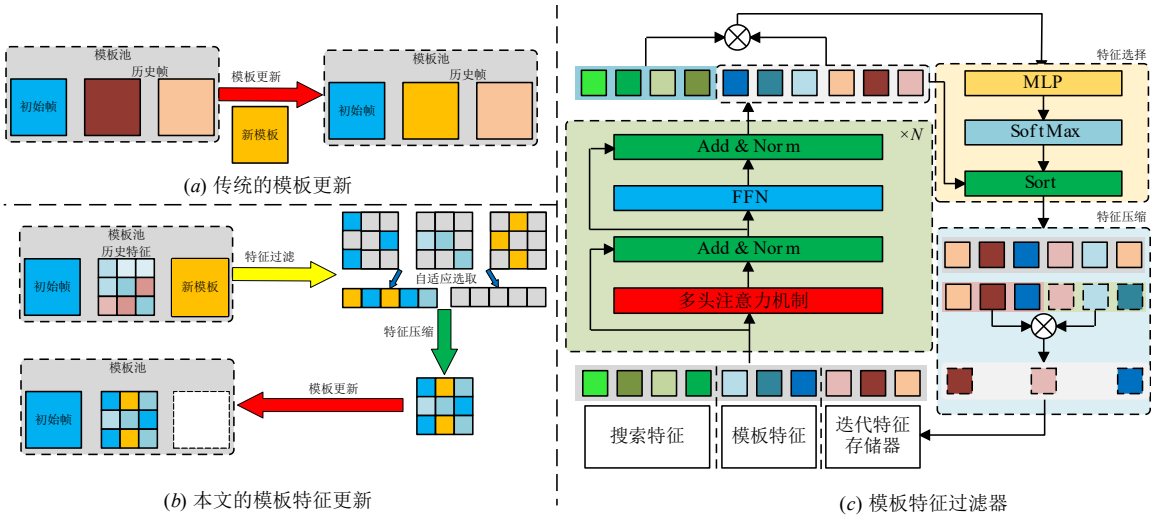


图2 模板特征过滤器示意图

为非相关特征序列$\boldsymbol{m}_1$,该过程的原理如式(4)所示:

$$\begin{aligned} &S'=\mathrm{Sort}(S),\ N_k=\lfloor k\times N\rfloor \\ &\begin{cases}\boldsymbol{m}_{\mathrm{R}}=\{\boldsymbol{T}_n|S'_n\geqslant S'_{N_k}\}\\ \boldsymbol{m}_{\mathrm{I}}=\{\boldsymbol{T}_n|S'_n<S'_{N_k}\}\end{cases}\end{aligned} \tag{4}$$

其中,S'表示降序排列后的相关性得分集合;N_k表示保留的模板特征块数量;$\lfloor\cdot\rfloor$表示向下取整函数;N为模板特征总数;$\boldsymbol{T}_n$表示第n个特征块;S_n表示第n个特征块的得分;S_{N_k}表示第N_k个特征块的相关性得分.

目前的主流跟踪器在处理弱相关信息时大都采用逐层消除候选区域的策略[20,21,31],但这种直接删除特征的方式会破坏视觉信息的整体性,造成信息丢失,当特征在复杂场景下不具有辨别力时会导致性能下降. 为避免这一问题,本文采用一种模板特征压缩策略,将非相关信息压缩至关联度较高的相关特征中,避免造成信息丢失. 针对非相关特征序列中的每个特征块$\boldsymbol{T}_i\in\boldsymbol{m}_{\mathrm{I}}$,首先在相关特征序列中找到其最近邻的特征块$\boldsymbol{T}_j\in\boldsymbol{m}_{\mathrm{R}}$,计算过程如下:

$$s_{i,j}=\frac{\boldsymbol{T}_i^{\mathrm{T}}\boldsymbol{T}_j}{\|\boldsymbol{T}_i\|\|\boldsymbol{T}_j\|} \tag{5}$$

$$\omega_i=\frac{\exp(s_{i,j})}{\sum\limits_{\boldsymbol{T}_i\in\boldsymbol{m}_{\mathrm{I}}}\exp(s_{i,j})+e} \tag{6}$$

$$\omega_j=\frac{e}{\sum\limits_{\boldsymbol{T}_i\in\boldsymbol{m}_{\mathrm{R}}}\exp(s_{i,j})+e} \tag{7}$$

其中,$s_{i,j}$表示二者对应区域的相似性得分;ω_i为每个非相关特征的压缩权值;ω_j为每个相关特征本身的压缩权值.

随后,根据压缩权值对非相关特征和相关特征进行整合,获得当前时刻过滤后的模板特征$\boldsymbol{T}_z^{\mathrm{F}}$,与初始帧特征$\boldsymbol{T}^{z_0}$合并到迭代特征存储器中,得到更新后的模板池$\boldsymbol{M}_t$,迭代流程如式(8)所示:

$$\begin{aligned} &\boldsymbol{T}_z^{\mathrm{F}}=\omega_j\boldsymbol{T}_j+\sum_{\boldsymbol{T}_i\in\boldsymbol{m}_{\mathrm{I}}}\omega_i\boldsymbol{T}_i \\ &\boldsymbol{M}_t=\boldsymbol{T}_z^{\mathrm{F}}\bigcup\boldsymbol{T}^{z_0}\end{aligned} \tag{8}$$

为直观说明模板特征过滤与模板池构建流程,图3展示了模板池与模板特征过滤的可视化图,蓝色虚线框表示模板池,红色虚线框以一帧待更新模板为示例进行了特征过滤可视化. 模板池始终包含初始帧的特征,当模板池已满时,新的高置信度模板经过多层过滤自适应筛选出最适合当前搜索区域的特征,用于后续特征交互,并将迭代特征更新回模板池. 可视化结果显示,模板过滤器能有效划分模板中的弱相关区域,模板池保留初始帧原始信息的同时可以动态构建模板特征表示,有效提升跟踪器在复杂动态场景下的鲁棒性.

2.3 搜索特征鉴别器

在目标跟踪任务中,模板和搜索区域之间的交互至关重要. 搜索区域包含大量背景信息,模板直接和无关区域的特征交互,会产生不良的关系建模,强化与目标无关的特征,进而导致前景与背景特征混淆. 因此,本文提出一种搜索特征鉴别器,在交互前判别搜索区域特征,确保搜索帧信息充分聚合,同时削弱模板和搜索区域中噪声的不良交叉,具体结构如图4所示.

图4搜索特征鉴别器由模板信息感知器和轻量化的二分类网络2个部分组成. 模板信息感知器接收当前时刻过滤的模板特征$\boldsymbol{T}_z^{\mathrm{F}}$作为输入,分别送入2个1×1大小的卷积层,减少特征通道数的同时保留关键的空间信息;应用全局平均池化操作,将每个通道的空间信息压缩成单个标量,调整尺寸以适配原始特征图维度;通过SoftMax函数处理后,生成归一化权重向量$\boldsymbol{W}$,式(9)展示了这一过程:

图3　模板池与模板过滤可视化示意图

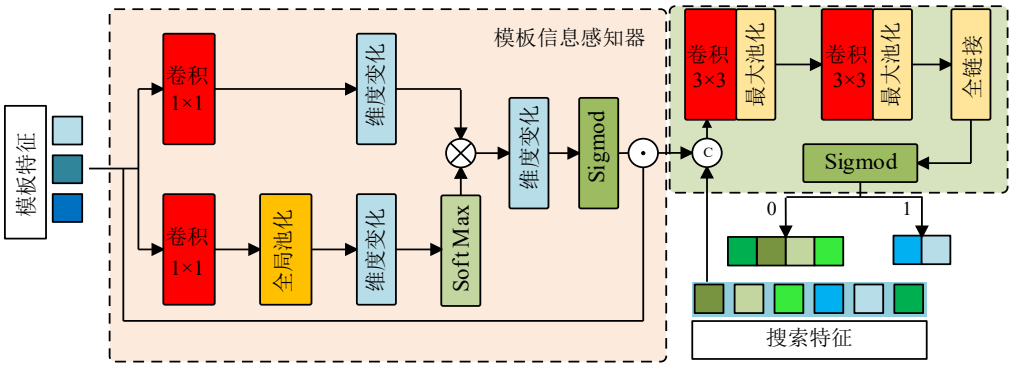


图4　搜索特征鉴别器示意图

$$\boldsymbol{W}=\mathrm{SoftMax}\left(\mathrm{GlobalPool}\left(\mathrm{Conv}(\boldsymbol{T}_z^{\mathrm{F}})\right)\right) \tag{9}$$

通过权重向量 $\boldsymbol{W}$ 指导网络加权 $\boldsymbol{T}_z^{\mathrm{F}}$ 的关键特征,将加权结果通过 Sigmoid 函数进行转换,输出最终的注意力图 $\boldsymbol{M}$,如式(10)所示:

$$\boldsymbol{M}=\sigma\left(\boldsymbol{W}\odot\mathrm{Conv}(\boldsymbol{T}_z^{\mathrm{F}})\right) \tag{10}$$

其中,σ 表示 Sigmoid 函数.

将注意力图 $\boldsymbol{M}$ 与原始模板特征 $\boldsymbol{T}_z^{\mathrm{F}}$ 进行特征重标定,得到最终的模板信息感知特征,该过程的主要原理如式(11)所示:

$$\boldsymbol{T}_z^{\mathrm{P}}=\boldsymbol{T}_z^{\mathrm{F}}\odot\boldsymbol{M} \tag{11}$$

其中,$\boldsymbol{T}_z^{\mathrm{P}}$ 表示模板信息感知特征.

搜索特征鉴别器的另一部分由轻量化的 CNN(Convolutional Neural Network)网络组成,该部分接收模板信息感知特征 $\boldsymbol{T}_z^{\mathrm{P}}$ 与搜索特征 $\boldsymbol{T}_x$ 作为输入,输出搜索特征的类别概率. 2 层卷积分别由 384 和 192 个 3 × 3 大小的卷积核组成,采用 GELU(Gaussian Error Liner Unit)函数激活,每个卷积层后接一个 2 × 2 大小的池化层降维;

最后用一个全连接层进行非线性变换，使用Sigmoid激活函数，将输入的搜索特征 $\boldsymbol{T}_x$ 划分为搜索帧独立特征 $\boldsymbol{T}_x^{\mathrm{I}}$ 与搜索帧相关特征 $\boldsymbol{T}_x^{\mathrm{COR}}$. 在端到端训练过程中，分类器的监督标签由定位任务的置信度动态生成，搜索区域的前景和背景划分作为正负样本的标注. 利用二进制交叉熵损失优化特征划分能力，使分类网络能够自适应学习区分判别性特征，同时避免影响定位任务的可靠性. 图5展示了随机选择的搜索帧特征鉴别器可视化结果，目标潜在的相关特征 $\boldsymbol{T}_x^{\mathrm{COR}}$ 被映射为图中的显著图像块，搜索帧独立特征 $\boldsymbol{T}_x^{\mathrm{I}}$ 被映射为掩码图像块.

图5 搜索特征鉴别可视化

2.4 分离注意力机制特征交互编码器

传统的注意力机制通常采用全部的模板帧与搜索帧特征进行特征交互，包括自注意力计算与交叉注意力计算，分别用 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\in\mathbf{R}^{HW\times C}$ 表示查询、键和值. 它们进行建模的过程可以表示为

$$f=\mathrm{Attn}\left([\boldsymbol{Q}=\boldsymbol{T}_x,\mathbf{KV}=\boldsymbol{T}_z]\right)$$
$$\mathrm{Attn}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\mathrm{Softmax}\left(\left(\frac{\overline{\boldsymbol{Q}}\,\overline{\boldsymbol{K}}^{\mathrm{T}}}{\sqrt{C}}\right)\overline{\boldsymbol{V}}\right)\boldsymbol{W}_{\mathrm{o}}\tag{12}$$

其中，$\overline{\boldsymbol{Q}}\,\overline{\boldsymbol{K}}\,\overline{\boldsymbol{V}}$ 是不同的线性变换，$\overline{\boldsymbol{Q}}=\boldsymbol{Q}\boldsymbol{W}_q$，$\overline{\boldsymbol{K}}=\boldsymbol{K}\boldsymbol{W}_k$，$\overline{\boldsymbol{V}}=\boldsymbol{V}\boldsymbol{W}_v$，$\boldsymbol{W}_q\boldsymbol{W}_k\boldsymbol{W}_v$ 和 $\boldsymbol{W}_{\mathrm{o}}$ 分别表示查询、键、值和输出的线性变换权重.

这种特征交互模式的跟踪器只能在每个图像对内进行全局交互，建立有限的时间相关性. 本文以传统注意力机制为基础，引入时空信息传播令牌，针对划分的不同特征设计分离注意力机制编码器，以此适配特征自适应选取后的特征交互操作，具体结构如图6所示.

针对划分后的搜索帧独立特征 $\boldsymbol{T}_x^{\mathrm{I}}$，首先与搜索帧相关特征 $\boldsymbol{T}_x^{\mathrm{COR}}$ 合并为完整的搜索帧特征 $\boldsymbol{T}_x$. 通过计算搜索帧内部的自注意力保证相关搜索标记之间的充分互动，避免二进制划分可能造成的功能孤立问题，整体流程可以表示为

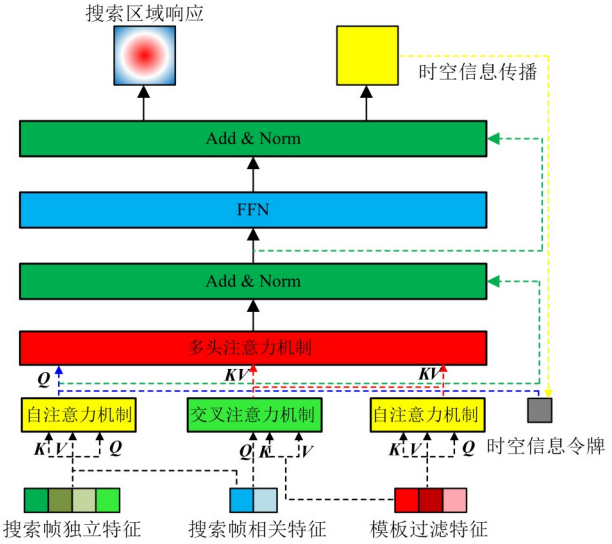


图6 分离注意力机制编码器

$$\begin{aligned}&\boldsymbol{T}_x=\boldsymbol{T}_x^{\mathrm{I}}\cup\boldsymbol{T}_x^{\mathrm{COR}}\\&\boldsymbol{Q}=\boldsymbol{T}_x,\boldsymbol{KV}=\boldsymbol{T}_x\\&\boldsymbol{T}_x'=\mathrm{LayerNorm}\left(\boldsymbol{T}_x+\mathrm{Attn}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})\right)\end{aligned}\tag{13}$$

其中，$\boldsymbol{T}_x'$ 表示搜索帧特征的自注意力输出结果，LayerNorm表示层归一化.

针对当前时刻过滤后的模板特征 $\boldsymbol{T}_z^{\mathrm{F}}$，通过计算内

部自注意力进一步增强模板帧的特征表示,充分捕捉目标的关键视觉线索,整体流程可以表示为

$$\begin{aligned}&\boldsymbol{Q}=\boldsymbol{T}_z^{\mathrm{F}},\ \boldsymbol{KV}=\boldsymbol{T}_z^{\mathrm{F}}\\&\boldsymbol{T}_z^{\mathrm{F}}{}'=\mathrm{LayerNorm}\left(\boldsymbol{T}_z^{\mathrm{F}}+\mathrm{Attn}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})\right)\end{aligned}\tag{14}$$

其中,$\boldsymbol{T}_z^{\mathrm{F}}$表示过滤后的模板帧特征自注意力输出结果.

针对鉴别的搜索帧相关特征$\boldsymbol{T}_x^{\mathrm{COR}}$,允许其与过滤后的模板特征$\boldsymbol{T}_z^{\mathrm{F}}$进行初始特征交互,通过计算二者的交叉注意力,强化与目标相关的区域表征,提升后续计算搜索区域响应的判别精度,整体流程可以表示为

$$\begin{aligned}&\boldsymbol{Q}=\boldsymbol{T}_x^{\mathrm{COR}},\boldsymbol{KV}=\boldsymbol{T}_z^{\mathrm{F}}\\&\boldsymbol{T}_x^{\mathrm{COR}}{}'=\mathrm{LayerNorm}\left(\boldsymbol{T}_x^{\mathrm{COR}}+\mathrm{Attn}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})\right)\end{aligned}\tag{15}$$

其中,$\boldsymbol{T}_x^{\mathrm{COR}}{}'$表示搜索帧相关特征的交叉注意力输出结果.

在跟踪中,相邻帧的目标变化不大[7],基于这一思想进行全局特征交互时本文引入了时空信息传播令牌,编码器以自回归方式将时空信息令牌从t^{th}帧传播到$(t+1)^{\mathrm{th}}$帧,每帧的时空信息令牌包含了上一帧目标的外观及位置信息与过滤后的模板特征形成良好互补关系,指导生成更精准的交叉响应,有效优化跟踪轨迹.

在跟踪开始时,跟踪器会初始化一个空令牌E_{empty},E_{empty}仅在跟踪开始时初始化一次,后续帧的空令牌为占位符,用于接收前一帧的时空信息. 在处理第t帧时,时空信息令牌E_t包括2个部分,分别是$t-1$帧中传递的时空信息E_{t-1}以及第t帧包含目标外观及位置特征. 将E_t与$t+1$帧的空令牌E_{empty}^{t+1}结合,生成更新后的时空令牌E_{t+1}. 随后,E_{t+1}作为下一帧的初始令牌反馈送回输入端进行交叉注意力计算与后续时空信息的传播. 整体流程可以表示为

$$\begin{aligned}&\boldsymbol{Q}=[\boldsymbol{T}_x';\ E_{\mathrm{empty}}],\ \boldsymbol{KV}=[\boldsymbol{T}_z^{\mathrm{F}};\ \boldsymbol{T}_x^{\mathrm{COR}}{}']\\&\boldsymbol{f}_t{}'=\mathrm{LayerNorm}\left([\boldsymbol{T}_x';\ E_{\mathrm{empty}}]+\mathrm{MHA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})\right)\\&\boldsymbol{f}_{\mathrm{c}}=\boldsymbol{f}_t{}'+\mathrm{FFN}\left(\mathrm{LayerNorm}(\boldsymbol{f}_t{}')\right),\boldsymbol{f}_{\mathrm{c}}=[\boldsymbol{f}_t;\ E_t]\\&E_{t+1}=E_t+E_{\mathrm{empty}}^{t+1}\end{aligned}\tag{16}$$

其中,$\boldsymbol{f}_t{}'$表示中间运算结果;$\boldsymbol{f}_{\mathrm{c}}$表示未解耦的交叉注意力结果;$E_{\mathrm{empty}}$表示初始化的空令牌;$E_t$表示当前帧的时空信息令牌;$E_{t+1}$表示更新的时空令牌;$\boldsymbol{f}_t$表示当前帧的融合特征表示,后续将送至预测子网中.

时空令牌本质是一个动态存储单元,保存目标的时序外观和运动轨迹特征. 在训练第1阶段,仅使用单帧图像对优化角点预测头,此时时空令牌未被激活;在训练第2阶段,引入3帧连续序列,时空令牌参与跨帧信息传播,逐帧更新令牌并计算损失;在推理阶段,模型逐帧处理视频流,时空令牌持续传递历史信息,避免依赖长时记忆导致的误差累积. 图7展示了引入时空信息传播令牌前后的互相关响应可视化结果. 时空信息传播令牌有效集合了当前帧目标的外观与轨迹信息,相较于未引入时空信息传播令牌的原始互相关响应,引入后的响应区域更精准,有效提升了跟踪器的回归精度,详细精度提升将在后文的消融实验中进一步说明.

2.5　测头和损失

在分离注意力机制编码器执行特征交互后,本文采用STARK[16]中提出的边界框角点预测头进行角点位置回归. 边界框角点预测头由一个全卷积网络(Fully Convolutional Network,FCN)组成,FCN包括了L个堆叠的Conv-BN-ReLU(Convolutional layer-Batch Normalization-Rectified Linear Unit)层以及Soft-argmax函数,最终输出一个双通道特征图,每个通道分别对应于边界框左上角和右下角概率图$P_{\mathrm{tl}}(x,y)$和$P_{\mathrm{br}}(x,y)$,通过式(17)计算角点概率分布的期望可以获得预测框左上角和右下角坐标$(x_{\mathrm{tl}},y_{\mathrm{tl}})$和$(x_{\mathrm{br}},y_{\mathrm{br}})$:

$$\begin{cases}(x_{\mathrm{tl}},y_{\mathrm{tl}})=\left(\dfrac{\sum\limits_{(x,y)}x\cdot P_{\mathrm{tl}}(x,y)}{\sum\limits_{(x,y)}P_{\mathrm{tl}}(x,y)},\dfrac{\sum\limits_{(x,y)}y\cdot P_{\mathrm{tl}}(x,y)}{\sum\limits_{(x,y)}P_{\mathrm{tl}}(x,y)}\right)\\(x_{\mathrm{br}},y_{\mathrm{br}})=\left(\dfrac{\sum\limits_{(x,y)}x\cdot P_{\mathrm{br}}(x,y)}{\sum\limits_{(x,y)}P_{\mathrm{br}}(x,y)},\dfrac{\sum\limits_{(x,y)}y\cdot P_{\mathrm{br}}(x,y)}{\sum\limits_{(x,y)}P_{\mathrm{br}}(x,y)}\right)\end{cases}\tag{17}$$

本文使用一个分数头对跟踪结果进行评判,提供更加可靠的新模板,分数头由一个3层的MLP(Multi-Layer Perceptron)组成,使用Sigmoid函数激活,当跟踪结果的得分高于阈值δ时,则认为当前状态可信度较高,并将其作为新模板更新回迭代特征存储器,算法的整体跟踪流程如图8所示.

为了避免联合学习中出现的次优解和将位置定位和前景与背景的分类任务分开处理的问题,本文采用了多阶段的训练策略. 第1阶段的训练主要关注定位任务,而分类任务则作为辅助任务处理. 除了分数头部分外,整个网络采用端到端的训练方式. 使用式(18)定义的定位损失函数优化性能,通过这一训练策略,模型能够精准地完成目标定位,不受分类任务干扰.

$$L=\lambda_{\mathrm{iou}}L_{\mathrm{iou}}(b_i,\hat{b}_i)+\lambda_{L_1}L_1(b_i,\hat{b}_i)\tag{18}$$

其中,b_i和$\hat{b}_i$分别表示真实框和预测框;λ_{iou}、λ_{L_1}是人为设定的超参数;L_{iou}表示边界框重叠精确度损失;L_1表示边界框中心距离损失.

在第2阶段,采用二进制交叉熵损失优化分数头,提高模型区分前景背景的能力,使用的二进制交叉熵损失被定义为

$$L_{\mathrm{ce}}=y_i\ln(P_i)+(1-y_i)\ln(1-P_i)\tag{19}$$

其中,y_i是前景与背景的分类标签;P_i是预测结果的置信度得分. 在该阶段,冻结其他参数,防止影响模型的

图7 搜索帧互相关响应可视化结果

定位能力. 在完成两阶段的训练后,模型能够同时具备定位和分类能力.

3 实验分析

本节详细介绍了提出的FAST跟踪器在多个大规模数据集的实验结果,并与近几年主流的先进跟踪器在各项实际挑战下进行全面比较,通过消融实验进一步说明了提出的几个组件对跟踪性能的影响,同时探讨了算法实际运行速度和成功率的综合性能.

3.1 实验环境及参数设置

本算法在Ubuntu20.04系统环境下运行,使用Python3.8+Pytorch1.8.1框架编程实现,采用PySOT库对跟踪器的输出结果进行评价. 实验均在Intel®Xeon(R) CPU E5-2660 V2@2.20 GHz × 40和4 NVIDIA GeForce RTX 3090 GPUs显卡上运行. 搜索区域为目标区域的4倍,模板区域是目标区域的2倍,模板图像和搜索图像在训练和测试中被统一设置为256 × 256像素和128 × 128像素;使用MAE(Masked AutoEncoder)[32]预训练参数的ViT-B[33]作为初始化编码器;MLP采用2个由GELU激活的隐藏层和1个输出层,通道尺寸分别被设置为384、192和2;多头注意力有8个头,宽度为256;FFN层包含1 024个隐藏单元;模板池内存上限$N_{\max}$被设置为$3\times N_z$,保留比例k设置为0.75;置信度阈值δ设置为0.5;定位损失超参数λ_{L_1}和λ_{iou}分别被设置为5和2,学习率从10^{-2}~10^{-5}逐渐降低.

3.2 训练集和测试集

选取COCO[34]、GOT-10K[35]、TrackingNet[36]和LaSOT[26]这4个训练集,采用两阶段的训练措施,分别用于训练网络的定位和分类能力. 采用随机水平翻转和亮度变化进行数据增强;每个数据集被选中概率均相同,第1阶段训练了300个epoch,每个epoch有

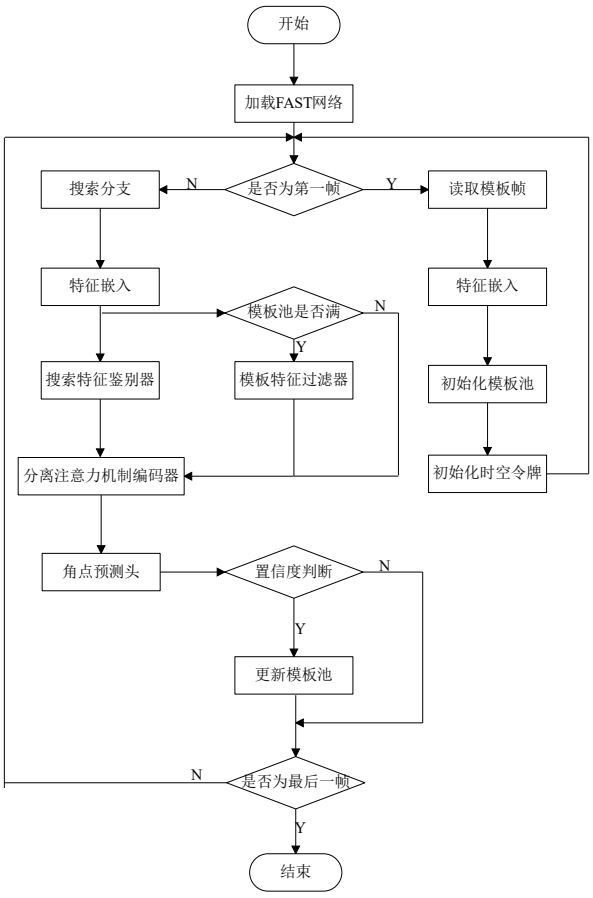


图8　FAST跟踪流程

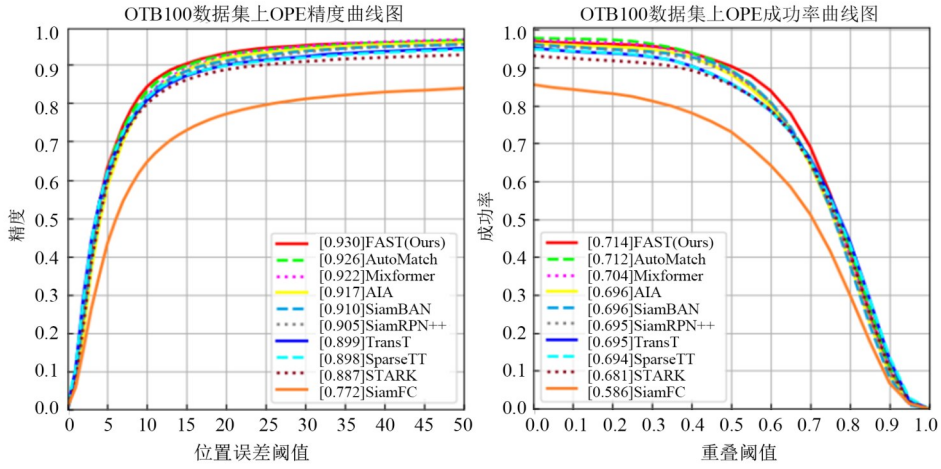
60 000个图像对;第2阶段在第1阶段的基础上进行微调,训练了50个epoch. 选取OTB100[25]、LaSOT[26]和UAV123[27]这3个测试集评估跟踪器不同场景下的性能.

3.3　OTB100基准实验

OTB100数据集[25]包含100个完全标注的真实世界视频序列,是单目标跟踪领域算法性能评估的重要判据. 该数据集涵盖了运动模糊、光照变化、遮挡、形变等11种实际挑战,为评价跟踪算法的准确性与鲁棒性提供了有效的测试场景. 本文在OTB100数据集上对FAST算法进行了定量实验、定性实验与消融实验,通过成功率曲线和精度这2个指标来评估跟踪性能.

3.3.1　定量实验

本文选取SimaFC[7]、STARK[16]、SparseTT[21]、TransT[13]、SiamRPN++[8]、SiamBAN[10]、AIA[18]、Mixformer[17]、AutoMatch[12]这9种对比算法进行了OTB基准的定量实验,结果如图9所示. FAST跟踪器在成功率和精度2项指标上分别取得71.2%和93.0%的表现,与所选对比算法相比,成功率略低于AutoMatch[12]位居第2,但精度排名第1. 与同样采用了长短时更新策略的AIA[18]相比,成功率提升了0.6%,精度提升了1.3%;相对于进行了背景稀疏的SparseTT算法精度超过了3.2%,准确率超过了1.8%. 以上数据说明本文设计的特征自适应选取方式应用于实际跟踪任务时的有效性,极大程度提升了跟踪器性能.

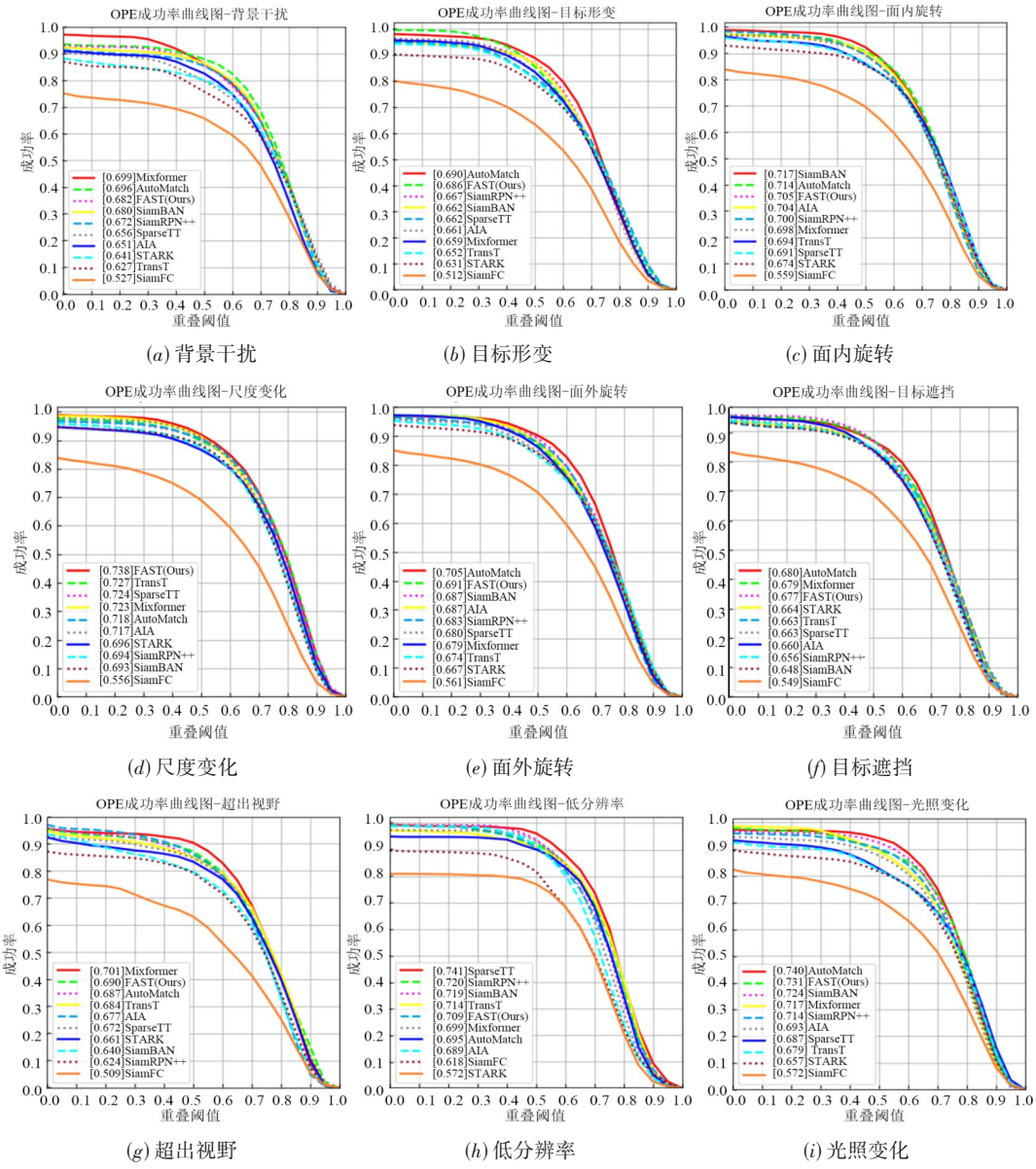


(a) 精度曲线　　(b) 成功率曲线

图9　OTB100整体的成功率曲线和精度曲线

为了全面评估本文算法面临实际挑战下的表现,在OTB100的多种属性上与对比算法进行了成功率与精度的全面比较,图10和图11分别展示了各算法在9种属性上的成功率和精度得分.

在成功率方面,FAST跟踪器在尺度变化这一属性下取得最优,在形变、面外旋转、超出视野、光照变化这4个属性上取得次优解. 这几种属性都包含了动态背景、相机角度或目标在运动过程中的变化,说明FAST在外观建模能力上表现出色,模板特征过滤器构建的动态模板集合能有效应对目标在跟踪中的变化情况.



(*a*) 背景干扰 (*b*) 目标形变 (*c*) 面内旋转

(*d*) 尺度变化 (*e*) 面外旋转 (*f*) 目标遮挡

(*g*) 超出视野 (*h*) 低分辨率 (*i*) 光照变化

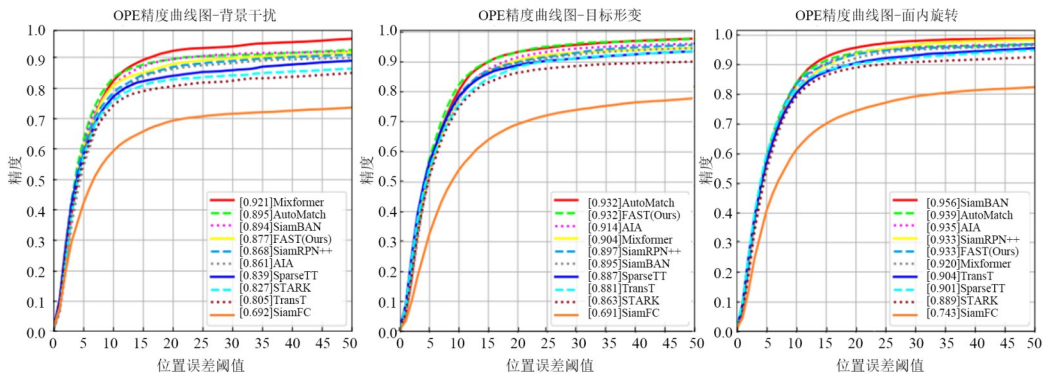
图 10 OTB100 数据集 9 种属性下成功率曲线

FAST在低分辨率属性上表现不佳,可能是由于原有的少量像素在变化后发生了特征失真,增加了特征划分的难度;在精度方面,跟踪器在尺度变化和遮挡2种挑战下表现最优,得益于时空信息令牌逐帧精准修正的搜索交互响应,在其他多项属性上FAST也取得了靠前的排名. 上述实验结果充分印证了本文算法在多种实际挑战下表现优异,具有较强的鲁棒性.

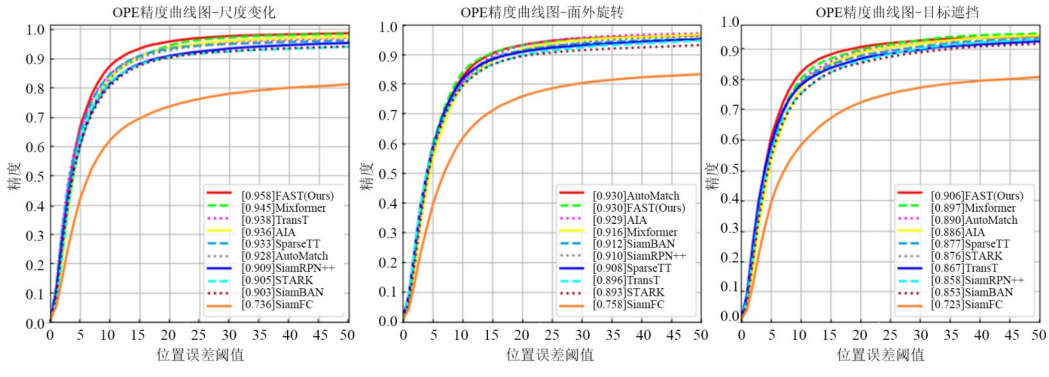
3.3.2 定性实验

为了直观体现FAST跟踪器的跟踪结果,同时定性评估算法性能,随机选取OTB100数据集中的5个视频序列,并选择Transfomer跟踪框架下的先进跟踪器TransT[13]、Siamese跟踪框架下的代表跟踪器SiamRPN++[8]及经典相关滤波器跟踪器SRDCF(Spatially Regularized Discriminative Correlation Filter)[37]进行可视化结果对比. 在图12中,红色框表示目标的真实框,绿色框、黄色框、紫色框和蓝色框分别代表了FAST、TransT、SiamRPN++、SRDCF这4个跟踪器的跟踪结果. 跟踪框与真实框之间的重叠比例越高代表跟踪结果越好.

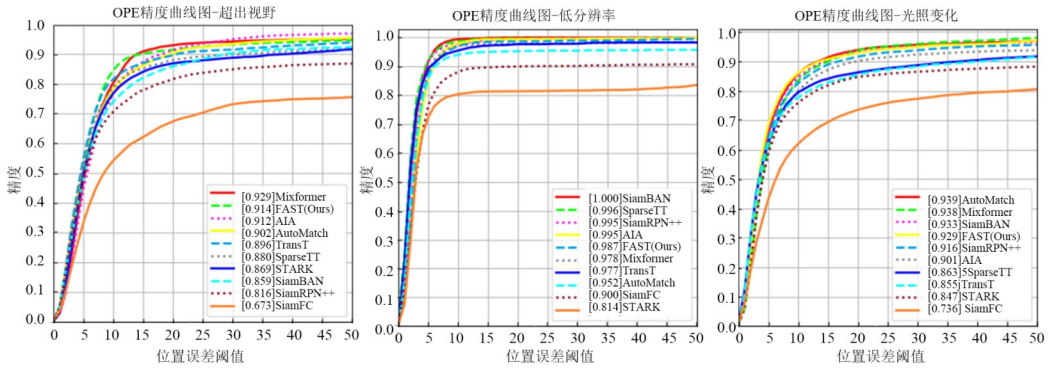
在Basketball视频序列中,如图12(*a*)所示,该序列的主要难点包括目标的快速移动、背景杂乱以及运动模糊. 在第73帧时,目标短时间内的快速移动导致TransT的跟踪框发生剧烈扰动,SiamRPN++跟踪器也在505帧后发生跟踪框漂移导致跟踪失败. 这一结果体现出本文所提出的双路权重共享骨干特征提取网络与特征自适应网络可有效提取目标的深度特征,并结合时



(*a*) 背景干扰　　(*b*) 目标形变　　(*c*) 面内旋转



(*d*) 尺度变化　　(*e*) 面外旋转　　(*f*) 目标遮挡



(*g*) 超出视野　　(*h*) 低分辨率　　(*i*) 光照变化

图 11　OTB100 数据集 9 种属性下精度曲线

空令牌信息给予高置信度的目标轨迹预测,有效地应对目标短时间内的快速移动,该结果进一步说明本文算法在处理运动模糊、背景杂乱等复杂情况下的鲁棒性和准确性.

在 Bolt 视频序列中,如图 12(*b*)所示,该序列的重要难点包括存在大量相似性干扰及目标的快速运动问题. 在第 11 帧时,TransT 发生跟踪框漂移. 从第 176 帧~194 帧,对比算法均发生了不同程度的跟踪框漂移,194 帧后 TransT 跟踪彻底失败. 本文算法针对子任务特点进行模板特征和搜索特征的有效过滤与鉴别,保持较高的准确率,这一结果进一步体现本文算法在应对相似性干扰和快速运动等挑战下的优越性和鲁棒性.

在 Box 视频序列中,如图 12(*c*)所示,该序列的主要难点包括跟踪目标大小不断发生变化且存在遮挡问题. 在第 32 帧时,跟踪过程中出现了与目标相似的背景,导致 SRDCF 和 SiamRPN++发生不同程度的跟踪漂移. 在第 498 帧后,随着目标在移动中被遮挡,对比算法接连跟丢目标. 虽然在 597 帧后,本文算法和对比算法均定位到目标,但本文算法与真实框之间的重叠率最高,且整个序列一直保持高精准度的稳健跟踪. 由此可以体现出,与对比算法相比,本文算法在尺度变化和目标遮挡等复杂情况下能保持稳健的跟踪.

在 Dive 视频序列中,如图 12(*d*)所示,该序列的主

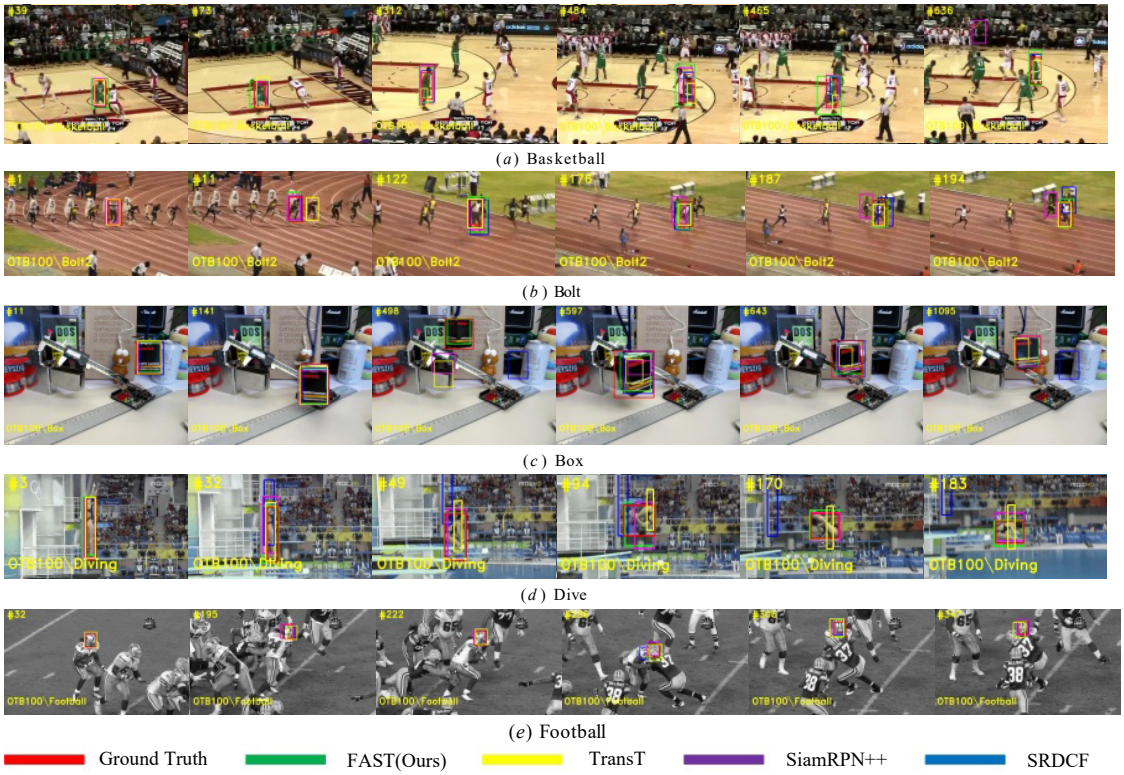


图 12 OTB100数据集上选定视频序列的定性结果

要难点在于目标的比例变化、面内旋转等挑战. 从第31帧~第49帧之间,目标在短时间内伴随着尺度变化与剧烈的面内旋转,导致SRDCF的跟踪框漂移越来越剧烈,直至最后完全失跟. 在第183帧时,对比算法的跟踪框的重叠率开始不断下降,而本文算法得益于分离注意力机制的设计,实现更加稳健的跟踪效果.

在Football视频序列中,如图12(*e*)所示,该序列的主要难点在于有许多与待跟踪目标相似的运动员,且目标保持着持续的位移,进而产生相互遮挡. 在前100帧时,所有算法均可以应对目标邻近的相似性背景,但在第109帧时,随着目标周围的干扰背景集中,SRDCF跟踪器发生了严重的跟踪漂移. 从第183帧~第361帧,由于不断发生遮挡与快速位移,对比跟踪器发生了不同程度的跟踪漂移和跟踪失败. 相比之下,本文算法得益于嵌入时空信息的分离注意力机制,可以更好地利用目标浅层的定位信息和深层语义信息,同时依靠特征自适应选取网络,使跟踪器聚焦目标的原始特征,在整个跟踪过程中保持较高的成功率.

3.3.3 消融实验

为分析各个模块对算法最终性能的贡献程度,本文在OTB100数据集上进行了消融实验,通过不同模块的组合分别评估算法的表现,以此体现各个模块的效果. 分别划分为FAST-base、FAST(0)、FAST(1)、FAST(2)和FAST(3)这5组模型进行实验,其中,FAST-base仅包含孪生骨干与用于特征交互的编码器-解码器架构,采用基于置信度的方式进行模板更新. 实验结果如表1所示.

表1 跟踪器各组件的消融实验结果 单位:%

模型编号	模板特征过滤器	搜索特征鉴别器	时空信息传播	成功率	精度
FAST-base				64.6	85.8
FAST(0)	√			67.2	87.6
FAST(1)	√	√		69.4	89.3
FAST(2)	√		√	69.0	90.1
FAST(3)	√	√	√	71.2	93.0

实验结果表明,采用模板特征过滤器的FAST(0)相较于FAST-base在成功率和精度上分别提升了2.6个百分点和1.8个百分点;FAST(1)相比仅使用模板特征过滤器的FAST(0)在成功率和精度上分别获得了2.2个百分点和1.7个百分点的提升;对比FAST(0)和FAST(2)的结果表明,本文提出的时空信息传播方式可以使算法的成功率和精度分别提升1.8个百分点和2.5个百分点;综合应用本文特征自适应选取方法的FAST(3)在2项指标上表现最佳,不仅验证了每个模块的有效性,更说明3个模块同时工作对整体性能的提升.

为分析本文模板特征过滤器更新方式效能,在不

改变其他模块的情况下替换了多种不同的模板更新策略,并进行了5组消融实验. 实验结果如表2所示,第1组表示仅使用初始帧,不进行模板更新;第2组表示按照一定的时间间隔以先入先出的方式更新模板;第3组表示引入分数头更新高置信度的模板;第4组表示特征过滤后将弱相关特征直接删除;第5组为本文提出的模板特征过滤器,在应用分数头的基础上压缩过滤特征进行迭代更新.

表2　不同模板更新方式的消融实验结果　　单位:%

编号	模板更新方式	成功率	精度
1	固定模板	69.4	90.0
2	周期性更新	70.0	91.7
3	置信度更新	70.3	92.4
4	基于剔除的特征过滤	70.8	92.6
5	基于压缩的特征过滤	71.2	93.0

消融实验结果表明模板更新策略对跟踪算法的性能有显著影响. 与固定模板相比,通过定期更新模板,可以有效捕捉目标的外观变化,提高跟踪的成功率和精度;置信度更新能减轻错误信息的累积,在此基础上过滤出当前时刻最优的模板特征可显著提升跟踪器的准确率;而特征压缩可以避免信息丢失,这种优化处理方式可以有效提高跟踪性能.

3.4　LaSOT数据集实验结果

LaSOT(Large-scale Single Object Tracking, LaSOT)数据集[26]是一个庞大的目标跟踪评估平台,为跟踪算法提供了一个全面且充满挑战的测试环境. LaSOT数据集拥有1 400个视频序列,总计超过35万帧图像,其中测试集包含了280个序列,每个序列平均有2 448帧. 它包含了多样化的真实场景视频,覆盖多种目标类别和挑战,如形变、遮挡、光照变化和目标形变等. 评价指标包括归一化精度、准确率和AUC(Area Under the Curve),综合反映了跟踪器在不同条件下的性能.

为全面评估FAST跟踪器的性能,本文选取相关滤波、孪生网络、卷积神经网络和Transformer四大类跟踪范式,每种跟踪范式选取5种算法进行了对比. 包括基于相关滤波的CSK(Circulant Structure of tracking-by-detection with Kernels)[38]、KCF(Kernelized Correlation Filter)[39]、DSST(Discriminative Scale Space Tracking)[40]、Staple[37]、SRDCF[41];基于CNN网络的ECO(Efficient Convolution Operators for tracking)[42]、ATOM[11]、Ocean[43]、GlobalTrack[15]、DiMP50[44];基于孪生网络的SiamFC[7]、SiamRPN++[8]、SiamBAN[10]、SiamCAR[9]、SiamGAT[19];基于Transformer的SparseTT[21]、CSWinTT[24]、OSTrack[20]、Mixformer[17]、GRM[22],跟踪器在LaSOT数据集上的实验结果如表3所示. 由实验结果可知,FAST跟踪器的跟踪归一化精度为79.6%,AUC得分为70.1%,在跟踪归一化精度、准确率和AUC这3项指标上的表现均优于其他对比算法. 上述实验结果表明了FAST跟踪器应用于LaSOT数据集上的效能和优势.

表3　LaSOT数据集实验结果　　单位:%

跟踪器	类型	AUC↑	P_n↑	P↑
CSK	CF	22.9	14.9	14.9
KCF		26.7	19.0	17.8
DSST		28.3	21.3	20.7
Staple		34.6	27.8	24.3
SRDCF		34.8	24.8	24.5
ECO	CNN	32.4	33.8	30.1
ATOM		51.5	57.6	50.5
Ocean		56.0	65.1	56.6
GlobalTrack		52.1	59.7	51.7
DiMP50		57.7	66.4	57.9
SiamFC	Siamese	33.6	42.0	33.9
SiamRPN++		49.5	57.0	49.3
SiamBAN		51.4	59.8	52.1
SiamCAR		50.7	60.0	51.0
SiamGAT		53.9	63.3	53.0
SparseTT	Transformer	66.0	74.0	70.1
CSWinTT		66.2	75.2	70.9
OSTrack		69.1	78.7	75.2
Mixformer		69.2	78.7	74.7
GRM		69.9	79.3	75.8
FAST(Ours)	Transformer	70.1	79.6	75.9

为进一步分析FAST算法在背景干扰(Background Clutters, BC)、完全遮挡(Full OCclusion, FOC)、变形(DEFformation, DEF)、相机运动(Camera Motion, CM)这4类算法主要针对问题上的表现,将这4个属性与所选算法在LaSOT数据集上进行对比实验,实验结果如图13所示.

由图13的结果分析,本文算法在BC、DEF、FOC、CM这4个属性上的3项指标显著优于其他对比算法. 特别是背景干扰这一属性,本文算法的归一化精度相较于排名第2的OSTrack算法[20]提升了3.4%,数据有效验证了本文基于特征自适应选取的方法可有效避免噪声区域的信息交互,防止错误的交叉关系影响相关性判别. 目标形变和相机运动这2项属性上的表现再次说明FAST具有出色的目标外观建模能力,能良好应对动态变化,在目标形变这一属性上,成功率超过排名第2的GRM算法[22]1.4%,这是由于本文的模板特征过滤器能够动态地迭代跟踪序列的代表性特征,通过不断整合模板的时空信息,选择适合当前帧的特征,提升跟踪的鲁棒性.

上述实验数据,进一步说明本文自适应特征选取方

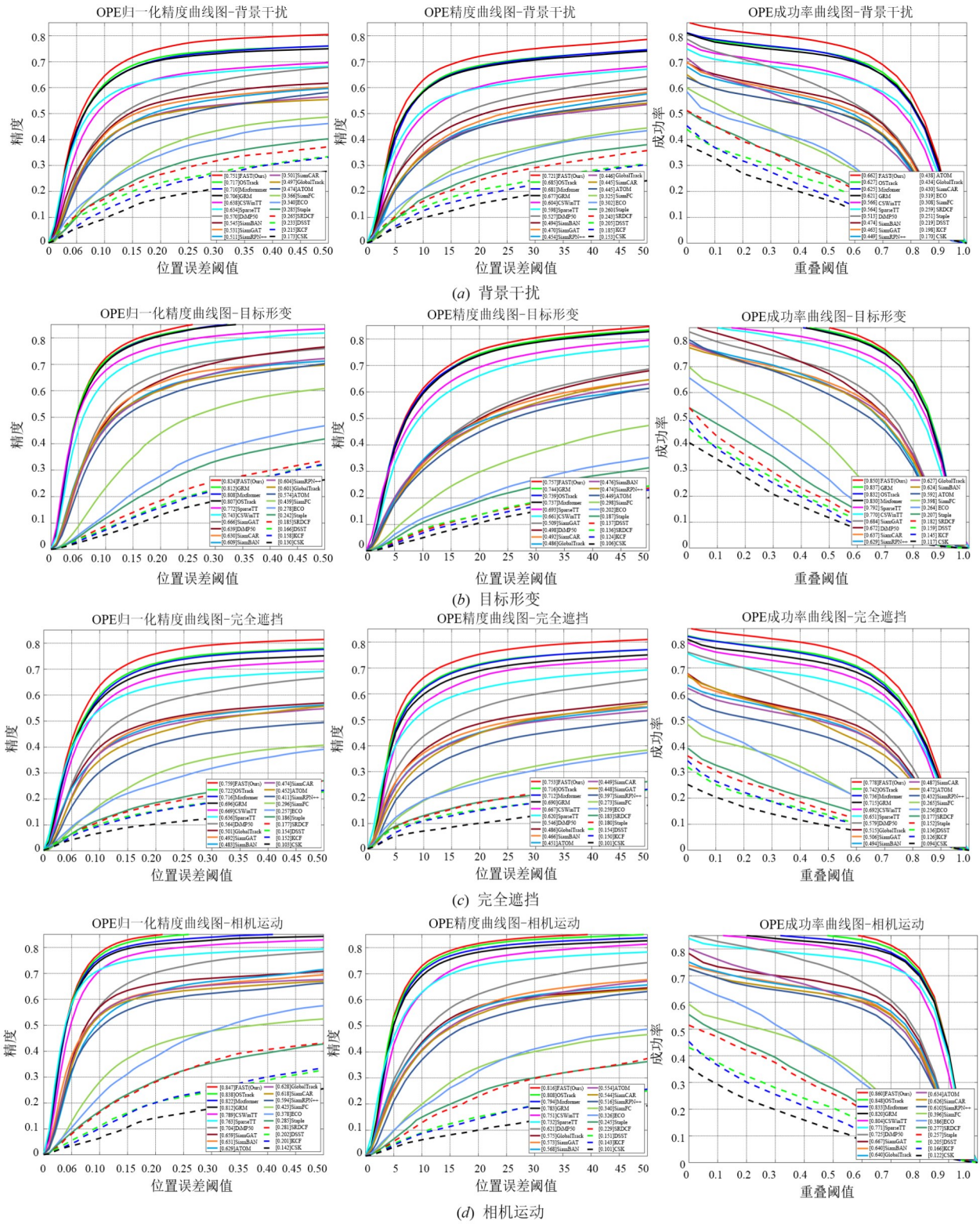


图 13 LaSOT 数据集上 4 个属性归一化精度、精度和成功率对比结果

法的优势,通过引入模板特征过滤器,筛选和迭代历史帧信息,生成最佳模板特征,有效缓解历史信息冗余和干扰的问题,显著提升跟踪器应对目标变化的性能;此外,设计的搜索特征划分器能够动态区分相关特征与独立特征,降低了不恰当特征关联带来的负面影响,缓解背景干扰下信息聚焦不准确的问题;引入的分离注意力

机制编码器,结合时空信息令牌,在目标发生遮挡或超出视野时可以缓解跟踪框的漂移,确保跟踪的准确性.

3.5 UAV123数据集实验结果

UAV123数据集[27]包含了123个长视频序列,涵盖不同的跟踪场景和复杂环境.每个视频序列的长度和拍摄场景有所不同,都包含了由无人机视角捕捉的目标物体.这些视频中的目标物体一般具有动态变化的运动轨迹,且在相机运动、场景遮挡、环境变化等因素的影响下,跟踪任务变得更加复杂.

对于UAV123数据集,定量分析的对比算法包括AIA[18]、SeqTrack[45]、TransT[13]这3种近几年优秀的代表性跟踪器;Mixformer[17]中提出了一个与本文类似的模板动态更新策略;OSTrack[20]、CSWinTT[24]分别采用削弱背景和增强前景的方式削弱噪声区域交互;SiamGAT[19]、SiamCAR[9]、SiamRPN++[8]分别为高性能、无锚和有锚孪生网络跟踪器;TCTrack++[46]和HiFT[47]为专用于UAV序列的跟踪算法;KCF[35]为工业界应用最广泛的相关滤波算法.本文在UAV123数据集的12种属性下与选定跟踪器进行了全面对比,成功率和精度结果分别如表4和表5所示.

表4 UAV123数据集成功率结果　　单位:%

算法	VC	SV	SOB	POC	OV	LR	Ⅳ	FOC	FM	CM	BC	ARC	Overall
KCF	28.2	30.7	37.9	32.1	25.7	20.2	29.6	21.0	18.7	34.7	34.1	26.9	36.9
SiamRPN++	76.0	71.1	63.9	63.8	68.5	52.8	72.3	44.7	64.8	74.8	55.9	68.6	74.4
SiamCAR	76.2	73.5	66.0	64.3	65.7	59.9	71.8	49.6	67.7	74.7	59.9	69.9	76.3
SiamGAT	79.9	72.2	74.8	71.2	75.2	65.3	72.8	55.5	70.8	81.1	58.6	76.6	79.7
HiFT	72.0	70.5	62.5	59.7	64.3	53.6	62.0	42.7	69.3	74.3	48.3	66.4	72.9
TCTrack++	62.5	61.1	55.6	53.3	57.6	41.9	51.8	35.5	47.5	60.7	43.5	58.1	64.0
TransT	81.2	78.3	75.7	72.1	74.3	65.7	73.3	49.6	74.1	80.7	54.5	75.4	80.7
OSTrack	86.4	81.1	81.1	77.1	79.8	68.4	75.7	60.5	74.9	84.1	61.4	80.4	83.2
SeqTrack	86.9	81.8	81.9	77.7	78.5	69.8	77.0	58.9	75.9	85.3	60.9	79.9	83.6
CSWinTT	87.8	83.5	84.3	80.5	79.2	68.8	80.9	65.7	80.4	88.5	68.2	85.1	85.3
AIA	88.4	83.7	82.4	79.9	79.1	71.0	83.4	65.5	82.4	88.1	71.0	84.5	85.5
Mixformer	88.6	83.8	84.7	80.3	81.0	69.1	81.9	65.4	81.1	89.3	69.4	84.8	85.5
本文算法	88.4	84.1	84.7	80.1	79.6	68.6	81.5	65.5	82.5	89.1	69.2	85.0	85.4

表5 UAV123数据集精度结果　　单位:%

算法	VC	SV	SOB	POC	OV	LR	IV	FOC	FM	CM	BC	ARC	Overall
KCF	43.6	47.1	57.8	45.1	38.6	38.1	41.8	37.4	29.6	48.3	45.4	42.4	52.3
SiamRPN++	81.8	77.9	72.0	73.3	79.2	65.9	77.8	61.4	72.4	81.7	64.0	76.6	80.4
SiamCAR	78.4	78.1	71.7	70.5	71.7	69.5	76.2	62.8	74.1	78.7	67.7	74.5	80.4
SiamGAT	83.5	82.4	81.5	78.1	81.3	73.6	77.9	69.0	78.6	86.7	65.7	82.3	84.3
HiFT	76.4	76.8	71.3	68.4	70.0	65.5	70.0	58.6	77.8	77.9	59.4	73.3	78.7
TCTrack++	70.4	70.9	65.9	63.9	64.4	62.0	63.7	54.7	58.9	71.1	59.1	67.6	73.1
TransT	85.0	83.3	82.3	76.8	79.8	75.5	78.4	62.3	80.9	85.2	60.8	80.9	85.2
OSTrack	89.2	87.4	88.8	85.3	87.0	79.9	81.2	75.2	83.0	89.5	68.5	87.3	88.8
SeqTrack	91.4	87.9	89.3	86.0	86.5	81.0	82.8	74.0	84.2	90.7	68.4	86.9	89.1
CSWinTT	91.2	89.0	91.4	87.8	84.7	79.7	87.1	80.2	88.0	93.2	76.1	91.1	90.3
AIA	91.8	89.5	89.5	87.1	84.3	82.7	89.9	79.7	90.1	92.7	80.0	90.2	90.7
Mixformer	92.8	89.8	92.1	87.8	87.7	80.3	88.4	78.8	88.7	95.0	77.8	91.2	90.9
本文算法	91.6	91.0	92.3	87.6	87.5	79.9	88.1	80.5	90.1	95.2	78.3	91.3	91.2

分析实验结果可知,本文算法在成功率和精度上分别达到了85.4%和91.2%,精度为所有对比算法中最高,成功率略低于AIA[18]和Mixformer[17]位居第2,整体表现验证了在UAV123数据集上的优越性能.在成功率上,FAST跟踪器在尺度变化(Scale Variation,SV)、相似性干扰(Similar OBject,SOB)、快速运动(Fast Motion,FM)这3项挑战上取得领先结果,在视角变化(Viewpoint Change,VC)、超出视野(Out of View,OV)、完全遮挡FOC、相机运动(Camera Motion,CM)和宽高比变化(Aspect Ratio Change,ARC)这5项挑战取得了第2的排

名；在精度上，FAST 跟踪器在部分遮挡（Partial OCclusion，POC）、超出视野 OV 和背景杂乱（Background Clutter，BC）这 3 个属性上取得第 2 名，在尺度变化 SV、相似性干扰 SOB、完全遮挡 FOC、快速运动 FM、相机运动 CM、宽高比变化（Aspect Ratio Change，ARC）这 6 个属性上的精度值排名第 1.

以上数据充分说明了本文跟踪器在处理复杂动态场景和目标变化时具有较强的鲁棒性，尤其在涵盖相机视角变化或目标本身变化这一情况时表现出色，引入了具有时域信息的令牌传播机制，跟踪器也能在一定程度上提升遮挡、目标消失等场景下的跟踪性能. 然而，本文算法及 Mixformer[17]、CSWinTT[24]等需要依靠高精度视觉特征的跟踪器在低分辨率场景下表现不佳，在低分辨率情况下目标表征缺乏足够细节，使跟踪器难以提取具有区分性的特征导致特征表达能力下降，并且跟踪器更新模板时可能会受到模糊特征的干扰，导致误更新或累积误差，从而使跟踪目标漂移或丢失. 仅有 SeqTrack[45]跟踪器通过将跟踪转化为文本坐标生成减轻了视觉特征的依赖性，在低分辨率场景下取得优秀表现. 总体来说，本文算法在 UAV123 的其他多种属性上表现优异，均取得了有竞争力的得分，具有较强的鲁棒性和准确性.

本文在对比算法中选取除了刻意轻量化的 TCTrack++[46] 和 HiFT[47] 以外的各类深度跟踪器在 UAV123 数据集上进行跟踪速度比较，成功率与跟踪速度比较的可视化结果如图 14 所示. 不同颜色形状的图形代表不同的跟踪器，红色虚线为实时线，如果算法位于实时线右侧，说明算法能满足大于 24 帧的实时应用需求；位于实时线左侧，则算法无法满足实时性要求；图形越靠近图中右上方的位置代表跟踪器的性能越好. FAST 算法的运行速度为 42 FPS，运行速度显著优于其他高精度的深度跟踪器；FAST 在图中位置最接近右上角，说明本文算法的综合性能较为优异，能够实现成功率与实时性的良好平衡.

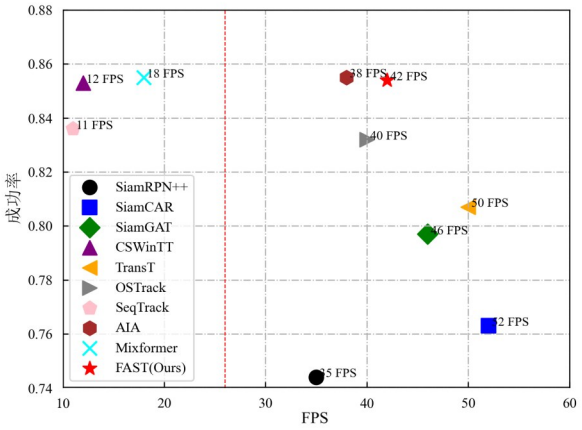


图 14 UAV123 数据集跟踪成功率与速度比较

3.6 其他数据集实验结果

为进一步分析 FAST 算法在多种场景下的表现，在 GOT-10K[35]与 TrackingNet[36]数据集上与多种近年来的先进跟踪器进行了对比实验，重点针对与本文算法相似的 Transformer 跟踪器进行了分析，表 6 展示了选定对比跟踪器在 GOT-10K 与 TrackingNet 数据集上的跟踪结果.

表 6 GOT-10K 与 TrackingNet 数据集的跟踪结果 单位：%

跟踪器	作者及年份	GOT-10K			TrackingNet		
		AO	$SR_{0.5}$	$SR_{0.75}$	AUC	P_{norm}	P
TransT[13]	Chen et al. 2021	67.1	76.8	60.9	81.4	86.7	80.3
STARK[16]	Yan et al. 2021	68.8	78.1	64.1	81.3	86.1	—
MixFormer[17]	Cui et al. 2022	70.7	80.0	67.8	83.1	88.1	81.6
AIATrack[18]	Gao et al. 2022	69.6	80.0	63.2	82.7	87.8	80.4
OSTrack[20]	Ye et al. 2022	73.7	83.2	70.8	83.9	88.5	83.2
SeqTrack[45]	Chen et al. 2022	74.5	84.3	71.4	83.9	88.8	83.6
GRM[22]	Gao et al. 2023	73.4	82.9	70.4	84.0	88.7	83.3
ARTrack[48]	Wei et al. 2023	75.5	84.3	74.3	85.1	89.1	84.8
ODTrack[14]	Zheng et al. 2024	77.0	87.9	75.1	85.1	90.1	84.9
AQATrack[49]	Xie et al. 2024	76.0	85.2	74.9	84.8	89.3	84.3
FAST	Ours	74.7	84.1	72.6	84.6	89.4	84.6

GOT-10K：GOT-10K[35]是一个大规模、高多样性的通用目标跟踪基准，包含超过 10 000 个视频片段，涵盖了超过 560 种运动目标类别和 87 种运动模式. 在数据集划分上，GOT-10K 的训练集和测试集的目标类别完全不重叠，促进跟踪器在未见目标上的泛化能力. 评估指标方面，GOT-10K 采用了平均重叠率（Average Overlap，AO）和成功率（Success Rate，SR）来衡量跟踪器的性能，平均重叠率表示所有真实边界框与预测边界框之间的

平均重叠程度,而成功率则衡量重叠率超过0.5和0.75的帧的比例. 在GOT-10K数据集上,FAST算法展现出了卓越的综合性能,AO得分达到74.7%,超越了TransT、MixFormer和AIATrack等多数同类型Transformer跟踪器,性能接近最新跟踪器;在成功率方面,FAST在$SR_{0.5}$和$SR_{0.75}$上分别取得84.1%和72.6%的成绩,虽然并非最高水平,但仍然位居先列,具备一定的泛化性.

TrackingNet:TrackingNet[36]是一个大规模的通用目标跟踪数据集,包含超过30 000个视频片段,平均每个视频长度为16.6 s,总计超过$1\,400\times10^4$个密集标注的边界框. 数据主要来源于YouTube视频,经过筛选和处理,确保视频质量和标注的准确性. 训练集从YouTube-Bounding Boxes数据集中筛选而来,包含30 132个视频,测试集由511个新视频组成,标注过程结合自动跟踪和人工校正,提高标注的密度和准确性. 在TrackingNet数据集上,FAST算法的AUC得分达到84.6%,在所有对比方法中位列前茅,优于同类型的Transformer跟踪器SeqTrack和GRM,并与性能最优的ODTrack仅有0.5个百分点的微小差距;在P_{norm}和P指标上,FAST分别取得89.4%和84.6%,超越大多数对比方法,展现出更稳定的归一化精度表现. FAST在TrackingNet上不仅具备高度竞争力,在多个关键指标上均取得领先或接近最优的成绩,充分验证了其在复杂场景下的精确性和稳健表现.

VOT2020:VOT2020[50]包含60个精心挑选的视频序列,涵盖多样化的目标类别和复杂场景. 每个序列标注了目标的精确边界框. 为确保对每个序列的充分利用,VOT2020采用基于锚点的评估策略,在序列的多个预定义位置重新初始化跟踪器,而非仅在序列起始位置运行. 这种协议更全面地测试算法对初始位置变化的鲁棒性,避免单一初始化带来的偏差. 当跟踪器失败时,系统会自动在后续帧中重新初始化,评估算法的恢复能力. 表7展示了不同跟踪器在VOT2020数据集上的表现. FAST算法在多个评估指标上均优于现有方法,在主要指标EAO上,FAST达到59.1%,相比SeqTrack提升了6.9个百分点,相较于VOT2020挑战冠军PRT,提升幅度达到6.1个百分点. 同时,FAST在准确性方面达到76.3%,优于STARK和Mixformer等近期领先的Transformer跟踪器,展现了更强的空间定位能力;在鲁棒性方面,FAST也取得了87.2%的优异表现,为所有方法中最高,体现出其在处理遮挡、形变等复杂场景时的强大适应能力.

表7　VOT2020跟踪基准的比较　　单位:%

指标	SiamMask[51]	Ocean[43]	PRT[52]	STARK[16]	Mixformer[17]	SeqTrack[45]	FAST
EAO	32.1	43.0	53.0	50.5	53.5	52.2	59.1
Accuracy	62.4	69.3	70.0	75.9	76.1	—	76.3
Robustness	64.8	75.4	86.9	81.9	85.4	—	87.2

3.7　速度、浮点数和参数分析

根据表8的比较数据,FAST算法的实际运行速度可达42帧/s,虽然其速度并非最快,但仍然保持了相对高效的运行速度,在速度、计算量和参数规模3个方面实现了良好平衡. FAST以42帧/s的运行速度,达到了接近实时的处理性能,优于STARK和MixFormer等方法,与OSTrack和GRM相当,表现出较强的推理效率;在计算复杂度方面,FAST的浮点运算量为53.7 GFLOPs,显著低于OSTrack和ODTrack,在保证性能的同时降低了计算开销. 尽管略高于STARK和MixFormer,但FAST在速度上的表现相对优异,显示出较高的性价比;此外,FAST的参数规模为68.5 M,处于中等水平,明显小于ODTrack和SeqTrack,相比STARK和MixFormer等模板更新方法,虽然参数略多,但换来更优异的性能表现. FAST在保持较低计算成本和适中模型规模的同时,提供了较高的运行速度,展现了良好的实际应用价值.

表8　速度、浮点数和参数比较结果

跟踪器	速度/fps	浮点数/G	参数/M
SiamPRN++[8]	35	48.9	54.0
STARK[16]	32	18.5	42.4
OSTrack[20]	41	65.3	—
SeqTrack[45]	15	148.0	89.0
Mixformer[17]	25	23.0	35.6
GRM[22]	45	—	—
ODTrack[14]	32	73.0	92.0
AQATrack[49]	44	58.3	72.0
FAST(Ours)	42	53.7	68.5

4　结论

本文提出一种能自适应选取模板特征与搜索特征的FAST算法,旨在通过减轻噪声区域的交互方式提升跟踪器应对形变、遮挡、背景复杂等多种实际挑战下的跟踪性能.

(1)提出一种模板特征过滤器,将图像级别的模板更新方式优化为特征级更新,自适应选取模板池的强

相关特征,并压缩弱相关特征;搜索特征鉴别器通过一个轻量的卷积网络选取特征中适合交互的潜在前景与无关背景.

(2)引入时空信息传播令牌,跨帧传递目标的序列特征,优化响应精度,设计了适配以上模块的分离注意力机制编码器,通过更灵活的交互方式提升了跟踪器的判别能力.

(3)在OTB100、LaSOT、UAV123等多个大规模公开数据集上进行广泛实验,实验结果表明,FAST在多个基准上具有较高的成功率和精度,有效解决目标发生遮挡、外观变化时引起的目标丢失和跟踪漂移问题.

本文算法和许多基于视觉特征的先进跟踪器均在低分辨率场景下表现不佳,为优化视觉跟踪算法,可以采用2种方法进行改进:一是引入多模态文本信息以丰富特征多样性;二是集成轻量化的超分辨率网络提高特征精度. 未来,将围绕这2个方向进行研究,促进该领域的进一步发展.

参考文献

[1] 孙家伟. 基于域不变投影的全天候目标跟踪方法研究[D]. 南京: 南京邮电大学, 2022.
SUN J W. Research on All-day Target Tracking Algorithm Based on Domain Invariant Projection[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2022. (in Chinese)

[2] 李淑慧, 邓志红, 冯肖雪, 等. 强杂波背景下基于变分贝叶斯推理的机载雷达目标跟踪算法[J]. 电子学报, 2022, 50(5): 1089-1097.
LI S H, DENG Z H, FENG X X, et al. Variational Bayesian inference-based airborne radar target tracking algorithm in strong clutter[J]. Acta Electronica Sinica, 2022, 50(5): 1089-1097. (in Chinese)

[3] 钟钰彬, 杨鹏, 窦磊. 基于纵横比自适应的相关滤波跟踪算法[J]. 电子学报, 2024, 52(6): 2112-2122.
ZHONG Y B, YANG P, DOU L. Correlation filtering tracking algorithm based on adaptive aspect-ratio[J]. Acta Electronica Sinica, 2024, 52(6): 2112-2122. (in Chinese)

[4] 姜珊, 底晓强, 韩成. 融合时空特性的孪生网络视觉跟踪[J]. 兵工学报, 2021, 42(9): 1940-1950.
JIANG S, DI X Q, HAN C. Siamese network for visual tracking with temporal-spatial property[J]. Acta Armamentarii, 2021, 42(9): 1940-1950. (in Chinese)

[5] 才华, 王学伟, 付强, 等. 基于动态模板更新的孪生网络目标跟踪算法[J]. 吉林大学学报(工学版), 2022, 52(5): 1106-1116.
CAI H, WANG X W, FU Q, et al. Siamese network target tracking algorithm based on dynamic template updating[J]. Journal of Jilin University (Engineering and Technology Edition), 2022, 52(5): 1106-1116. (in Chinese)

[6] 谢青松, 刘晓庆, 安志勇, 等. 基于前景优化的视觉目标跟踪算法[J]. 电子学报, 2022, 50(7): 1558-1566.
XIE Q S, LIU X Q, AN Z Y, et al. Visual object tracking algorithm based on foreground optimization[J]. Acta Electronica Sinica, 2022, 50(7): 1558-1566. (in Chinese)

[7] BERTINETTO L, VALMADRE J, HENRIQUES J F, et al. Fully-convolutional Siamese networks for object tracking[C]//European Conference on Computer Vision. Cham: Springer, 2016: 850-865.

[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. Piscataway: IEEE, 2019: 4277-4286.

[9] GUO D Y, WANG J, CUI Y, et al. SiamCAR: Siamese fully convolutional classification and regression for visual tracking[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2020: 6268-6276.

[10] CHEN Z D, ZHONG B N, LI G R, et al. Siamese box adaptive network for visual tracking[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2020: 6667-6676.

[11] 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. Piscataway: IEEE, 2019: 4655-4664.

[12] ZHANG Z P, LIU Y H, WANG X, et al. Learn to match: Automatic matching network design for visual tracking[C]// 2021 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2021: 13319-13328.

[13] CHEN X, YAN B, ZHU J W, et al. Transformer tracking[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2021: 8122-8131.

[14] ZHENG Y Z, ZHONG B N, LIANG Q H, et al. ODTrack: Online dense temporal token learning for visual tracking[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 38(7): 7588-7596.

[15] HUANG L H, ZHAO X, HUANG K Q. GlobalTrack: A simple and strong baseline for long-term tracking[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 11037-11044.

[16] 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 and Pattern Recognition. Piscataway: IEEE, 2021: 10428-10437.

[17] CUI Y T, JIANG C, WANG L M, et al. MixFormer: End-to-end tracking with iterative mixed attention[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2022: 13598-13608.

[18] GAO S Y, ZHOU C L, MA C, et al. AIATrack: Attention in attention for transformer visual tracking[C]//17th European Conference on Computer Vision. Cham: Springer, 2022: 146-164.

[19] GUO D Y, SHAO Y Y, CUI Y, et al. Graph attention tracking[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2021: 9538-9547.

[20] YE B T, CHANG H, MA B P, et al. Joint feature learning and relation modeling for tracking: A one-stream framework[C]//17th European Conference on Computer Vision. Cham: Springer, 2022: 341-357.

[21] FU Z H, FU Z H, LIU Q J, et al. SparseTT: Visual tracking with sparse transformers[EB/OL]. (2022-05-08)[2025-01-01]. https://arxiv.org/abs/2205.03776.

[22] GAO S Y, ZHOU C L, ZHANG J. Generalized relation modeling for Transformer tracking[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2023: 18686-18695.

[23] YANG K, ZHANG H J, SHI J Y, et al. BANDT: A border-aware network with deformable transformers for visual tracking[J]. IEEE Transactions on Consumer Electronics, 2023, 69(3): 377-390.

[24] SONG Z K, YU J Q, CHEN Y P, et al. Transformer tracking with cyclic shifting window attention[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2022: 8781-8790.

[25] WU Y, LIM J, YANG M H. Object tracking benchmark[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834-1848.

[26] 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. Piscataway: IEEE, 2019: 5369-5378.

[27] MUELLER M, SMITH N, GHANEM B. A Benchmark and simulator for UAV tracking[M]//Computer Vision - ECCV 2016. Cham: Springer International Publishing, 2016: 445-461.

[28] FU Z H, LIU Q J, FU Z H, et al. STMTrack: Template-free visual tracking with space-time memory networks[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2021: 13769-13778.

[29] 刘广文, 谢欣月, 付强, 等. 基于时空模板焦点注意的Transformer目标跟踪算法[J]. 吉林大学学报(工学版), 2025, 55(3): 1037-1049.
LIU G W, XIE X Y, FU Q, et al. Spatiotemporal Transformer with template attention for target tracking[J]. Journal of Jilin University (Engineering and Technology Edition), 2025, 55(3): 1037-1049. (in Chinese)

[30] DANELLJAN M, HÄGER G, KHAN F S, et al. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2016: 1430-1438.

[31] ZHOU X, GUO P, HONG L, et al. Reading relevant feature from global representation memory for visual object tracking[J]. Advances in Neural Information Processing Systems, 2023, 36: 10814-10827.

[32] HE K M, CHEN X L, XIE S N, et al. Masked autoencoders are scalable vision learners[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2022: 15979-15988.

[33] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[EB/OL]. (2020-12-22) [2024-12-20]. https://arxiv.org/pdf/2010.11929/1000.

[34] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft COCO: Common objects in context[M]//Computer Vision - ECCV 2014. Cham: Springer International Publishing, 2014: 740-755.

[35] 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.

[36] MÜLLER M, BIBI A, GIANCOLA S, et al. TrackingNet: A Large-scale dataset and benchmark for object tracking in the wild[M]//Computer Vision-ECCV 2018. Cham: Springer International Publishing, 2018: 310-327.

[37] DANELLJAN M, HÄGER G, KHAN F S, et al. Learning spatially regularized correlation filters for visual tracking[C]//2015 IEEE International Conference on Computer Vision. Piscataway: IEEE, 2015: 4310-4318.

[38] HENRIQUES J F, CASEIRO R, MARTINS P, et al. Exploiting the circulant structure of tracking-by-detection with kernels[M]//Computer Vision-ECCV 2012. Berlin: Springer, 2012: 702-715.

[39] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.

[40] DANELLJAN M, HÄGER G, SHAHBAZ KHAN F, et al. Accurate scale estimation for robust visual tracking[J].

Advances in Neural Information Processing Systems, 2023, 36: 10814-10827.
[41] BERTINETTO L, VALMADRE J, GOLODETZ S, et al. Staple: Complementary learners for real-time tracking[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2016: 1401-1409.
[42] 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. Piscataway: IEEE, 2017: 6931-6939.
[43] ZHANG Z P, PENG H W, FU J L, et al. Ocean: Object-aware anchor-free tracking[C]//16th European Conference on Computer Vision. Glasgow: Springer, 2020: 771-787.
[44] BHAT G, DANELLJAN M, VAN GOOL L, et al. Learning discriminative model prediction for tracking[C]//2019 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2019: 6181-6190.
[45] CHEN X, PENG H W, WANG D, et al. SeqTrack: Sequence to sequence learning for visual object tracking[C]// 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2023: 14572-14581.
[46] CAO Z A, HUANG Z Y, PAN L, et al. Towards real-world visual tracking with temporal contexts[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(12): 15834-15849.
[47] CAO Z A, FU C H, YE J J, et al. HiFT: Hierarchical feature transformer for aerial tracking[C]//2021 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2021: 15437-15446.
[48] WEI X, BAI Y F, ZHENG Y C, et al. Autoregressive visual tracking[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2023: 9697-9706.
[49] XIE J X, ZHONG B N, MO Z Y, et al. Autoregressive queries for adaptive tracking with spatio-temporal transformers[C]//2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2024: 19300-19309.
[50] KRISTAN W M, LEONARDIS A, MATAS J, et al. The eighth visual object tracking VOT2020 challenge results[C]// 16th European Conference on Computer Vision. Glasgow: Springer, 2020: 547-601.
[51] HU W M, WANG Q, ZHANG L, et al. SiamMask: A framework for fast online object tracking and segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(3): 3072-3089.
[52] MA Z A, WANG L Y, ZHANG H T, et al. RPT: Learning point set representation for Siamese visual tracking[C]// 16th European Conference. on Computer Vision. Glasgow: Springer, 2020: 653-665.

作者简介

王彩霞 女，1978年2月生，辽宁鞍山人. 长春理工大学副教授、硕士生导师. 主要从事智能信息处理技术、计算机视觉与目标跟踪、传感与信号处理等方面的研究.
E-mail: wcxhao@sina.com

周鸿策 男，2000年6月生，吉林松原人. 长春理工大学电子信息工程学院硕士研究生. 主要研究方向为计算机视觉和目标跟踪.
E-mail: zhccust@163.com

安 琪 女，1996年1月生，吉林松原人. 长春理工大学电子信息工程学院硕士研究生. 主要研究方向为计算机视觉和目标跟踪.
E-mail: 240404484@qq.com

李义鹏 男，2001年8月生，河南新乡人. 长春理工大学电子信息工程学院研究生. 主要研究方向为计算机视觉和图像超分辨率重构.
E-mail: 2337297789@qq.com