

分类号: TP391

密 级: 公开

学校代码: 10363

学 号: 2210110112



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题 目 高遮挡环境下动态视觉在线
多目标跟踪研究

论 文 作 者 李 艺

指 导 教 师 刘有余 (教授)

学 科 (专 业) 机械工程

研 究 方 向 机械电子工程

论文提交日期: 2024 年 05 月 31 日

分类号：TP391

单位代码：10363

密 级：公开

学 号：2210110112

高遮挡环境下动态视觉在线多目标跟踪研究

RESEARCH ON ONLINE MULTI-OBJECT TRACKING IN
DYNAMIC VISUAL SCENARIOS WITH HIGH
OCCLUSION

学生姓名：李艺

指导教师：刘有余 (教授)

专 业：机械工程

研究方向：机械电子工程

论文答辩日期：2024 年 05 月 20 日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在 ___ 年解密后适用本版权书。

本学位论文属于

不保密 ☒。

学位论文作者签名：李艺

日期：2024年05月25日

指导教师签名：刘有录

日期：2024年05月25日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所提交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：李艺

日期：2024 年 05 月 25 日

高遮挡环境下动态视觉在线多目标跟踪研究

摘要

视觉在线多目标跟踪(Multi-Object Tracking, MOT)是计算机视觉领域的研究热点,广泛应用于自动驾驶和人机交互等领域。然而,在实际应用中,面对高度遮挡、运动模糊和视角变化等复杂场景,目前的 MOT 方法表现不佳,限制了智能体在这些场景下的环境感知能力。本文从卡尔曼滤波器、外观特征模型、关联跟踪器等方面开展研究,提出了高遮挡环境下动态视觉在线 MOT 的综合应用方案。主要工作及取得的成果如下:

基于噪声尺度自适应策略,以运动匹配成本和目标置信度分别作为过程和观测噪声的尺度参数,并通过幂次变换构建非线性自适应因子,对不同轨迹的过程和观测协方差矩阵进行差异化调整,提出幂次-自适应卡尔曼滤波(Power-Adaptive Kalman Filter, PAKF)。实验显示其能有效提升基于运动的 MOT 算法应对目标非线性运动的鲁棒性。

在 OSNet 网络基础上,简化全尺度残差块 (Omni-Scale Residual Block, OSRB)结构,添加空间与信道注意力模块,引入多尺度特征融合机制,提出轻量级重识别(Re-Identification, Re-ID)网络 Fast OSNet,还融合表征与度量学习,为其设计联合损失函数,结合指数移动平均(Exponential Moving Average, EMA)和二段线性变换,提出自适应轨迹外观状态更新方法分层自适应指数移动平均(Hierarchical Adaptive Exponential Moving

Average, HAEMA), 综合二者为所提轻量级深度外观特征提取方案。将其与关联算法 BYTE 结合, 提出了 FDBTrack。在 MOT17 与 MOT20 测试集上的结果显示, 其可作为一种辅助手段改善在线 MOT 方法在高遮挡、动态场景下的跟踪性能。

引入 EMA 方法建模轨迹置信度, 并结合检测置信度构成优化权重, 为代价矩阵引入置信度线索, 构建两阶段的全检测数据关联框架 CWTracker。其无需调整置信度门限即可面对多种快速变化的复杂场景, 在高遮挡、动态场景中更具优势, 且跟踪速度优于主流关联算法 BYTE, 在无人汽车、无人机等边缘移动载体上具备应用价值。

结合所提 PAKF、外观特征提取方案与 CWTracker 得出用于高遮挡、动态环境下在线 MOT 的综合应用方案, 其能有效应对动态高遮挡场景下的 MOT 任务, 在 MOT20 验证集上则取得了最优性能, 且速度为 BOT-SORT 的 4.2 倍, 满足时间关键型任务需求。

关键词: 多目标跟踪, 自适应卡尔曼滤波, 重识别, 指数移动平均, 置信度优化

RESEARCH ON ONLINE MULTI-OBJECT TRACKING IN DYNAMIC VISUAL SCENARIOS WITH HIGH OCCLUSION

ABSTRACT

Visual online Multi-Object Tracking (MOT) is a research hotspot in computer vision, widely applied in autonomous driving and human-computer interaction. However, in practical applications, current MOT methods have shown unsatisfactory performance when confronted with complex scenarios involving high occlusion, motion blur, and viewpoint changes. These limitations restrict the environmental perception capabilities of intelligent agents in such scenarios. Therefore, researching dynamic visual online MOT algorithms in highly occluded scenarios holds significant value. This article conducts research on the Kalman filter, appearance feature models, and association algorithms, proposing a comprehensive application solution for dynamic visual online MOT with highly occluded environments. The main work and achievements are as follows:

Based on a noise-scale-adaptive strategy, a Power-Adaptive Kalman Filter (PAKF) was proposed by using motion matching costs and object confidences as scale parameters for process and observation noises. And then, two non-linear adaptive factors were constructed through power transformation to differentially adjust the process and observation covariance matrices for different trajectories. Experiments demonstrated that it significantly enhances the robustness of motion-based MOT algorithms against nonlinear movements.

In order to address the challenges of highly occluded and camera-perspective-dynamic environments, a lightweight re-identification (Re-ID) network called Fast OSNet was developed by simplifying the structure of the

Omni-Scale Residual Block (OSRB) and incorporating spatial and channel attention modules, as well as a multi-scale feature fusion mechanism. Representation learning and metric learning were integrated, and a joint loss function was designed for them. An adaptive trajectory appearance state update method, called Hierarchical Adaptive Exponential Moving Average (HAEMA), was constructed by combining Exponential Moving Average (EMA) based on exponential moving average and two-stage linear transformation. A lightweight deep appearance feature extraction scheme was proposed by synthesizing Fast OSNet and HAEMA. It was integrated with the association algorithm BYTE, leading to the proposal of FDBTrack. Results on the MOT17 and MOT20 benchmarks indicated that it could serve as an auxiliary method to improve the tracking performance of online MOT approaches under conditions of heavy occlusion and dynamic scenes.

The introduction of EMA to model trajectory confidence, combined with detection confidence, created optimized weights. These were integrated into the cost matrix, establishing a two-stage full-detection data association framework called CWTracker. Capable of adapting to various rapidly changing complex scenes without adjusting confidence thresholds, it excelled in high occlusion and dynamic scenes. Additionally, its tracking speed exceeded that of mainstream association algorithms like BYTE, offering application potential for edge mobile devices such as unmanned vehicles and drones.

By merging the proposed PAKF, appearance feature extraction approach, and CWTracker, a comprehensive solution for online MOT in challenging environments was developed. This effectively addressed MOT tasks in dynamic, highly occluded conditions, achieving top performance on the MOT20 validation set. It was 4.2 times faster than BOT-SORT, fulfilling the requirements for time-sensitive tasks.

KEY WORDS: Multi-Object Tracking, Adaptive Kalman Filtering, Re-identification, Exponential Moving Average, Confidence-Optimization

目录

摘 要.....	I
ABSTRACT.....	III
专用术语注释表.....	IX
第 1 章 绪论	1
1.1 研究背景与意义.....	1
1.2 研究现状.....	2
1.2.1 检测跟踪范式.....	2
1.2.2 深度外观特征提取器.....	3
1.2.3 MOT 中的数据关联研究.....	5
1.3 论文来源及研究目标.....	6
1.3.1 论文来源.....	6
1.3.2 研究目标.....	6
1.4 论文结构.....	6
第 2 章 幂次-自适应卡尔曼滤波在线多目标跟踪研究.....	9
2.1 定义与注释.....	10
2.2 幂次-自适应卡尔曼滤波	11
2.2.1 模型定义.....	11
2.2.2 状态预测.....	12
2.2.3 状态更新.....	13
2.3 算法流程.....	14
2.4 实验与分析.....	15
2.4.1 数据集与评价指标.....	15
2.4.2 实验设置.....	15
2.4.3 幂次参数优化.....	16

2.4.4 消融实验.....	17
2.4.5 复杂场景评估.....	18
2.4.6 基准评估.....	19
2.4.7 定性分析.....	20
2.5 本章小结.....	21
第 3 章 融合轻量级深度外观特征提取的在线多目标跟踪研究.....	22
3.1 Fast OSNet.....	22
3.1.1 Fast OSNet 的结构	22
3.1.2 联合损失函数.....	24
3.2 HAEMA.....	26
3.3 FDBTrack.....	27
3.4 实验与分析.....	28
3.4.1 数据集与评价指标.....	28
3.4.2 实验设置.....	29
3.4.3 Re-ID 网络消融及对比实验.....	29
3.4.4 MOT 消融实验.....	31
3.4.5 网络替换实验.....	31
3.4.6 对比实验.....	34
3.4.7 基准评估.....	34
3.4.8 定性分析.....	36
3.5 本章小结.....	37
第 4 章 置信度优化全检测关联在线多目标跟踪研究.....	38
4.1 轨迹置信度.....	38
4.2 CWTracker.....	39
4.3 CWTracker++	43
4.4 实验与分析.....	44
4.4.1 实验设置.....	44

4.4.2 消融实验.....	44
4.4.3 对比实验.....	47
4.4.4 基准评估.....	49
4.4.5 定性分析.....	51
4.5 本章小结.....	52
第 5 章 综合应用与实验验证.....	53
5.1 综合应用.....	53
5.2 实验验证.....	55
5.2.1 实验设置.....	55
5.2.2 消融实验.....	55
5.2.3 对比实验.....	56
5.2.4 定性分析.....	57
5.3 本章小结.....	61
第 6 章 结论与展望.....	62
6.1 结论.....	62
6.2 创新点.....	63
6.3 未来展望.....	64
参考文献.....	65
攻读学位期间成果目录.....	73

专用术语注释表

缩略词	英文全称	中文全称
MOT	Multi-Object Tracking	多目标跟踪
PAKF	Power-Adaptive Kalman Filter	幂次-自适应卡尔曼滤波
OSRB	Omni-Scale Residual Block	全尺度残差块
Re-ID	Re-Identification	重识别
EMA	Exponential Moving Average	指数移动平均
HAEMA	Hierarchical Adaptive Exponential Moving Average	分层自适应指数移动平均
KF	Kalman filter	卡尔曼滤波
SOTA	State-Of-The-Art	当今最好
VOT	Visual Object Tracking	视觉目标跟踪
SOT	Single-Object Tracking	单目标跟踪
TBD	Tracking-By-Detection	检测跟踪
IOU	Intersection-Over-Union	交并比
HA	Hungarian Algorithm	匈牙利算法
JDT	Joint Detection and Tracking	联合检测与跟踪
JDE	Joint Detection and Embedding	联合检测与嵌入
CMC	Camera Motion Compensation	相机运动补偿
CNN	Convolutional Neural Network	卷积神经网络
DAE	Deep Appearance Extractor	深度外观提取器
JPDA	Joint Probability Data Association	联合概率数据关联
MHT	Multiple Hypothesis Tracking	多假设跟踪
LSTM	Long Short-Term Memory	长短期记忆网络
AKF	Adaptive Kalman Filter	自适应卡尔曼滤波
NSAKF	Noise Scale Adaptive Kalman Filter	噪声尺度自适应卡尔曼滤波模型
PA-P	Power-Adaptive Prediction	幂次-自适应预测

PA-U	Power-Adaptive Update	幂次-自适应更新
MOTA	Multi-Object Tracking Accuracy	多目标跟踪准确度
MOTP	Multi-Object Tracking Precision	多目标跟踪精度
IDs	ID Switch	身份切换
Frag	Fragment	轨迹碎片
FP	False Positive	假阳性
FN	False Negative	假阴性
IDF1	Identification F1-Score	IDF1 分数
HOTA	High Order Tracking Accuracy	高阶跟踪准确度
AssA	Association Success	关联成功度
MT	Mostly Tracked	长期跟踪
ML	Mostly Lost	长期丢失
FPS	Frames Per Second	每秒帧数
FDBTrack	Fast Deep BYTE Track	快速深度 BYTE 跟踪
SOSRB	Simple Omni-Scale Residual Block	简化全尺度残差块
DWConv	Depthwise Convolution	逐深度卷积
AG	Aggregation Gate	聚合门
CBAM	Convolutional Block Attention Module	卷积块注意力模块
PAP	Path-level Average Pooling	部件级平均池化
GAP	Global-level Average Pooling	全局级平均池化
Rank-1	Rank-1 Accuracy	首位命中率
mAP	mean Average Precision	平均精度均值
FLOPs	FLoating point OPerationS	浮点运算次数
CAM	Class Activation Map	类激活图
FCW	First-stage Confidence Weight	一阶段置信度权重
SCW	Second-stage Confidence Weight	二阶段置信度权重

第 1 章 绪论

1.1 研究背景与意义

随着硬件算力的提升与大数据时代的到来，世界正面临着从信息时代到人工智能时代的重大跨越，人类正赋予机器感知世界和自主决定的能力，而在线 VOT 技术^[1]正是机器感知世界的基础之一。具体而言，VOT 技术就是应用机器视觉、贝叶斯滤波、机器学习和模式识别等技术，对视觉传感器取得的时间维度视频帧序列进行分析，得出感兴趣目标完整运动轨迹的过程。

根据跟踪目标的数量不同，VOT 可以分为 SOT^[2]与 MOT^[3]两类。前者旨在跟踪视频序列中的单个预定义目标，其利用运动模型生成候选框并预测目标位置，通过手工特征或深度特征建模目标外观，最终通过观测模型来更新目标模型以实现跟踪效果。而后者旨在同时跟踪视频序列中的多个目标，其首先检测同一帧中的所有感兴趣目标，进而利用运动模型或外观模型，提取每个目标的特征，并为所有目标分配单独的身份，再结合目标关联算法连接视频连续帧中的所有相同目标，以形成多个完整的目标运动轨迹^[4]。MOT 作为机器视觉领域的研究热点，在智能视频监控^[5]、自动驾驶^[6]、人机交互^[7]和位姿识别^[8]领域有着广泛的应用。然而，较之 SOT，MOT 任务中存在的目标数量不固定、同类目标特征区分度不高、目标间相互遮挡和运动拍摄视角等难题一直困扰着其研究与应用。而且在实际应用场景中，MOT 算法需经常应对目标高度遮挡与动态拍摄的复杂情况，目标间的高度遮挡会增加目标检测步骤误检与漏检的风险，且目标间的相互重叠会干扰外观特征的提取^[9]，而相机运动又会与目标的运动耦合，从而影响运动预测的准确性^[10]。通常采用复杂的模型与精巧的算法能够有效提升 MOT 在此类场景下的性能，但会造成跟踪速度的显著下降^[11]。因此，在实际应用中，还需对考虑算法跟踪性能与跟踪速度间的权衡。

针对上述问题，课题瞄准高遮挡、动态环境下的视觉在线 MOT 算法，围绕运动预测模型、外观特征提取模型与数据关联算法等关键点开展研究，分别得出幂次-自适应卡尔曼滤波、多尺度特征融合轻量级外观特征提取与置信度优化全检测数据关联等

技术，并提出综合应用方案，对提升在线 MOT 在高遮挡、动态环境下的跟踪性能具有重要理论意义与应用价值。

1.2 研究现状

1.2.1 检测跟踪范式

TBD^[12]作为当前最有效的在线 MOT 算法范式，被工业界与学界广泛运用^[4,13-15]。其旨在收集当前帧内所有目标特征，计算它们与已知轨迹集合间的各类相似度，如 IOU^[13]、社群关系^[16]和外观相似度^[4]等，得出匹配代价矩阵，并利用数据关联算法寻求最小化成本匹配，以更新轨迹状态，如此往复，最终得出所有目标的完整轨迹，如图 1-1 所示。得益于深度学习检测器(如 YOLO^[17]、Faster R-CNN^[18]等)的日渐成熟，MOT 算法的跟踪精度与准确度已取得了长足的进步。然而，由于频繁的遮挡、相似的外观和快速运动的视角，高遮挡、动态场景下的鲁棒在线 MOT 仍是个挑战。

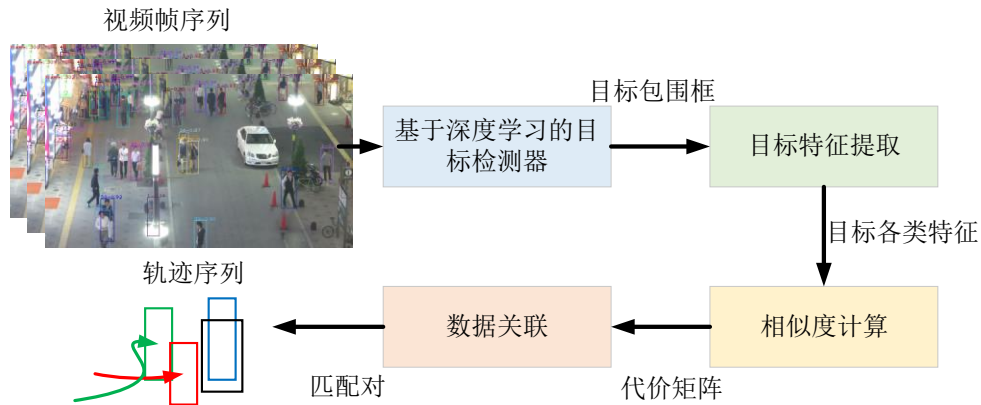


图 1-1 TBD 类算法范式流程图

Bewley 等人^[13]基于线性 KF^[19]、Faster R-CNN 和 HA^[20]，提出了一种基于运动的在线跟踪器 SORT，性能显著优于传统方法。然而，碍于检测器性能和简单运动模型限制，难以应对高遮挡场景中常见的长距离关联。为此，Wojke 等人^[4]引入外观线索和级联匹配，提出了 DeepSORT，有效提升了对轨迹的跟踪鲁棒性，但额外的外观嵌入过程大幅降低了关联速度。一些学者提出 JDT 范式，通过联合目标检测与后续跟踪任务，减少计算冗余。文献^[21]最早提出该思想，通过在两阶段 R-CNN 检测网络中添加多尺度特征图交互和偏移量回归支路的方法，实现了端到端的 MOT，有效提升了跟踪速度。Bergmann 等^[22]提出 Tracktor++，将区域建议网络生成的建议框用真实框与跟踪

框替代, 实现了基于跟踪框回归的 MOT, 重构关联代价函数, 实现了序列到序列的跟踪, 在当年 MOT 挑战赛表现出色, 但以上两种方法均未采用独立的运动预测模型, 导致其难以应对高遮挡环境下频繁的目标遮挡与交叉。此外, 还有一些学者尝试将检测与外观特征提取步骤联合, 融合外观嵌入^[23]网络和检测器为一个多任务模型, 提出 JDE 范式^[24], 达到了与基于运动的方法相近的关联速度。然而, 检测与嵌入任务对外观特征不同的需求, 干扰了模型训练过程^[14, 25, 26], 致使其跟踪性能有所损失。近期, 得益于检测器性能的快速提升, 基于运动的方法又受到了重视。ByteTrack^[27]在 SORT 的基础上, 增加了对低置信度检测的关联, 有效的从高遮挡目标中恢复了中断的轨迹。而 OC-SORT^[28]则以观测为中心, 重构了线性 KF 运动模型, 在非线运动场景中展现了强大的跟踪鲁棒性。与 SORT 类似, 二者都易于拓展。Aharon 等人^[11]基于 ByteTrack, 引入强外观嵌入模型 BOT^[29]和 CMC 模块, 提升了应对长期遮挡和相机运动的能力。Maggiolion 等人^[30]也基于 OC-SORT 进行了类似的工作。额外模块的加入显著的提升了方法在复杂场景的跟踪性能, 却也造成了跟踪速度的大幅下降。

综上, 采用 TBD 范式能通过方便的集成各类额外的模块, 提升 MOT 算法在高遮挡、动态环境下的跟踪性能, 例如当面对动态拍摄场景时可以加入相机补偿前处理, 当面对高遮挡场景时可以引入外观嵌入模型。然而, 过多的模块会大幅降低跟踪速度并提升算法部署难度。因此, 如何通过合理的模块设计均衡 MOT 算法的跟踪效果、跟踪速度与部署难度将成为一个研究重点。

1.2.2 深度外观特征提取器

随着 MOT 任务应用场景的复杂化, 仅提升检测器性能对整体 MOT 算法的改善已出现瓶颈。此外, 由于卡尔曼滤波的马尔科夫性, 导致基于运动方法难以应对长期的目标遮挡与动态的拍摄场景。针对以上问题, 许多学者通过引入行人 Re-ID 技术, 为跟踪过程引入额外的目标外观线索。Re-ID 可被视为一个特征嵌入问题, 即通过函数将高维的行人图像转化为低维特征向量^[31]。当前的主流方法是在图像分类网络的基础上, 结合特殊的损失函数, 如对比损失^[32]、ID 损失^[33]、三重损失^[34]和圆损失^[35]等, 实现特征嵌入学习。但同类目标间相似的外观、遮挡噪声和目标姿态不对齐等问题, 对网络提出了更高的要求。AlignedReID^[36]通过一个行人部位对齐算法, 缓解了目标姿

态不对齐造成的 Re-ID 性能下降, 但算法整体运行缓慢。PCB^[33]设计了多个局部特征输出分支, 并对每个分支单独训练, 显著提升了性能。ICNN^[37]在 PCB 的基础上, 融合了全局特征, 进一步提升了 Re-ID 性能。然而, 愈发增长的网络规模造成了应用部署的困难。Zhou 等人^[23]基于全尺度特征学习, 提出了一个多尺度特征融合的轻量级网络 OSNet, 取得了 SOTA 的 Re-ID 性能。但是复杂的块设计, 降低了其推理速度。

在 MOT 领域, 许多 TBD 类方法直接继承了 Re-ID 领域的研究成果。例如, Wojke 等人^[4]设计了一个简单的 CNN 作为 DAE, 将其引入 SORT 中, 提出 DeepSORT, 有效提升了算法的跟踪鲁棒性; He 等^[38]则通过 VGG 网络提取外观特征, 其跟踪准确度较 DeepSORT 有所提升, 身份切换发生次数大幅下降; 后续许多学者如 Mahmoudi^[16]、Bea^[15]和 Sheng^[39]等, 分别利用 GoogleNet、ResNet 和 InceptionNet 等网络作为 DAE, 并调整训练数据, 改进损失函数, 加深网络结构, 尝试提取更为鲁棒的表观特征; Du 等人^[40]提出的 GIAOTrack 直接使用了 OSNet 作为其 DAE; StrongSORT^[41]和 BOT-SORT 均使用了 BOT 中训练的 ResNest50^[42]网络。虽然, 引入 DAE 有效提升了 MOT 复杂场景下的跟踪性能。但是, DAE 的复杂化并未进一步提升算法鲁棒性, 反而造成运行速度大幅下降, 并提升了内存占用。在算力和内存受限的移动端设备中, 此类方法的缺点则更为凸显。此外, 在线 MOT 和 Re-ID 任务间的差异, 造成了二者对网络性能侧重的不同。Re-ID 任务训练样本少、类内相似度小、类间模糊性大, 要求网络能区分更细粒度的特征^[31], 而在线 MOT 任务则要求网络在保证性能的前提下, 推理速度更快, 内存占用更小。

针对 DeepSORT 类方法时效性差、部署困难的缺陷, JDE 类方法一度成为研究热点。Zhang 等人^[14]引入无锚框的 CenterNet 检测器, 提出 FairMOT, 提升了跟踪性能和关联速度。针对目标检测与 Re-ID 不同的特征需求, Liang 等人^[25]设计了一个特征解耦模块, Yu 等人^[26]则是引入了通道注意力机制, 二者都取得了一定提升。但这些方法的跟踪性能均低于同期的 DeepSORT 类方法。而要完全解决 JDE 类方法的问题, 需要多任务学习^[43]和少样本学习^[44]领域的进展。事实上, 随着检测器精度的提升、JDE 类方法的发展和强跟踪器^[27]的普及, DAE 已逐渐转为特殊场景下提升跟踪鲁棒性的辅助手段。因此, DAE 的设计应在提升外观嵌入鲁棒性的同时, 更需注重性能与计算开销间

的均衡。

1.2.3 数据关联

对于 MOT 任务而言，数据关联步骤成功关键在于如何正确的关联不可靠的检测与轨迹^[45]。数据关联可分为在线关联与离线关联，分别对应在线与离线 MOT 任务，其中离线关联可利用全局时空信息，关联准确度优于在线方法。然而，由于涉及全局的序列优化，其在时间关键型任务上不具备实用价值。目前大部分工作的研究重点均放在更具挑战性的在线关联上。

传统的数据关联方法如 JPDA^[46]和 MHT^[47]被设计用于对于雷达目标的跟踪，将其引入视觉 MOT 任务中效果并不理想。随着目标检测器和 TBD 范式的发展，SORT 所使用的数据关联范式逐渐成为主流，即通过计算当前检测框与历史轨迹间的各类相似度距离，构建关联成本矩阵，并利用 HA 得出检测与轨迹间的匹配关系。DeepSORT、Tracktor++和 TransTrack^[48]等算法均未脱离该范畴。通常情况下，这类方法是行之有效的，然在高遮挡环境下，目标间相似的空间位置和缺损的外观信息，导致代价矩阵变得不可靠，轨迹与目标匹配对间的模糊性增加，进而容易引发关联错误。Yoon 等^[49]基于 LSTM 在线学习帧间目标关联信息，有效提升数据关联性能，然该模型无法修复错位轨迹。齐美彬等^[50]提出一种双层数据关联策略，将相邻帧关联视为图分割过程，基于滑动时间窗口进行局部关联，在此基础上，结合时空特征实现全局关联，有效改善算法跟踪鲁棒性，但该方法存在后处理步骤，时效性不佳。BYTE^[27]引入高、低置信度阈值，将检测划分为高置信度、低置信度和不可靠集合，先匹配可靠的高置信度检测集合，再处理的低置信度检测，隔绝不可靠集合，有效提升了跟踪准确度和鲁棒性，但更多的匹配过程降低了 BYTE 的关联速度，并且低置信度目标的引入造成了跟踪精度的下降与轨迹切换发生率的提高。Bae 等人^[15]提出一种量化轨迹置信度的方法，将轨迹分为高、低置信度集合，并通过不同的线索构建独立的成本矩阵，有效缓解了匹配过程的模糊性。MOTDT^[51]则是利用外观线索进行首次关联，再根据位置线索实现二次关联。此外，Stadler 等人^[52]聚焦于匹配代价矩阵，通过引入相似性约束阈值，筛选模糊的匹配关系，并按特定策略单独处理，从另一个方向实现了多阶段的数据关联。

以上方法本质上均是通过人为设定的门限阈值划分匹配成本矩阵，将复杂的匹配

问题转化为多个简单的子问题，从而减轻模糊匹配的干扰。然而，固定的阈值难以适应 MOT 任务多样的场景，导致了算法跟踪性能的不稳定表现，尤其是在场景变化迅速的无人驾驶和无人机跟踪应用中，这些应用时常面对高遮挡、动态、运动模糊和小目标等问题。另一方面，相较于轨迹置信度，SOTA 方法们^[11, 27, 28, 30]通常更青睐于利用检测置信度来简化匹配问题。这是由于检测置信度可由检测器直接输出，而轨迹置信度则需要额外的计算过程和复杂的经验建模，会增加调试成本，并降低关联速度。但是，对于匹配过程而言轨迹与检测却是同等重要的，舍弃轨迹置信度蕴含的潜在信息，如轨迹的时空连续性和运动模式一致性，同样会增加身份切换发生的概率。

1.3 论文来源及研究目标

1.3.1 论文来源

来源 1：教育厅高校自然科学研究研究生项目“基于速度场的机械臂双目动态避障研究”(YJS20210446)，本人为项目负责人，且承当视觉智能动态环境感知系统研究。

来源 2：安徽工程大学研究生实践与创新项目“多特征融合注意力检测在线多目标跟踪装置研究”(YJS2023CX007)，本人为项目负责人，且承当多特征融合注意力外观特征提取网络与数据关联算法研究。

1.3.2 研究目标

面向高遮挡、动态场景下的视觉 MOT 任务，基于 TBD 范式特点，研究自适应线性 KF 理论，探索一种低计算开销的自适应多目标运动预测方法，提升算法对非线性运动的鲁棒性；剖析空间与信道注意力作用机制，引入人为先验，探索多尺度特征融合方法，构建针对 MOT 任务的轻量外观特征提取器，并设计融合表征学习与度量学习的联合损失函数，为提升在线 MOT 方法在复杂场景下的跟踪性能提供一种轻量级的辅助手段；研究鲁棒的轨迹置信度建模方法，结合目标置信度，探索一种基于置信度优化的全检测关联跟踪器，对破解长时间高遮挡目标的数据关联难题有科学意义。该项研究将为提升视觉 MOT 系统面对高动态、高遮挡、复杂随机开放场景下的应用性能提供新思路。

1.4 论文结构

论文结构共分为 6 章，其中第 2 章至第 5 章分别介绍所提的幂次-自适应卡尔曼滤

波器、多尺度融合轻量级外观特征提取方案、置信度优化全检测关联跟踪器和综合应用方案。研究方法及技术路线如下图 1-2 所示。

(1) 研究一种基于幂次-自适应卡尔曼滤波器的运动预测方法

针对多阶段关联 MOT 算法在高遮挡、动态环境下的局限性，受噪声尺度自适应卡尔曼滤波器启发，以运动匹配代价和置信度分数分别作为过程和观测噪声尺度参量，结合幂次变换构建自适应调节因子，对不同轨迹的过程和观测协方差矩阵进行差异化调整，构成幂次自适应预测步与更新步，二者组合构成幂次自适应卡尔曼滤波器。

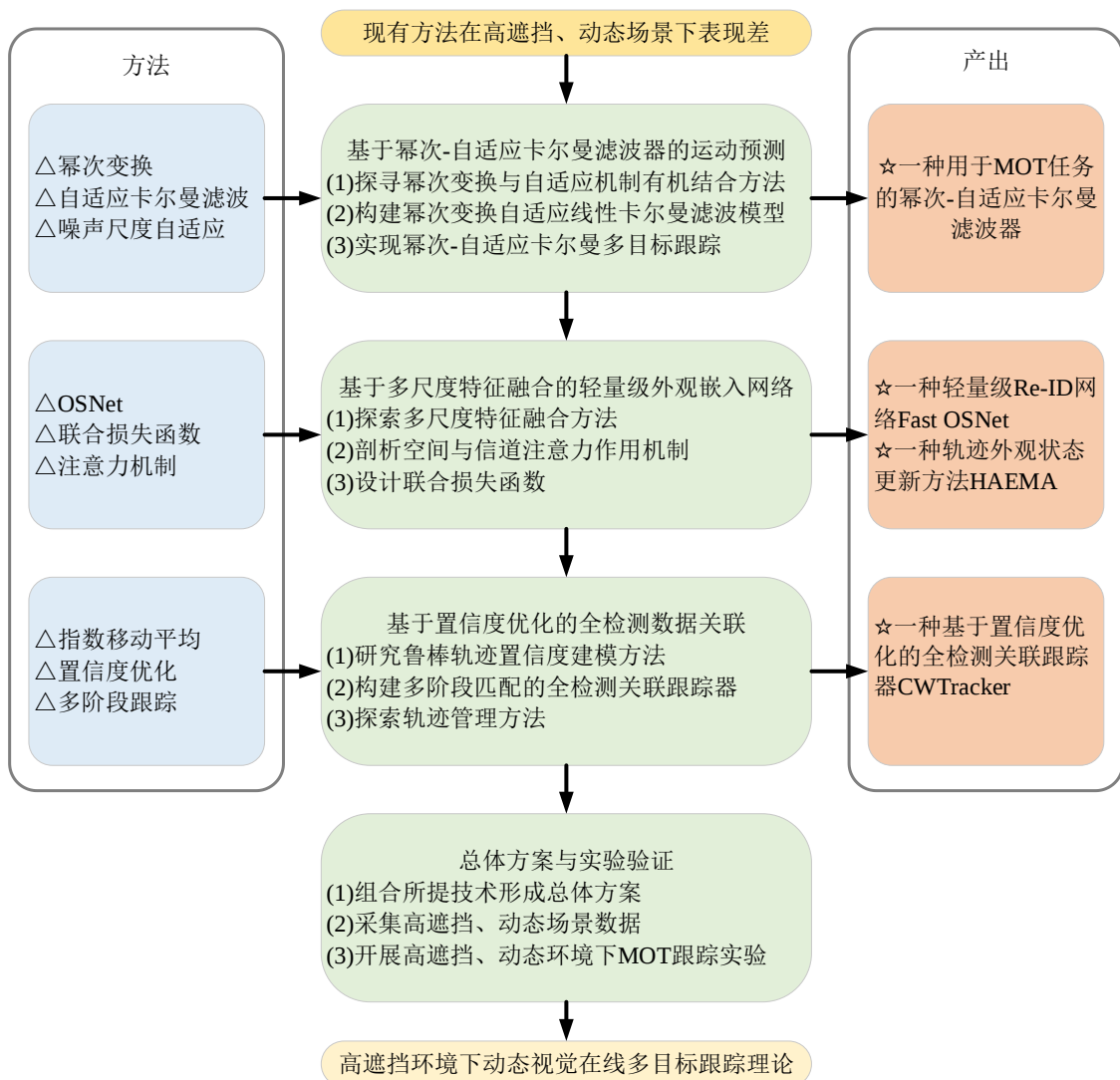


图 1-2 研究方法与技术路线

(2) 研究一种多尺度融合轻量级外观嵌入网络

现有用于 TBD 范式的深度外观提取方案难以平衡速度、性能和内存使用。针对

MOT 任务实际需求，通过简化 OSNet 网络结构，增加空间与信道注意力模块，并引入全局和局部级别的多尺度特征融合机制，提出了一种轻量级 Re-ID 网络 Fast OSNet。此外，为了提升轨迹外观状态对遮挡噪声的鲁棒性，在指数移动平均的基础上，引入了两阶段线性变换的自适应更新权重，提出了分层自适应指数移动平均 HAEMA。二者共同组成了所提轻量级方案。

(3) 研究一种基于置信度优化的全检测数据关联方法

类 BYTE 方法们大多依赖置信度门限阈值实现多阶段的数据关联。然而，人为设置的阈值会造成算法面对多变的高遮挡、动态场景时的不稳定。以检测置信度和匹配成本为参量，通过指数移动平均建模轨迹置信度，设计一种用于代价矩阵的置信度优化方法，为数据关联过程引入轨迹和检测的置信度线索，提出一种基于置信度优化的全检测关联跟踪器 CWTracker。

(4) 综合应用方案与实验验证

综合所提 PAKF、Fast OSNet 与 CWTracker，得出用于高遮挡、动态场景的视觉在线 MOT 综合应用方案。采集实际场景下的视频数据集，综合实际场景数据与公开数据集，开展消融实验与对比实验，以定量衡量各模块贡献，并同 SOTA 方法进行横向对比。最终，在实际场景与数据集场景上开展可视化的定性分析。

第 2 章 幂次-自适应卡尔曼滤波在线多目标跟踪研究

与传统 SORT 类方法相比，在数据关联中引入低分检测能有效提升算法的跟踪准确度与鲁棒性。但是，低分检测的引入同样会引起跟踪精度的下降，症结在于卡尔曼滤波固定的噪声设置难以均衡高、低分检测框的需求。此外，动态拍摄时变化视角，会导致卡尔曼滤波预测结果漂移，从而产生大量的轨迹身份切换。

在线 TBD 框架，一直是 MOT 领域的研究热点。MOTDT^[51]结合检测与跟踪结果作为观测候选，构建评价函数，选择最优观测，提升了跟踪性能，但只是回避了不可靠的检测，并未解决其造成的问题。Hua 等人^[53]尝试引入基于低秩的运动前、背景分割，以降低检测器误检，但额外的计算过程，使其难以满足实时要求。Zhang 等人^[27]研究发现大多数方法通过分数阈值滤去低分检测，导致大量不可恢复的轨迹中断与碎片，额外增加了低分检测框的匹配过程，取得了 SOTA 的跟踪性能。然其仍存在局限性，即所有轨迹共享一套 KF 模型。事实上，大部分方法^[4, 13, 28]均采取类似设计，然低分检测通常蕴含高水平的观测噪声^[51, 54]，若采取与高分检测一致的噪声参数设置将降低定位精度。此外，视角的突然变动将使预测结果大幅偏离真实值，从而引起定位精度下降，甚至造成轨迹中断与错接。部分研究尝试引入扩展卡尔曼滤波^[55]、无迹卡尔曼滤波^[56]乃至粒子滤波^[57]，以代替简单的线性 KF，然此类方法因线性化及采样过程，并不适用于在线应用场景。一些研究将 AKF 引入单目标跟踪任务^[58, 59]，取得了良好的效果，但此类算法存在额外的自适应因子计算步骤，将其直接用于 MOT 是昂贵且不明智的。另一方面，ByteTrack 与 OC-SORT^[28]的研究，证明遵循合理的关联策略，简单的线性 KF 也能取得 SOTA 性能。Zhang 等人^[40]在线性 KF 的基础上，引入检测置信度作为观测噪声尺度参量，提出一种 NSAKF 模型，其可有效提升 TBD 算法的定位精度，然其自适应调节因子呈线性变化，无法突出调节重点。

受 α -IOU^[60]与 NSAKF 启发，在线性 KF 中引入运动匹配代价、检测置信度分别作为过程、观测噪声尺度参量，并由幂次变换^[61]构造非线性自适应因子，以动态调整不同轨迹的过程、观测协方差矩阵，提出 PAKF 模型。其由 PA-P 与 PA-U 步骤组成。为验证所提 PAKF 性能，将其分解为单 PA-P、单 PA-U 与 PAKF 3 种方案。在 MOT17 验证集^[62]上，通过正交试验法分别优化 3 种方案的幂次参数，揭示了 PAKF 性能与幂次参数间的潜在变化关系。选择 ByteTrack 与 SORT 作为基线方法，将它们与所提 3 种方

案及 NSAKF 结合，在 MOT17 数据集上开展了消融实验，并对高遮挡、动态的复杂场景单独开展研究。在更具挑战性的 HiEve 基准数据集^[63]上，评估了所提的 PA-ByteTrack，并同其他先进方法进行了比较，以验证其有效性。另外，进行了定性分析，以演示所提方法的跟踪效果。

2.1 定义与注释

定义目标包围框为 $\mathbf{L}=[x, y, w, h]$ ，包围框速度为 $\dot{\mathbf{L}}=[\dot{x}, \dot{y}, \dot{w}, \dot{h}]$ 。 x 和 y 为包围框的中心点像素坐标， w 和 h 分别为包围框的宽和高，第 t 帧检测响应集合为 $\mathbb{D}_t = \{D_i^t\}_{i=1}^{n_t}$ ， n_t 为 t 时刻检测响应数，第 i 个检测响应为元组 $D_i^t = (\mathbf{L}_i^t, s_i^t)$ ， s_i^t 为 $\mathbf{L}_i^t$ 对应的置信度分数。第 t 帧的轨迹集合为 $\mathbb{T}_t = \{T_j^t\}_{j=1}^{m_t}$ ， m_t 表示第 t 帧存在的轨迹数量，第 j 条轨迹为集合 $T_j^t = \{\tau_j^k\}_{k=t_j^s}^t$ 。其中 t_j^s 指示轨迹初始帧。设轨迹 j 在第 k 帧的状态为元组，如式(2-1)所示。

$$\tau_j^k = (\mathbf{L}_j^k, \dot{\mathbf{L}}_j^k, \mathbf{P}_j^k, \kappa_j^{k+1}, l_j^k) \quad (2-1)$$

式中， $\mathbf{P}_j^k$ 为轨迹 j 在第 k 帧的状态估计协方差矩阵；

κ_j^{k+1} 为所提 PA-P 过程噪声自适应调节权重。

l_j^k 为轨迹 j 在第 k 帧的状态指示。

匹配是一个二部图概念^[13]，其为一个非邻边集合 $A_t = \{a_g^t\}_{g=1}^{\varepsilon_t}$ 。其中 ε_t 为边数， $a_g^t = (D_i^t, \tau_j^t)$ 。由 $\mathbb{D}_t$ 、 $\mathbb{T}_{t-1}$ 与带权边集 $E_t = \{e_g^t\}_{g=1}^{o_t}$ (o_t 为边数)，组成成本图 $\psi_t = (\mathbb{D}_t, \mathbb{T}_{t-1}, E_t)$ ，其具备多个匹配，由 HA 求得最优匹配 A_t^* 。由 A_t^* 和 $\mathbb{D}_t$ 构成 t 时刻观测集合 $\mathbb{Z}_t = \{Z_j^t\}_{j=1}^{h_t}$ ， h_t 为已关联轨迹数。设第 j 条轨迹在第 t 帧的观测如式(2-2)所示，

$$Z_j^t = (\mathbf{L}_j^t, s_j^t, \text{IOU}(a_j^t)) \quad (2-2)$$

式中，IOU 相似度计算如式(2-3)所示；

a_j^t 为一个容量为 2 的元组，代表成本图 ψ_t 的边。

$$\text{IOU}(a_j^t) = \frac{\text{Area}(\text{bbox}(\mathbf{L}_j^t) \cap \text{bbox}(\dot{\mathbf{L}}_j^t))}{\text{Area}(\text{bbox}(\mathbf{L}_j^t) \cup \text{bbox}(\dot{\mathbf{L}}_j^t))} \in [0, 1] \quad (2-3)$$

式中， $\text{bbox}(\cdot)$ 表示视频帧中的包围框。

幂次变换^[61]表达式如式(2-4)所示，

$$y = cx^\alpha + b \quad (2-4)$$

式中 $\alpha \in (0, +\infty)$ 为幂次参数。当 $\alpha = 1$ 时，式(2-4)等同于线性变换，不同的 α ，能使曲线展现出不同的性质。具体来说，通过调节 α ，幂次变换能突出感兴趣区间的，抑制不感兴趣区间。因此特性，其常被用于图像增强^[61, 64]领域。

2.2 幂次-自适应卡尔曼滤波

固定的噪声设置致使 KF 难以兼顾高、低置信度检测。普通 AKF 存在额外计算步骤，难以适应 MOT 任务。以 $\text{IOU}(a_g^t)$ 、 s_t 分别为过程、观测噪声参量，结合幂次变换，提出 PAKF。本节聚焦于单轨迹，如无特别说明，省略轨迹与观测序号。

所提 PAKF 整体框架如图 2-1 所示。首先，输入 $\mathbb{T}_{t-1}$ ，利用 PA-P 预测第 t 帧的目标估计状态向量 $\hat{\mathbf{x}}'_t$ 与状态估计协方差矩阵 $\mathbf{P}'_t$ 的预测值，将 $\hat{\mathbf{x}}'_t$ 传至数据关联算法；其次，计算预测框与检测框之间的 IOU 相似度，利用 HA 得出 A_t^* ，为每条轨迹分配对应的观测 Z_j^t ，并通过幂次变换更新 κ ；最后，PA-U 结合轨迹预测、观测和 ξ ，更新 $\hat{\mathbf{x}}_t$ 与 $\mathbf{P}_t$ ，生成 $\mathbb{T}_t$ 。

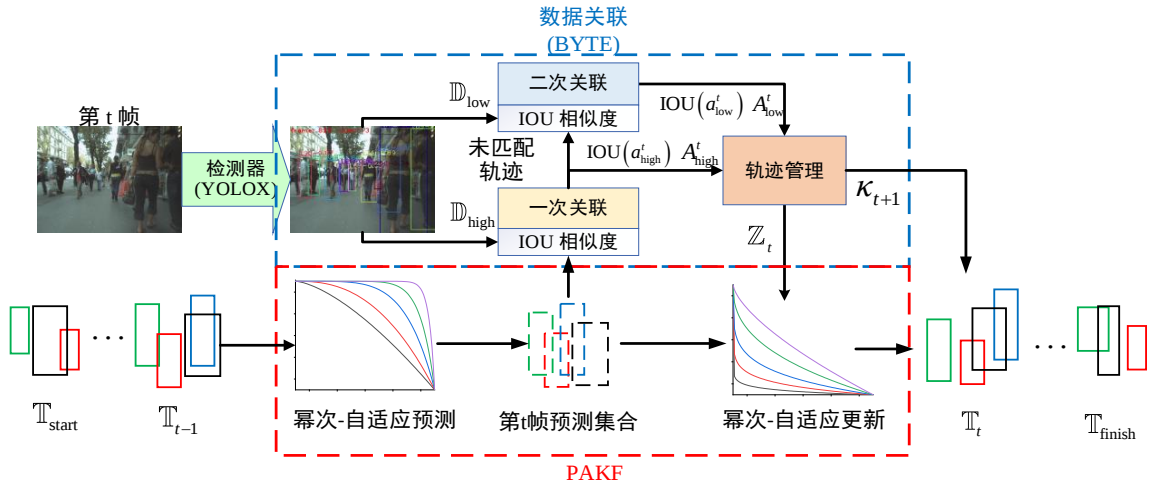


图 2-1 PAKF 整体框架

2.2.1 模型定义

由 τ_{t-1} 生成状态估计向量 $\hat{\mathbf{x}}_{t-1} = [\mathbf{L}_{t-1}, \dot{\mathbf{L}}_{t-1}]^T$ 、 κ_q^t 与 $\mathbf{P}_{t-1}$ ，由 Z_t 生成观测向量 $\mathbf{z}_t = [\mathbf{L}_t]^T$ 、 s_t 与 $\text{IOU}(a_j^t)$ 。所有目标初始速度为 0，并处于线性运动状态。检测器仅输出 $\mathbf{L}$ ，故定义状态过渡矩阵如式(2-5)所示，

$$\Phi_{8 \times 8} = \begin{bmatrix} \mathbf{1}_{4 \times 4} & \mathbf{1}_{4 \times 4} \\ \mathbf{0}_{4 \times 4} & \mathbf{1}_{4 \times 4} \end{bmatrix} \quad (2-5)$$

观测矩阵如式(2-6)所示,

$$\mathbf{H}_{4 \times 8} = [\mathbf{1}_{4 \times 4} \quad \mathbf{0}_{4 \times 4}] \quad (2-6)$$

所有目标均遵从式(2-7)的运动、观测模型,

$$\begin{cases} \mathbf{x}_t = \Phi \mathbf{x}_{t-1} + \boldsymbol{\mu} \\ \mathbf{z}_t = \mathbf{H} \mathbf{x}_t + \mathbf{v} \end{cases} \quad (2-7)$$

式中, $\boldsymbol{\mu}$ 为符合正态分布的 4 维列向量, 表示过程噪声;

$\mathbf{v}$ 为符合正态分布的 4 维列向量, 表示观测噪声。

过程协方差矩阵 $\mathbf{Q}$ 与观测协方差矩阵 $\mathbf{R}$, 如式(2-8)所示,

$$\begin{cases} \mathbf{Q}_{8 \times 8} = E[\boldsymbol{\mu} \boldsymbol{\mu}^T] \\ \mathbf{R}_{4 \times 4} = E[\mathbf{v} \mathbf{v}^T] \end{cases} \quad (2-8)$$

状态估计协方差矩阵如式(2-9)所示。

$$\mathbf{P}_{8 \times 8}^t = E[(\mathbf{x}_t - \hat{\mathbf{x}}_t)(\mathbf{x}_t - \hat{\mathbf{x}}_t)^T] \quad (2-9)$$

式中, $\mathbf{x}$ 为目标真实状态。

2.2.2 状态预测

所提 PA-P 过程如式(2-10)所示,

$$\begin{cases} \hat{\mathbf{x}}'_t = \Phi \hat{\mathbf{x}}_{t-1} \\ \mathbf{P}'_t = \Phi \mathbf{P}_{t-1} \Phi^T + \kappa_t \mathbf{Q} \end{cases} \quad (2-10)$$

其中, $\kappa_t \in (0,1)$ 为所提过程噪声自适应调节因子, 其计算过程如式(2-11)所示,

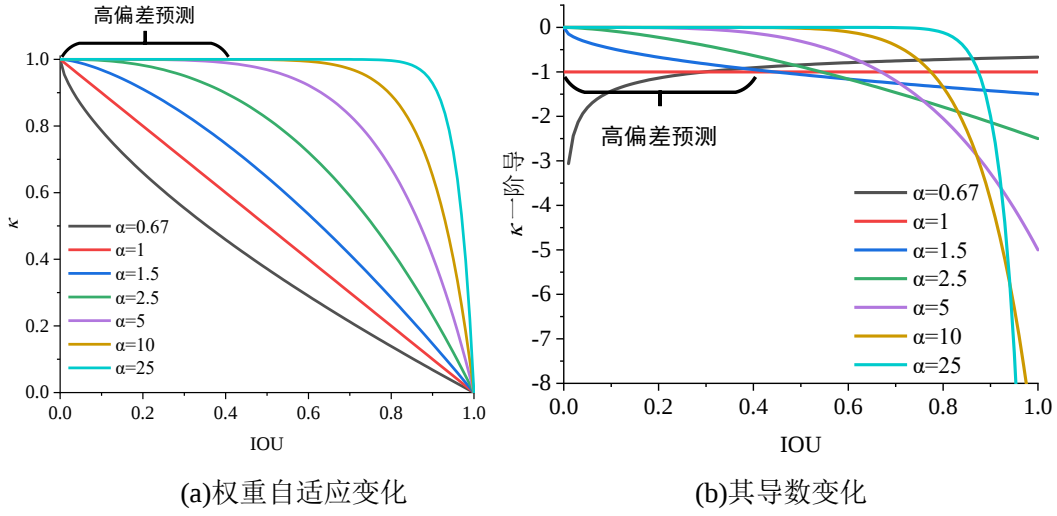
$$\kappa_t = 1 - \text{IOU}(a_j^{t-1})^\alpha \quad (2-11)$$

κ_t 相对 IOU 导数如式(2-12)所示

$$\frac{d\kappa_t}{d\text{IOU}(a_j^{t-1})} = -\alpha \text{IOU}(a_j^{t-1})^{\alpha-1} \quad (2-12)$$

不同过程噪声幂次参数 α 对应 κ 及其导数变化曲线如图 2-2 所示。

由图 2-2(a)可知, IOU 越接近 0, 预测偏差越大, 则 κ 随之增大。当 $\alpha > 1$ 时, κ 的曲线逐渐向右上方偏移; 当 $0 < \alpha < 1$ 时, 情况相反; $\alpha = 1$ 时, 则呈线性变化。由图 2-2(b)可知, 当 IOU 处于高偏差阶段且 $\alpha > 1$ 时, 曲线较之线性变换更加平缓, 接近 1 则变化迅速。利用该特性可抑制高偏差样本, 使其维持较大过程协方差。


 图 2-2 κ 及其导数变化情况

2.2.3 状态更新

所提 PA-U 过程首先权衡观测误差与估计误差，求解卡尔曼增益如式(2-13)所示，

$$\mathbf{K}_t = \mathbf{P}'_t \mathbf{H}^T (\mathbf{H} \mathbf{P}'_t \mathbf{H}^T + \xi_t \mathbf{R})^{-1} \quad (2-13)$$

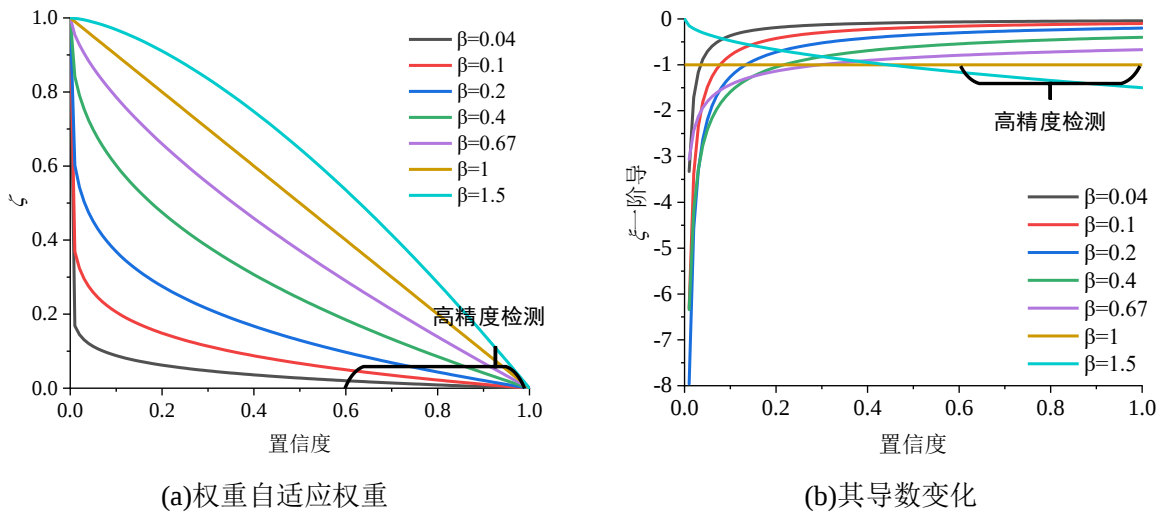
其中， $\xi_t \in (0,1)$ 为所提 PA-U 观测噪声自适应因子，其计算过程如式(2-14)所示，

$$\xi_t = 1 - (s_t)^\beta \quad (2-14)$$

式中， β 为观测噪声幂次参数。 ξ_t 相对 s_t 导数如式(2-15)所示，

$$\frac{d\xi_t}{ds_t} = -\beta (s_t)^{\beta-1} \quad (2-15)$$

不同 β 对应 ξ 及其导数变化曲线如图 2-3 所示。


 图 2-3 ξ 及其导数变化情况

由图 2-3(a)可知， s 与 ξ 负相关；当 $\beta < 1$ 时，曲线向左下方拉伸；当 $0 < \beta < 1$ 时，情

况相反。由图 2-3(b)可知，当 s 较高且 $\beta < 1$ 时，曲线较之线性变换更加平缓，靠近 0 则变化迅速。利用该特性，可使高分段检测样本维持较低观测协方差。

最后，结合 $\mathbf{z}_t$ 、 $\hat{\mathbf{x}}'_t$ 、 $\mathbf{P}'_t$ 与 $\mathbf{K}_t$ ，更新 $\hat{\mathbf{x}}_t$ 与 $\mathbf{P}_t$ ，如式(2-16)所示，

$$\begin{cases} \hat{\mathbf{x}}_t = \hat{\mathbf{x}}'_t + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}'_t) \\ \mathbf{P}_t = (1 - \mathbf{K}_t\mathbf{H})\mathbf{P}'_t \end{cases} \quad (2-16)$$

值得注意的是，当 $\beta=1$ 时，NSAKF 正是所提 PA-U 的特例。

2.3 算法流程

BYTE 算法结合所提 PAKF 后的整体流程如算法 2-1 所示。

算法 2-1 : 结合 PAKF 后的 BYTE 算法流程

输入: 视频序列 V ; 目标检测器 Det ; PAKF 参数 α 和 β ; 数据关联框架 BYTE, 高、低置信度阈值 ζ

输出: 视频对应的轨迹集合 $\mathbb{T}_{\text{finish}}$

```

1:   初始化:  $\mathbb{T} \leftarrow \emptyset$ ;
2:   for frame in  $V$  do
3:        $\mathbb{D}_t \leftarrow \text{Det}(\text{frame})$ 
4:        $\mathbb{D}_{\text{high}}, \mathbb{D}_{\text{low}} \leftarrow \text{Divide}(\mathbb{D}_t, \zeta)$ 
5:        $\mathbb{B}_t \leftarrow \emptyset$  //中间量, 包围框预测状态集合//
6:       for  $\tau_{t-1}$  in  $\mathbb{T}_{\text{activate}}$  do
7:            $(\hat{\mathbf{x}}'_t, \mathbf{P}'_t) \leftarrow \text{PA-P}$ 
8:            $\mathbb{B}_t \leftarrow (\hat{\mathbf{x}}'_t, \mathbf{P}'_t)$ 
9:       end
10:       $(A_t^*, \psi_t^{\text{high}}, \psi_t^{\text{low}}) \leftarrow \text{BYTE}(\mathbb{D}_{\text{high}}, \mathbb{D}_{\text{low}}, \mathbb{B}_t)$ 
11:      for  $a_t$  in  $A_t^*$  do
12:          get IOU( $a_t$ ) from  $(\psi_t^{\text{high}}, \psi_t^{\text{low}})$ 
13:          get  $(\mathbf{L}_i^t, s_i^t)$  from  $D_i^t$ 
14:           $Z_j^t \leftarrow (\mathbf{L}_i^t, s_i^t, \text{IOU}(a_t))$ 
15:           $\mathbb{Z}_t \leftarrow Z_j^t$ 
16:      end
17:      for  $a_t$  in  $(\mathbb{Z}_t, \mathbb{B}_t)$  do
18:          由式(2-14)求得  $\xi_j^t$ 
19:          由式(2-13)求得  $\mathbf{K}_j^t$ 
20:          由式(2-15)更新  $(\hat{\mathbf{x}}_j^t, \mathbf{P}_j^t)$ 
21:          由式(2-11)预测  $\kappa_j^{t+1}$ 
22:           $\tau_j^t \leftarrow (\mathbf{L}_j^k, \hat{\mathbf{L}}_j^k, \mathbf{P}_j^k, \kappa_j^{k+1}, l_j^k)$ 
23:           $\mathbb{T}_t \leftarrow \tau_j^t$ 
24:      end
25:      轨迹状态管理
26:   end
27:   返回:  $\mathbb{T}_{\text{finish}}$ 

```

首先，对检测器输出，划分高、低置信度检测集合(3~4 行)，由 PA-P 前向传播预

测状态(5~9 行); 其次, 运用 BYTE, 获取 A_t^* 、 ψ_{high}^t 与 ψ_{low}^t , 并以此构建 Z_t (10~16 行); 再次, 由 PA-U 及式(2-11), 更新轨迹状态及 κ , 并对所有轨迹进行匹配状态管理(17~25 行); 循环遍历视频序列, 最终输出所有目标轨迹集合 $\mathbb{T}_{\text{finish}}$ 。

2.4 实验与分析

2.4.1 数据集与评价指标

数据集: HiEve 是一个大规模多目标跟踪与人类行为感知数据集; 包含 32 个序列, 近 4.9 万帧, 目标轨迹平均寿命在 480 帧以上; 体量与事件复杂度均超过 MOT17 与 MOT20^[65]; 在其测试集上, 评估了 ByteTrack 与 PA-ByteTrack, 以验证所提方法有效性。在 MOT17 验证集上, 优化 PAKF 幂次参数, 并进行消融实验。此外, 划分 MOT17 中的 05、10、11、13 序列作为高遮挡动态场景验证集, 以探究所提方法在高遮挡动态场景下表现。

评价指标: 引用 CLEAR^[66]指标中的 MOTA、MOTP、IDs、Frag、FP 和 FN 等度量, 综合评价算法跟踪性能; 引入 IDF1^[67]衡量跟踪鲁棒性。然而, MOTA 更加侧重检测性能, 故引入 HOTA^[68]以综合评估检测、定位与关联性能(因 HiEve 并不支持 HOTA 指标, 故在其测试集中仍使用 MOTA)。此外, 引入 MT 和 ML 等传统度量^[69], 以反映轨迹错误类型占比。

一个鲁棒、精准、高效的跟踪算法应同时具备较高的 HOTA、MOTA、MOTP、MT、IDF1 与关联速度, 与较低的 IDs、Frag、FN、FP 与 ML。为直观反映性能优劣, 以下数据中通过箭头表示度量最优变化方向。

2.4.2 实验设置

选取 ByteTrack^[27]、SORT^[13]及 DeepSORT^[4]为实验对照组。为保证实验公平性, 所有算法均采用开源的 YOLOX 检测器^[17], 在幂次参数优化、消融实验和复杂场景评估中采用 ByteTrack 提供的 YOLOX MOT17 权重^[27], 其在 MOT17、CrowdHuman^[70]、Cityperson^[71]和 ETHZ^[72]构成的混合训练集上进行训练; 在基准评估中采用 ByteTrack 公开的 YOLO-X MOT20 权重, 其在 Hieve、MOT20 与 CrowdHuman 构成的混合训练集上进行训练。DeepSORT 的 DAE 源自参考文献^[4], 对照组算法的超参数设置与其原

论文中保持一致，所有算法均未使用后处理^[40]。实验设备如下：CPU 型号为 AMD Ryzen7 3700X，基准频率 3.8GHz，内存大小为 32GB，显卡型号为 Nvidia RTX3060，基准频率为 1320MHz，Python 版本为 3.8，Pytorch 版本为 1.10.1。

2.4.3 幂次参数优化

采用正交试验法优化 PAKF 幂次参数。设 α 、 β 分别作为 2 因素，并各划分出 7 水平，设 α 水平分别为 0.04、0.1、0.2、0.4、0.67、1 和 1.5， β 水平分别为 0.67、1、1.5、2.5、5、10 和 25。首先，搜索单独使用 PA-P 与 PA-U 方案下的最优幂次参数，随后研究二者耦合情况，即完整版 PAKF。

(1) 独立 PA-P 与 PA-U 参数优化：选取 HOTA、MOTP 与 IDF1 以综合评估不同参数水平下性能，单 PA-P 与单 PA-U 性能变化分别如图 3(a)、(b)所示，其中绿色虚线为基线，表示原版 ByteTrack 性能；红色虚线为 NSAKF 方案的性能。

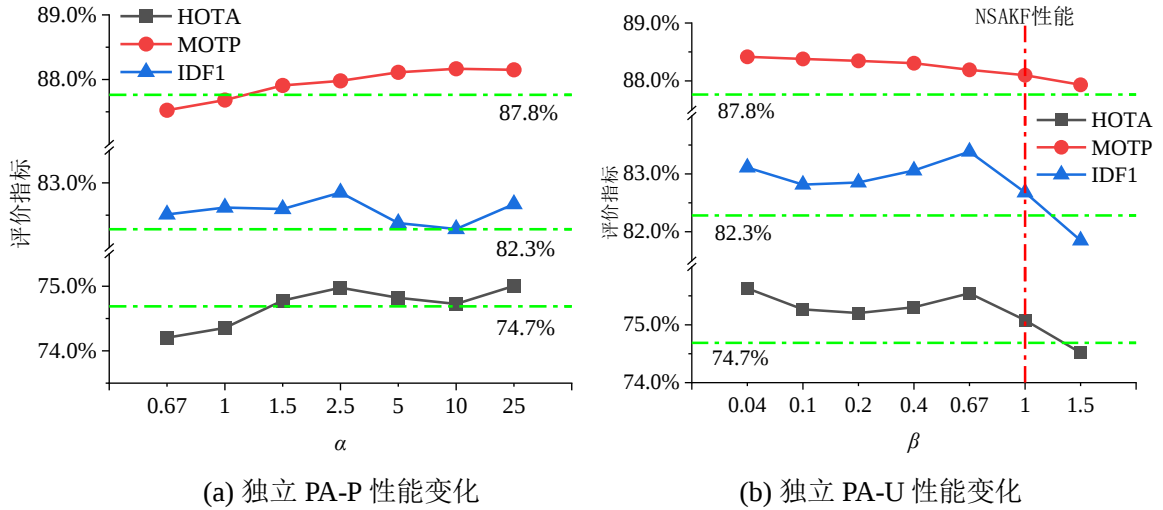


图 2-4 单 PA-P 与单 PA-U 性能变化

由图 2-4(a)可知，所提 PA-P 的 IDF1 始终高于基线；HOTA、MOTP 变化大体上与 α 呈正相关，且在 $\alpha > 1.5$ 后才超越基线，说明抑制高预测偏差样本是必要的。由图 2-4(b)可知：所提 PA-U 的 MOTP 与 β 负相关；IDF1、HOTA 变化高度一致，在 0.04 到 0.67 之间先降后升，且均超过 NSAKF 方案；在 0.67 到 1.5 之间快速下降，且在 $\beta=1.5$ 时低于基线，表明重点调节高置信度样本观测协方差的策略是正确的。单 PA-P 与单 PA-U 方案的 HOTA 分别在 $\alpha=2.5$ 、 $\beta=0.04$ 处取得最优结果(75.0%，75.6%)，故以此分别作为二者的最优幂次参数。

(2) PAKF 参数优化：选取 HOTA、MOTA、IDF1 与 IDs 度量，评估 PAKF 在不同因素水平下性能变化，如图 2-5 所示。

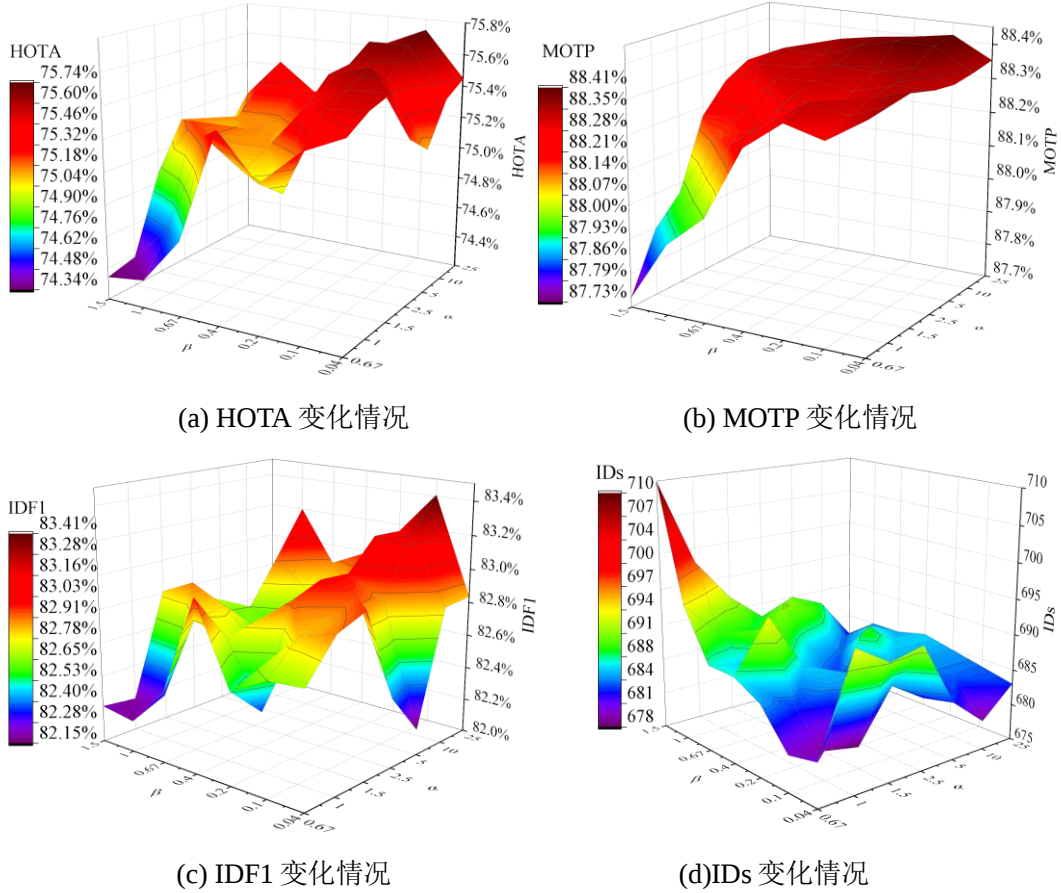


图 2-5 不同因素水平下 PA-ByteTrack 性能变化情况

由图 2-5 可见，随着 α 逐渐增大， β 逐渐减小，PA-ByteTrack 的各项性能逐步提升。HOTA、MOTP 与 IDF1 均在 $(\alpha=25, \beta=0.1)$ 处取得最优值，分别为 75.7%、88.4% 和 83.4%；IDs 在 $(\alpha=1.5, \beta=0.1)$ 处取得最优值 678，在 $(\alpha=25, \beta=0.1)$ 处则为 684，二者差距较小；故选取 $(\alpha=25, \beta=0.1)$ 为 PAKF 最优参数组合。

2.4.4 消融实验

综合以上最优参数设置，以 ByteTrack、SORT 与 DeepSORT 为对照组，在 MOT17 全验证集上，进行消融实验，结果如表 2-1 所示。

由表 2-1 中方案 1、2 和 3 可知，ByteTrack 的 MOTP 是对照组中最差的，仅为 87.8%，而 HOTA 和 IDF1 却为三者中最优。这表明低分检测确实降低了 ByteTrack 的跟踪精度。在方案 8 中，将 NSAKF 引入 ByteTrack，各项度量均较方案 3 提升 0.3% 至

0.4%。在方案 10 中仅引入 PA-U，相对方案 8，HOTA、MOTP 和 IDF1 分别提升 0.6%、0.3%和 0.4%，并取得最优 MT、ML 和 IDs，且几乎未降低关联速度。如方案 11 所示，引入完整的 PAKF，可取得最优 HOTA 和 IDF1，分别为 75.7%和 83.4%，但关联速度由 274.7FPS 下降至 213.4FPS，鉴于实时应用需求(>15FPS)，该损失仍可接受。在方案 9 中，引入单 PA-P，较之原版 ByteTrack，虽小幅提升跟踪性能(+0.3% HOTA，+0.4%MOTP 和+0.6%IDF1)，但关联速度为 219.4FPS。综上，方案 9、10 和 11 均能提升 ByteTrack 性能，其中引入 PAKF 效果最优，单 PA-U 次之，单 PA-P 最差。

表 2-1 消融实验(最优数据用粗体标出)

方案	方法	HOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	Frag↓	IDs↓	speed/FPS↑
1	SORT ^[13]	73.9	87.9	81.7	73.8	3.5	1391	765	277.6
2	DeepSORT ^[4]	73.9	87.9	80.8	73.4	5.0	932	834	24.1
3	ByteTrack ^[27]	74.7	87.8	82.3	78.0	3.7	811	713	274.7
4	SORT + NSAKF	74.9	88.5	82.5	75.1	3.9	1338	711	273.1
5	SORT + PA-P	74.0	88.0	81.7	74.5	3.7	1362	746	226.4
6	SORT + PA-U	75.1	88.6	82.0	75.1	3.5	1354	708	272.7
7	SORT + PAKF	75.4	88.7	82.6	74.9	3.5	1348	702	222.2
8	ByteTrack + NSAKF	75.1	88.1	82.7	76.7	4.0	787	701	271.7
9	ByteTrack + PA-P	75.0	88.2	82.9	78.2	3.9	844	696	219.4
10	ByteTrack + PA-U	75.6	88.4	83.1	79.1	3.5	839	683	271.6
11	PA-ByteTrack	75.7	88.4	83.4	78.9	3.5	850	684	213.4

另一方面，以 SORT 为基线的消融实验结果，见表 2-1 中的方案 4、5、6 和 7。与在 ByteTrack 上的表现类似，方案 7 性能最优，方案 6 次之，而方案 5 较之原版 SORT 几乎没有提升。对比方案 3、4 和 7 可知，引入 PAKF 能有效提升 SORT 性能(+1.5%HOTA、+0.8%MOTP 和+1.2%IDF1)，且各项性能均领先引入 NSAKF 的方案 0.2%至 0.5%。

2.4.5 复杂场景评估

在复杂场景验证集上，对比 PAKF 方案与 NSAKF 方案，实验结果如表 2-2 所示。

由表 2-2 可知，就 3 种对照组方法而言，基于外观匹配的 DeepSORT 在 HOTA 度量上超过其他基线至少 0.7%，SORT 的 IDs 及 Frag 最多，ByteTrack 的 MOTP 最低；对于 ByteTrack 而言，采用 PAKF，在 HOTA、IDF1、MT 和 IDs 等度量上均可取得最

优成绩，且在多个度量上大幅领先 NSAKF 方案(+1.6%HOTA, +0.5%MOTP, +1.7%IDF1, +3.5%MT)；对于 SORT 而言，采用 PAKF 后可在 MOTP 上取得最优成绩，领先 PA-ByteTrack1.4%，但在 HOTA 和 IDF1 上分别落后 1.2%和 2.2%；这是由于 SORT 没有利用低分检测，虽然避免了定位精度的下降，但也限制了跟踪准确度和关联鲁棒性的提升。

表 2-2 复杂场景评估结果(最优结果由粗体标出)

方法	HOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	Frag↓	IDs↓
SORT ^[13]	64.5	84.1	75.0	67.2	5.1	774	516
DeepSORT ^[4]	65.5	84.2	76.6	67.5	5.9	507	480
ByteTrack ^[27]	64.8	84.0	75.3	72.8	4.3	494	481
SORT + NSAKF	65.9	85.2	76.1	70.9	4.3	727	467
SORT + PAKF	66.2	86.3	75.9	70.7	3.7	735	458
ByteTrack + NSAKF	65.8	84.4	76.4	70.4	4.8	484	476
PA-ByteTrack	67.4	84.9	78.0	73.9	4.0	504	443

2.4.6 基准评估

在 HiEve 测试集上，评估了 ByteTrack 和所提 PA-ByteTrack(两者使用同一检测器)，并与其他 SOTA 方法们进行对比，结果如表 2-3 所示。

表 2-3 HiEve 测试集上评估结果(最优结果由粗体标出)

方法	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓	Frag↓
DeepSORT ^[4]	27.1	70.5	28.6	8.5	41.5	5894	42668	2220	3122
JDE ^[24]	33.1	72.3	36.0	16.3	44.2	9526	33327	3747	3605
FairMOT ^[14]	35.0	75.6	46.7	16.3	44.2	6523	37750	995	2312
CenterTrack ^[53]	42.4	75.7	38.4	23.9	26.6	5802	29766	4549	2940
NewTracker ^[73]	46.5	76.9	43.2	26.3	30.9	4667	30489	2133	1981
ByteTrack ^[27]	46.9	77.7	56.0	19.0	33.6	4472	32010	544	1516
PA-ByteTrack	47.4	77.9	54.9	19.5	33.0	4477	31576	600	1532

由表 2-4 可知，PA-ByteTrack 在 MOTA、MOTP 度量上取得最优；较其基线 ByteTrack，在 MOTA、MOTP 与 MT 度量上分别提升 0.5%、0.3%、0.5%，并降低 0.6%ML，但 IDF1 降低了 1.1%。这是由于 PAKF 促使更多检测参与关联，然 HiEve 过于复杂的场景，导致了更多的 IDs。即使如此，所提方法的 IDs 仍大幅低于其他 SOTA

方法(较 FairMOT 下降 39.7%，仅相当于 NewTrack 的 28.1%)。

2.4.7 定性分析

对照组算法与所提 PA-ByteTrack 在 MOT17-05 上的跟踪结果如图 2-6 所示，该序列为动态拍摄，且存在高度遮挡与模糊。

由图 2-6 可见，所有算法均成功跟踪了左数第一位女性，但 3 种对照组方法的包围框均存在一定程度的偏移，而由图 2-6(d)可见，PA-ByteTrack 的包围框更接近目标本体，类似的情况也发生在左数第二位男性与右数第一位女性。



(a) SORT



(b) DeepSORT



(c) ByteTrack



(d) PA-ByteTrack

图 2-6 MOT17-05 运行结果(#514、#516、#518)

2.5 本章小结

针对全检测关联 MOT 算法的局限性，提出了一种基于幂次变换的 PAKF 模型，该模型由 PA-P 和 PA-U 组成。以运动匹配成本和置信度分数分别作为过程和观测噪声的尺度参数，并引入线性卡尔曼滤波中。由幂次变换构建非线性自适应因子，对不同轨迹的过程和观测协方差矩阵进行差异化调整。在 MOT17 半验证集上，得益于幂次变换，采用 PA-U 的 ByteTrack 的 HOTA、MOTP 和 IDF1 比采用 NSAKF 版本的要领先 0.3%到 0.6%，且关联速度并未降低。此外，引入 PAKF 后 ByteTrack 的各项性能均获得更显著的提升，且额外计算导致的速度损失是可接受的。因此，建议在性能敏感场景使用完整版 PAKF，在对时间敏感场景使用 PA-U。此外，PAKF 对传统 SORT 方法同样有效，但更侧重于提高跟踪精度。与 ByteTrack 相比，PA-ByteTrack 的 MOTA、MOTP 和 MT 在 HiEve 测试集上分别领先 0.5%、0.3%和 0.5%。且明显优于其他 SOTA 方法。

第 3 章 融合轻量级深度外观特征提取的在线多目标跟踪研究

得益于深度学习检测器的日渐成熟，MOT 算法的准确度已取得了长足的进步。然而，由于卡尔曼滤波的马尔科夫性，基于运动的 MOT 方法难以应对长期的目标遮挡和快速的视角变化。引入额外的外观线索有助于提升 MOT 算法在高遮挡场景下的跟踪鲁棒性。但是，现有的深度外观提取方案难以平衡速度、性能和内存使用，难以应用于低算力的移动端场景。

为缓解 DeepSORT 类方法的 DAE 造成的速度下降与内存占用率提升。本章分析了 Re-ID 与 MOT 任务间的差异，针对性改进轻量级 Re-ID 网络 OSNet^[23]。通过修改其 OSRB 结构，引入通道和空间注意力机制，并添加额外的局部特征输出分支，提出了 Fast OSNet。结合表征^[74]与度量学习^[75]，联合多 ID 损失^[76]和圆损失^[35]为其设计损失函数，在 Market1501 数据集^[77]上开展了训练与评估。此外，为缓解遮挡噪声对轨迹外观特征的污染。基于 EMA，引入轨迹外观置信度，结合二段线性变换，设计自适应外观更新权重，提出了低代价、鲁棒、自适应的轨迹外观状态更新策略，名为 HAEMA。二者组合构成了所提轻量级深度外观提取方案。以 DeepSORT 为基线，引入所提方案与 BYTE^[27]，提出 FDBTrack。为验证了其可行性与有效性。首先，在 Market1501 测试集上进行了 Fast OSNet 网络的消融及对比实验。通过换用其他 Re-ID 网络，并可视化网络的 CAM，揭示了网络结构与跟踪性能间的潜在联系。其次，在 MOT17 半验证集上开展了 FDBTrack 的消融和对比实验。再次，在 MOT17 和 MOT20 测试集上进行了基准评估，并与近期的其他方法进行对比，以验证其有效性。

3.1 Fast OSNet

3.1.1 Fast OSNet 的结构

图 3-1 可视化了所提 Fast OSNet 结构，详细的结构如表 3-1 所示(输入大小为 $128 \times 64 \times 3$)。在基本保留 OSNet 网络架构的前提下，为适应 MOT 任务需求，做出以下 3 点改进：

(1) 简化全尺度残差块。OSNet 的 OSRB 与所提 Fast OSNet 的 SOSRB 如图 3-2 所

示。OSRB 有 3×3 、 5×5 、 7×7 、 9×9 卷积支路和残差支路。多个支路能提升网络性能，但也带来额外的计算开销^[78]。考虑到 MOT 任务中，DAE 的输入分辨率通常只有 128×64 ，小于 Re-ID 中常见的 256×128 ，过大的感受野，作用并不明显，故删除 9×9 支路。此外，OSRB 的感受野是通过叠加多个 DWConv^[79]实现的，这会显著增加内存访问成本，从而大幅降低推理速度。近期的研究^[80]发现直接采用大核卷积，能显著增加网络的实际感受野，从而提升网络性能，因此将 5×5 与 7×7 支路都替换为单个大卷积核的 DWConv。

(2) 引入空间与信道注意力机制。加入注意力机制能增强网络表征能力。OSNet 在每个模块后设置 AG，以动态组合不同支路间的输出特征，这实际上是一种支路间注意力机制。然而，层间的特征传递同样需要得到重视。为此，将 CBAM^[81]拆分为空间注意力与信道注意力模块，在特征图分辨率较高的 Conv1 后添加空间注意力模块，在信道较多的 Conv3 后添加信道注意力模块。

(3) 添加局部特征输出支路。参照 ICNN^[37]结构，利用 PAP 提取 4 个具备空间信息的局部特征输出，通过 GAP 得到全局特征输出。将其降维，得到 4 个局部特征向量 $h_1, h_2, \dots, h_4$ 和 1 个全局特征向量 h_g ，如图 3-1 红虚线框内所示。其中， $h \in \mathbb{R}^{1\times d}$ ， d 为最后一层特征维度。在训练时，每个输出分支均具有独立的 ID 损失，如图 3-1 蓝虚线框所示。此外，为进行度量学习，需拼接所有特征向量，并投影到单位超球面，形成外观嵌入向量 f ，如式(3-1)所示。网络在推理时，只输出 f 。

$$f = [h_1, h_2, \dots, h_4, h_g]_{1\times(d\times 5)} \quad (3-1)$$

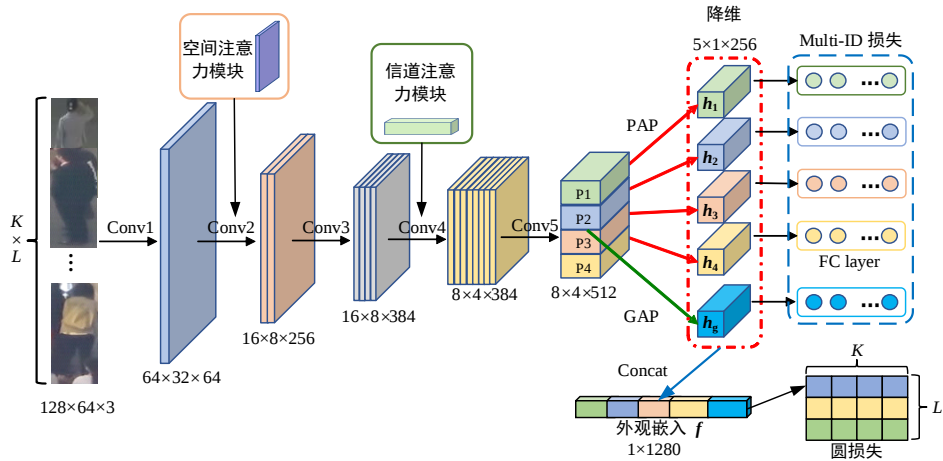


图 3-1 所提 Fast OSNet 网络结构图

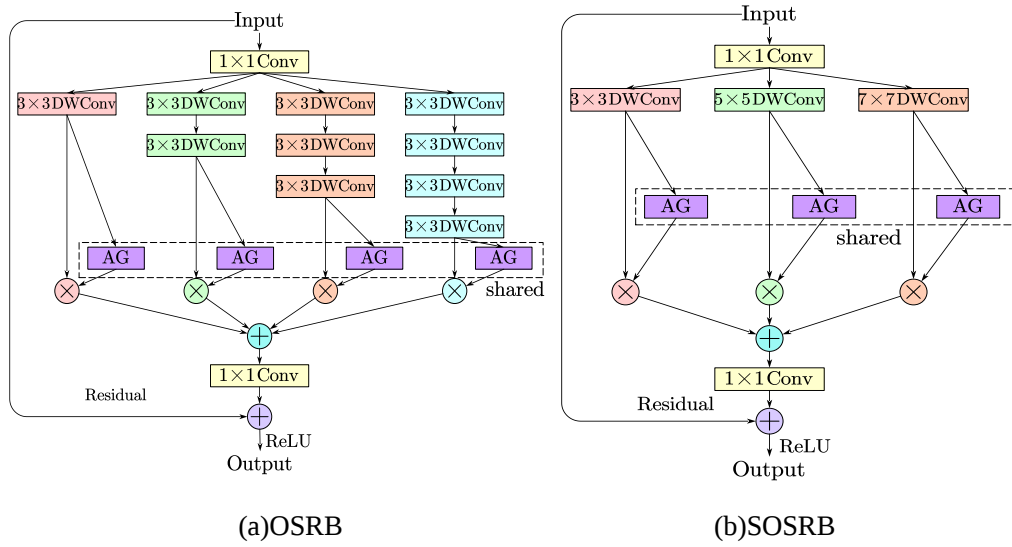


图 3-2 OSRB 与 SOSRB 结构对比

表 3-1 所提 Fast OSNet 结构

网络层级	输出大小	Fast OSNet	
Conv1	64×32×64	7×7 卷积，步幅为 2	
最大池化层	32×16×64	3×3 最大池化，步幅为 2	
空间注意力层	32×16×64	7×7 卷积，步幅为 1	
Conv2	32×16×256	SOSRB×2	
通道融合	16×8×256	1×1 卷积，步幅为 1；2×2 平均池化，步幅为 2	
Conv3	16×8×384	SOSRB×2	
通道融合	8×4×384	1×1 卷积，步幅为 1；2×2 平均池化，步幅为 2	
通道注意力层	8×4×384	压缩率为 0.5	
Conv4	8×4×512	SOSRB×2	
Conv5	8×4×512	1×1 卷积，步幅为 1	
全局与部件信息融合层	GAP×1	1×1×512	全局级平均池化
	PAP×4	4×1×512	部件级平均池化
维度压缩	5×1×256	1×1 卷积，步幅为 1	

3.1.2 联合损失函数

结合 ID 损失与圆损失，提出联合损失函数。所有输出分支后的 ID 损失训练模块，由全连接层与 SoftMax 函数组成，如图 3-1 中蓝色虚线框所示，输出维度为数据集 ID 数。用于表征学习的多分支 ID 损失如式(3-2)所示，

$$l_{\text{multi-ID}} = l_{\text{ID}}^p + l_{\text{ID}}^g \quad (3-2)$$

其中, l_{ID}^p 为局部特征 ID 损失, 如式(3-3)所示,

$$l_{ID}^p = -\frac{1}{N_c} \sum_{p=1}^n \sum_{i=1}^{N_c} \log \left(\frac{\exp(\mathbf{w}_p^{y_i}(\mathbf{h}_p^i)^T)}{\exp(\mathbf{w}_p^{y_i}(\mathbf{h}_p^i)^T) + \sum_{j=1}^{N_c-1} \exp(\mathbf{w}_p^j(\mathbf{h}_p^j))} \right) \quad (3-3)$$

式中, N_c 为数据集 ID 数;

p 为局部分支编号;

$\mathbf{W}_p$ 为第 p 个部件分支的分类权重矩阵;

$\mathbf{W}_p^{y_i}$ 与 $\mathbf{W}_p^j$ 分别为 $\mathbf{W}_p$ 的第 y_i 行与第 j 行, y_i 为正样本编号。

全局特征 ID 损失 l_{ID}^g , 如式(3-4)所示,

$$l_{ID}^g = -\frac{1}{N_c} \sum_{i=1}^{N_c} \log \left(\frac{\exp(\mathbf{w}_g^{y_i}(\mathbf{h}_g^i)^T)}{\exp(\mathbf{w}_g^{y_i}(\mathbf{h}_g^i)^T) + \sum_{j=1}^{N_c-1} \exp(\mathbf{w}_g^j(\mathbf{h}_g^j))} \right) \quad (3-4)$$

式中, $\mathbf{W}_g$ 为全局分支的分类权重矩阵。

另外, 为进行度量学习, 需构建一个 $L \times K$ 的小批量训练样本, 包含 L 个类别, 每个类别有 K 个样本, 如图 2 所示。对锚定外观特征 $\mathbf{f}_a$ 而言, 与正、负样本外观特征 $\mathbf{f}_p^i$ 、

$\mathbf{f}_n^j$ 的余弦距离 $d_{a,p}^i$ 、 $d_{a,n}^j$, 如式(3-5)所示,

$$\begin{cases} d_{a,p}^i = 1 - \mathbf{f}_a(\mathbf{f}_p^i)^T, & i \in 1, 2, \dots, K-1 \\ d_{a,n}^j = 1 - \mathbf{f}_a(\mathbf{f}_n^j)^T, & j \in 1, 2, \dots, (L-1)K \end{cases} \quad (3-5)$$

用于度量学习的圆损失如式(3-6)所示,

$$l_{\text{circle}} = -\log \left[1 + \overbrace{\sum_{j=1}^{K(L-1)} \exp(\rho[1 - d_{a,n}^j + m]_+(1 - d_{a,n}^j - m))}^{\text{负样本}} \times \overbrace{\sum_{i=1}^{K-1} \exp(-r[d_{a,p}^i + m]_+(m - d_{a,p}^i))}^{\text{正样本}} \right] \quad (3-6)$$

式中, ρ 为比例因子;

m 为间隔边界, 区间为(0,1), 用于促进相似性分离;

$[\cdot]_+$ 为铰链函数。

综合多 ID 损失与圆损失, 联合损失如式(3-7)所示,

$$l_{\text{total}} = l_{\text{multi-ID}} + w_c l_{\text{circle}} \quad (3-7)$$

式中, $w_c \in (0,1)$ 为圆损失调节权重。

3.2 HAEMA

EMA 固定的更新权重无法适应动态的跟踪过程，从而引发图像遮挡噪声对轨迹外观特征的污染。参考文献^[15]已证明检测器置信度与目标被遮挡程度呈正相关。因此，在 EMA 基础上，引入轨迹外观分数与多段线性变换，提出 HAEMA，其运作流程如下。

首先，数据关联算法得出第 t 帧匹配，根据第 t 帧的目标置信度 s_j^t 更新轨迹外观分数 c_i^{t-1} 。对二者求差，再利用二段线性变换，得出轨迹 i 在 t 时刻的外观更新权重 a_i^t ，如式(3-8)所示，

$$a_i^t = \begin{cases} (1 - \eta)(s_j^t - c_i^{t-1}) + \eta, & s_j^t \geq c_i^{t-1} \\ \eta(s_j^t - c_i^{t-1}) + \eta, & s_j^t < c_i^{t-1} \end{cases} \quad (3-8)$$

式中， $\eta \in [0, 0.5]$ 为自适应梯度，通过调节 η ，使得 a_i^t 在 2 种情况下具备不同性质。具体而言，当 $s_j^t \geq c_i^{t-1}$ 时，适当增加 a_i^t ，吸纳低遮挡噪声的目标外观特征；反之，则应降低 a_i^t ，抑制高遮挡噪声特征，进而提升轨迹关联正确率。此外，由于该自适应函数仅涉及线性变换，故不会过度加重计算开销。

其次，根据 a_i^t 与 f_j^t 更新轨迹 i 的外观状态，如式(3-9)所示，

$$e_i^t = \begin{cases} f_j^t, & t = t_s \\ (1 - a_i^t)e_i^{t-1} + a_i^t f_j^t, & otherwise \end{cases} \quad (3-9)$$

式中， e_i^{t-1} 为上一时刻轨迹 i 的外观状态；

t_s 为轨迹初始帧。

最后，对轨迹外观分数执行 EMA 更新，如式(3-10)所示，

$$c_i^t = \begin{cases} c_j^t, & t = t_s \\ qc_i^{t-1} + (1 - q)c_j^t, & otherwise \end{cases} \quad (3-10)$$

式中， q 为分数融合权重。

事实上，所提 HAEMA 相当于分层进行了两次 EMA，第一次的自适应 EMA 更新轨迹外观特征，第二次则更新轨迹外观分数。因此，所提 HAEMA 可兼容所有使用传统 EMA 更新的方法。

3.3 FDBTrack

所提 FDBTrackd 如算法 3-1 所示，其输入为视频序列 V ，目标检测器 Det ，Fast OSNet 和高、低分阈值 ς ；输出视频中所有感兴趣目标的轨迹，即每个目标在每帧的包围框与 ID 号。

在每个视频序列开始，初始化空轨迹集合 $\mathbb{T}_{\text{start}}$ 。对视频中的第 t 帧，首先，通过 Det 得出所有目标的包围框与置信度分数，依据 ς 将检测集合划分为高、低分检测框集合 $\mathbb{D}_{\text{high}}$ 、 $\mathbb{D}_{\text{low}}$ ，并通过线性 KF 预测上一帧轨迹在第 t 帧的位置(3~5 行)；其次，依据 $\mathbb{D}_{\text{high}}$ 裁剪原图，得出对应的图像块集合，并将它们的大小对齐为 128×64 ，利用所提 Fast OSNet，获得所有图像块的 1280 维特征向量 $\mathbf{f}$ ，计算 $\mathbb{D}_{\text{high}}$ 与 $\mathbb{T}_{t-1}$ 间的外观距离与 IOU 距离，构建融合相似度矩阵，如式(3-11)所示，利用 HA 进行第一次关联(6~9 行)；再次，在未关联轨迹 $\mathbb{T}_{\text{unmatch}}$ 与 $\mathbb{D}_{\text{low}}$ 之间执行第二次关联，由于低分检测框通常处于高遮挡与运动模糊情况下^[27]，外观极度失真，故只构建 IOU 相似度矩阵，如式(3-12)所示(10~13 行)；最后，对轨迹进行管理，包括删除长时间未匹配的旧轨迹，生成新轨迹，通过 KF 更新轨迹在 t 时刻的包围框，利用所提 HAEMA 更新第一次关联的轨迹外观特征(14~19 行)。如此，循环遍历视频序列，最终获得所有轨迹集合 $\mathbb{T}_{\text{finish}}$ 。

$$\mathbf{C}_{\text{high}} = \lambda \mathbf{C}_{\text{app}} + (1 - \lambda) \mathbf{C}_{\text{IOU}} \quad (3-11)$$

$$\mathbf{C}_{\text{low}} = \mathbf{C}_{\text{IOU}} \quad (3-12)$$

其中 λ 为相似度矩阵融合权重， $\mathbf{C}_{\text{app}}$ 、 $\mathbf{C}_{\text{IOU}}$ 分别为外观、IOU 相似度矩阵，如式(3-13)和式(3-14)所示。

$$\begin{cases} d_{i,j}^{\text{app}} = 1 - e_i f_j^T \\ g_{i,j}^{\text{app}} = \begin{cases} 1 & d_{i,j}^{\text{app}} \leq \varepsilon_{\text{app}} \\ +\infty & \text{otherwise} \end{cases} \\ \mathbf{C}_{\text{app}} = [d_{i,j}^{\text{app}} \times g_{i,j}^{\text{app}}]_{i \times j} \end{cases} \quad (3-13)$$

$$\begin{cases} d_{i,j}^{\text{IOU}} = 1 - \text{IOU}(\mathbf{L}_i^k, \mathbf{L}_j^k) \\ g_{i,j}^{\text{IOU}} = \begin{cases} 1 & d_{i,j}^{\text{IOU}} \leq \varepsilon_{\text{IOU}} \\ +\infty & \text{otherwise} \end{cases} \\ \mathbf{C}_{\text{IOU}} = [d_{i,j}^{\text{IOU}} \times g_{i,j}^{\text{IOU}}]_{i \times j} \end{cases} \quad (3-14)$$

式中， $d_{i,j}^{\text{app}}$ 和 $d_{i,j}^{\text{IOU}}$ 分别为外观距离与 IOU 距离；

ε_{app} 和 ε_{IOU} 分别为外观与 IOU 门限阈值, 用于过滤不可能的关联; 若检测 i 与轨迹 j 间的关联是可接受的, 则 $g_{i,j}^{app}$ 或 $g_{i,j}^{IOU}$ 为 1, 反之则为正无穷。

算法 3-1: FDBTrack 算法流程

输入: 视频序列 V ; 目标检测器 Det ; Fast OSNet; ς

输出: 视频多目标轨迹 T_{finish}

```

1:   初始化:  $T_{start} \leftarrow \emptyset$ ;
2:   for frame in  $V$  do
3:        $\mathbb{D}_t \leftarrow Det(frame)$ 
4:        $\mathbb{D}_{high}, \mathbb{D}_{low} \leftarrow Divide(\mathbb{D}_t, \varsigma)$ 
5:       利用 KF 预测  $L_i^t$ 
6:        $f_{high} \text{ set} \leftarrow Fast\ OSNet(\mathbb{D}_{high}, I_t)$ 
7:       由式(3-11)计算  $C_{high}$ 
8:       首次关联  $\mathbb{D}_{high}$  与  $T_{t-1}$ 
9:       获得  $\mathbb{D}_{unmatch}, \mathbb{D}_{first-match}, T_{unmatch}$  和  $T_{first-match}$ 
10:       $\mathbb{D}_{low} \leftarrow \mathbb{D}_{low} \cup \mathbb{D}_{unmatch}$ 
11:      由式(3-12)计算  $C_{low}$ 
12:      二次关联  $\mathbb{D}_{low}$  与  $T_{unmatch}$ 
13:      获得  $\mathbb{D}_{unmatch}, \mathbb{D}_{second-match}, T_{unmatch}$  与  $T_{second-match}$ 
14:       $e_{first-match} \text{ set} \leftarrow HAEMA(T_{first-match}, f_{high})$ 
15:       $T_{match} \leftarrow T_{first-match} \cup T_{second-match}$ 
16:      利用 KF 更新  $L_i^t$ 
17:      从  $T_{unmatch}$  中删除  $T_{delete}$ 
18:      由  $\mathbb{D}_{unmatch}$  创建  $T_{new}$ 
19:       $T_t \leftarrow T_{match} \cup T_{unmatch} \cup T_{new}$ 
20:   end
21:   返回:  $T_{finish}$ 

```

3.4 实验与分析

3.4.1 数据集与评价指标

数据集: 所提 Fast OSNet 和实验涉及的其他 Re-ID 网络均在 Market1501 数据集上进行训练与评估, 该数据集包含 1501 个 ID 的 3 万多张图片, 其中训练集包含 751 个 ID 的 12936 张图像, 查询子集包含 750 个 ID 的 3368 张图像, 测试集包含 750 个 ID 的 15013 张图像。而对于 MOT 任务, 所有方法均在 MOT17 与 MOT20 数据集上进行实验。MOT17 数据集包含 14 段视频, 训练集与测试集各占一半, 包含低分辨率、低光照、低帧率和相机运动等多种复杂拍摄场景, 整个数据集包含 1.1 万帧, 1342 条行人轨迹, 平均每帧有 19.15 个目标。而 MOT20 数据集包含 1.3 万帧, 3457 条行人轨迹, 平均每帧有 121.28 个目标, 较 MOT17 更加拥挤。在消融实验中, 采用 MOT17 训练集

的后半段作为验证集；在基准评估中，则分别在 MOT17 与 MOT20 测试集上评估。

评价指标：采用 Re-ID 任务中常见的 Rank-1 准确度与 mAP 作为 Re-ID 网络的评价指标。前者反映首位检索结果的准确率，而后者为所有检索的平均准确率。此外，为综合评价网络部署性能，引入网络参数 $\text{params}(\text{M})$ 、内存占用 $\text{memory}(\text{MB})$ 、FLOPs 与每秒推理批次数(批次大小为 100)。对于 MOT 任务，所采用的评价指标与第 2.4 节一致。

3.4.2 实验设置

为保证实验公平性，所有的 Re-ID 网络均未使用预训练权重。它们均使用相同的训练方案，即批大小为 64，其中 K 为 16，L 为 4，训练迭代 120 轮，优化器为 Adam，初始学习率为 3.5×10^{-4} ，在第 75 和 100 轮，衰减为 7×10^{-5} 和 1.4×10^{-5} 。为避免网络过拟合，运用了一些数据增强技术，如随机水平翻转、切割、颜色抖动与擦除。对于联合损失函数，ID 数、 ρ 、 m 和 w_c 分别设置为 751、128、0.3 和 0.001。对于 MOT 任务而言，消融实验中所有方法均相同的 YOLOX-X 检测器及对应的 MOT17 半训练集权重，HAEMA 中的 η 和 q 为 0.05 与 0.9， ζ 、 ε_{app} 和 ε_{IOU} 为 0.6、0.5 和 0.4。实验设备如下：CPU 型号为 AMD Ryzen7 3700X，基准频率 3.8GHz，内存大小为 32GB，显卡型号为 Nvidia RTX3060，基准频率为 1320MHz，所有程序均通过 Python 3.8 和 PyTorch 1.10 实现。

3.4.3 Re-ID 网络消融及对比实验

选取 DeepSORT 中使用的简单 CNN 模型(以下简称 DeepSORT CNN)、参考文献^[29]所使用的 ResNest50 和轻量级网络 ShuffleNetV2^[78]作为对照组。所提 Fast OSNet 在 Market1501 数据集上消融及对比实验结果如表 3-2 所示。

由表 3-2 可知：

(1) 由于大量使用 DWConv 和多分支的结构设计，OSNet 的推理速度是所有方法中最慢的，将 OSRB 替换为所提 SOSRB 后，其推理速度增加了 40.5 Batches/Sec.，内存占用减少了 19.8%，此外，实际感受野的增大，使得 Rank-1 与 mAP 分别提升 0.5 和 0.7。但是，采用大核卷积也导致 params 与 FLOPs 分别提升了 13.6%与 29.3%。

(2) 在添加通道和空间注意力模块后，Rank-1 和 mAP 分别提升了 0.3 与 0.2，且内

存占用和 FLOPs 几乎没有增加。params 增加了 0.2M，推理速度略有降低。相较于结果而言，这些代价是可接受的。

表 3-2 所提 Fast OSNet 消融及对比实验结果(最优数据用粗体标出)

网络	输入	Rank-1↑	mAP↑	params↓	memory↓	FLOPs↓	Batches/Sec.↑
OSNet(基线)	128×64	85.9	68.9	2.2	24.7	250.6	53.8
+ SOSRB	128×64	86.4	70.6	2.5	19.8	324.0	94.3
+SOSRB & attention	128×64	86.7	70.8	2.7	19.8	324.7	91.7
Fast OSNet	128×64	87.3	72.6	2.8	19.9	324.8	90.9
Fast OSNet	256×128	89.0	75.3	2.8	79.3	1300	73.5
Fast OSNet	64×32	73.2	54.5	2.8	5.0	81.8	99.0
DeepSORT CNN	128×64	75.4	51.8	11.5	16.8	1130	149.3
ResNest50	128×64	89.4	76.7	25.4	23.0	884.5	59.9
shuffleNetV2	128×64	77.6	56.8	5.3	6.5	96.6	129.9

(3) 通过添加局部与全局特征融合机制，得到了所提 Fast OSNet，其 Rank-1 和 mAP 再次提升了 0.6 和 1.8。虽然 params、内存占用和 FLOPs 小幅增加，但较 OSNet 而言，其 Rank-1 和 mAP 分别提升 1.4 和 3.7，特别是推理速度增加了 69.0%，内存占用减少了 19.4%。

(4) 提高输入分辨率至 256×128 ，使得所提方法的 Rank-1 和 mAP 分别提升了 1.7 和 2.7，在 memory 和 FLOPs 方面增加了近 4 倍，并导致推理速度下降 17.4 Batches/Sec.。因此，对于 Re-ID 任务而言，提高输入分辨率使模型能捕捉到更多细节信息，从而提高输出特征的可靠性，但也会带来更高的计算代价与内存占用。另一方面，降低输入分辨率至 64×32 ，可以将 memory 和 FLOPs 缩小近 4 倍，推理速度提升 8.1 Batches/Sec.，但 Rank-1 和 mAP 分别减低了 14.1 和 18.1。因此，在选择网络输入分辨率时需要根据具体任务需求进行权衡考虑。

(5) ResNest50 的 Rank-1 和 mAP 是所有方法中最优的。然而，其 params 和 FLOPs 分别为所提方法的 22.6 倍和 2.7 倍，且推理速度落后 31.0 Batches/Sec.。DeepSORT 中提出的 DeepSORT CNN 具备最低的 Rank-1 和 mAP，和最高的 FLOPs。反直觉的是，其推理速度是最快的。这是由于其未使用 DWConv 和多支路设计，以及 PyTorch 框架

对传统卷积的特殊优化。事实上，参考文献^[80]的工作已经证明，若框架对 DWConv 做出特殊优化，二者间的速度差异是可弥补的。此外，ShuffleNetV2 的内存占用最小，FLOPs 为所提方法的 29.7%，推理速度为 129.9 Batches/Sec.，然其 Rank-1 和 mAP 为 77.6 和 56.8，均大幅低于所提方法。

综上，所提方法不仅提升了 OSNet 的 Re-ID 性能，也改善了其运行速度和内存占用。与其他方法们相比，所提方法的 Re-ID 性能、推理速度和内存占用均不是最优的，但各方面性能较为均衡。

3.4.4 MOT 消融实验

以 DeepSORT 为基线，在 MOT17 半验证集上开展了消融实验，以量化评估 3 个改进点的贡献。表 3-3 报告了从基线到所提 FDBTrack 的演化路径。

表 3-3 所提 FDBTrack 消融实验结果(最优数据用粗体标出)

方法	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
DeepSORT ^[4]	62.6	73.9	72.3	61.7	65.8	5.9	407	27.3
+ Fast OSNet	63.6	74.1	74.1	63.4	62.9	7.9	422	31.0
+ BYET	67.9	77.8	79.7	70.1	60.7	11.5	360	44.4
FDBTrack	68.0	78.2	80.0	69.9	60.5	10.6	261	43.6

由表 3-3 可知，所提 Fast OSNet 将 DeepSORT 的 HOTA、MOTA、IDF1 和 AssA 分别提升了 1.0%、0.2%、1.8%和 1.7%，且关联速度提升了 3.7 FPS。这表明较之 DeepSORT CNN，所提网络能够提供更具有鉴别度的外观嵌入。在引入 BYTE 后，HOTA、MOTA、IDF1 和 AssA 分别提升了 4.3%、3.7%、5.6%和 6.7%，关联速度提升了 13.4FPS，这表明全检测关联方法不仅能有效提高跟踪性能，还能加速关联过程。在引入所提 HAEMA 后，HOTA、MOTA 和 IDF1 分别提升了 0.1%、0.4%和 0.3%，IDs 降低了 27.5%，且关联速度仅降低了 0.8FPS，是 DeepSORT 的 1.6 倍。因此，HAEMA 能提高跟踪准确度和鲁棒性，且额外的计算代价是可接受的。综上，Fast OSNet 和 HAEMA 都能有效的提升跟踪性能，并且它们的组合同样有效。

3.4.5 网络替换实验

为进一步探究不同 Re-ID 网络对跟踪性能的影响，选择第 3.4.3 节中涉及的其他网络作为对照组。表 3-4 报告了对照组网络与所提网络在 MOT17 半验证集上的评估结果。

表 3-4 各 Re-ID 模型在 MOT17 验证集上的跟踪结果(最优数据用粗体标出)

网络	输入	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
DeepSORT CNN ^[4]	128×64	67.9	77.8	79.6	70.0	60.7	11.5	363	64.8
ResNest50 ^[29]	128×64	68.0	77.9	79.7	69.9	60.5	11.1	323	34.9
shuffleNetV2 ^[78]	128×64	67.9	77.9	79.6	69.8	60.6	11.3	282	51.1
OSNet ^[23]	128×64	67.9	77.8	79.6	69.8	60.5	11.3	279	29.5
Fast OSNet	256×128	68.1	78.1	80.0	69.8	60.5	10.6	265	32.0
Fast OSNet	64×32	67.5	77.8	79.4	69.6	61.7	11.5	349	46.8
Fast OSNet	128×64	68.0	78.2	80.0	69.9	60.5	10.6	261	43.6

如表 3-4 所示, 在输入大小为 128×64 的情况下, 采用所提网络的方案表现最佳, 其 HOTA、MOTA 和 IDF1 分别比次优方案(ResNest50) 高出了 0.1%、0.3%和 0.3%。同时, 该方案在关联速度上也表现优异, 是采用 OSNet 方案的 1.6 倍。相比于对照组模型, 所提出的轻量级网络在需要同时兼顾关联速度和跟踪性能的场景中更具适用性。

由第 3.4.3 节实验可知, 提升输入分辨率能够有效提升网络的 Re-ID 性能。然而, 对于采用所提出的网络方案而言, 进一步提升分辨率并不能提升跟踪性能, 反而会导致 26.6%的关联速度下降。这是因为 MOT 数据集中目标的图像普遍较小, 过大的输入分辨率并不适用。另一方面, 降低输入分辨率也会导致 HOTA、MOTA、IDF1 和 AssA 分别下降 0.5%、0.4%、0.6%和 0.3%。因为过低的分辨率会损失大量外观信息, 进而降低跟踪性能。

值得注意的是, 尽管对照组网络间的 Re-ID 性能差别较大, 但它们的跟踪性能却极其接近, 这与它们在表 3-2 中的表现并不相符。为了进一步探究造成这一现象的原因, 可视化所有 Re-ID 网络在 128×64 输入分辨率下的 CAM^[82], 具体结果见图 3-3。由左至右, 分别为原图、DeepSORT CNN、ResNest50、ShuffleNetV2、OSNet 和 Fast OSNet 的 CAM。分析图 3-3 可知:

(1) ResNest50 的关注区域明显大于其他网络, 这得益其优异的结构设计和更大的模型规格。然而, 观察图 3-3(a)和(b)可发现, 过大的关注区域也导致网络关注了部分背景, 从而使得目标外观特征中夹杂了背景噪声, 这对于背景相似的 MOT 任务是不利的。而在 Re-ID 任务中, 同一 ID 的图像通常来自不同机位的不同设备, 背景间通常

并不具备相似性。这引起了 ResNest50 在两个任务间的差异。

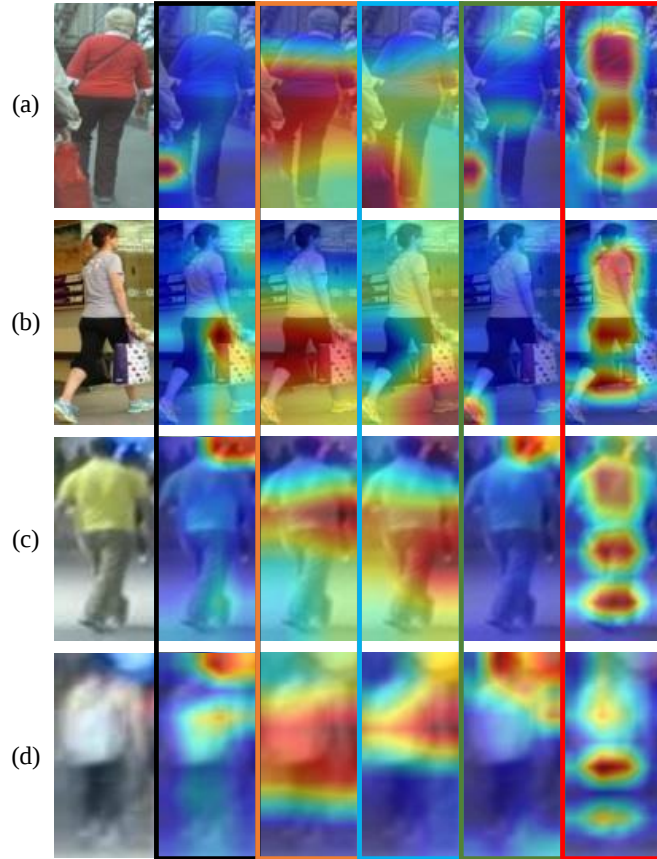


图 3-3 所涉及 Re-ID 网络的 CAM 的可视化

(2) 对比 DeepSORT CNN 与 OSNet 的 CAM 可知，二者的关注区域十分相似，这导致了二者在 MOT 任务上相近的性能。此外，尽管面积有所不同，但所有对照组网络均只关注目标的某一部分。这是由于它们仅考虑了全局特征，并没有引入空间信息。这导致输出特征极易受到遮挡噪声的干扰，从而关注错误的区域。如图 3-3(a)中 DeepSORT CNN 和 OSNet 的 CAM 所示。

(3) 所提 Fast OSNet 存在多个小的关注区域，它们相互叠加基本覆盖了目标全身。这得益于 4 个局部分支带来的输出冗余。这些额外的局部特征引入了空间信息，丰富了网络的输出多样性，变相扩大了网络的关注区域。这意味着，当一个分支关注了错误的区域时，还有其他分支能输出正确的信息。这提高了网络应对遮挡噪声的能力。而在 MOT17 和 MOT20 数据集中，存在大量复杂拥挤的街道场景。因此，相较于对照组网络而言，融合了局部与全局特征的所提网络更适用于高遮挡环境下的 MOT 任务。

3.4.6 对比实验

为公平的比较所提方法与其他 SOTA 在线方法的跟踪性能与关联速度, 表 3-5 报告了 OC-SORT、ByteTrack、Deep OC-SORT、BOT-SORT 与所提方法在同一设备上对 MOT17 验证集的跟踪结果。值得注意的是, 其中所有方法均使用相同的检测器, 且未使用任何后处理技术。表中 App 表示方法是否使用了外观线索, CMC 代表是否使用了相机运动补偿。

表 3-5 所提方法与对比方法在 MOT17 验证集上结果(最优数据用粗体标出)

方法	App	CMC	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
OC-SORT ^[28]			66.1	74.5	77.6	68.6	48.7	16.5	233	219.0
ByteTrack ^[27]			67.7	77.7	79.3	69.5	58.1	12.7	213	265.7
Deep OC-SORT ^[30]	√	√	68.3	75.1	81.0	72.4	50.2	15.3	138	6.6
BOT-SORT ^[11]	√	√	69.1	78.5	81.9	74.4	60.5	11.2	162	8.9
FDBTrack	√		68.0	78.2	80.0	69.9	60.5	10.6	261	43.6

如表 3-5 所示, 虽然 FDBTrack 的速度不及仅使用运动线索的 OC-SORT 和 ByteTrack, 但引入外观线索的提升依然显著。与 OC-SORT 相比, FDBTrack 在 HOTA, MOTA, IDF1, AssA 和 MT 方面分别提升了 1.9%、3.7%、2.4%、1.3%和 11.8%, 表明所提方法具有更优异的跟踪准确性和鲁棒性。与同样使用了 BYTE 的 ByteTrack 相比, FDBTrack 在 HOTA、MOTA、IDF1 等方面都表现更优秀。

相比与 Deep OC-SORT, 在 MOTA 和 MT 方面, FDBTrack 分别领先了 3.1%和 10.3%, 但在 HOTA、IDF1 和 AssA 等指标上稍逊一筹。FDBTrack 在跟踪性能和关联速度之间找到了一个平衡点。相对于结合了外观线索和 CMC 的方法(如 Deep OC-SORT, BOT-SORT), 虽然所提方法的跟踪性能稍逊一些, 但在关联速度上优势更大, 分别是 Deep OC-SORT 和 BOT-SORT 的 6.6 倍和 4.9 倍。

综上, 所提方案均衡的性能表现和关联速度使其成为现有在线 MOT 方法的有价值补充。

3.4.7 基准评估

为了在 MOTChallenge 基准中进行公平的比较, 所提方法在提交时, 使用了与 ByteTrack 相同的简单线性插值^[27]。表 3-6 和 3-7 报告了所提方法与近期的在线 SOTA

方法们在 MOT17 和 MOT20 测试集上的公开结果。值得注意的是，直接比较不同方法间的 FPS 是困难的，因为其所声称的速度取决于实现的设备。

表 3-6 MOT17 测试集基准评估结果(最优数据用粗体标出)

方法	App	JDE	CMC	HOTA↑	MOTA↑	IDF1↑	AssA↑	FP↓(10 ⁴)	FN↓(10 ⁴)	IDs↓	FPS↑
TransTrack ^[48]	√			54.1	75.2	63.5	47.9	5.0	8.6	3603	59.2
CSTrack ^[25]	√	√		59.3	74.9	72.6	57.9	2.4	11.4	3567	15.8
FairMOT ^[14]	√	√		59.3	73.7	72.3	58.0	2.8	11.7	3303	25.9
CrowdTrack ^[83]			√	60.3	75.6	73.6	59.3	2.6	10.9	2544	140.8
RelationTrack ^[26]	√	√		61.0	73.8	74.7	61.5	2.8	11.9	1374	8.5
ByteTrack ^[27]				63.1	80.3	77.3	62.0	2.5	8.4	2196	29.6
OC-SORT ^[28]				63.2	78.0	77.5	63.2	1.5	10.8	1950	29.0
Deep OC-SORT ^[30]	√		√	64.9	79.4	80.6	65.9	1.7	9.9	1023	28.1
BOT-SORT ^[11]	√		√	65.0	80.5	80.2	65.5	2.3	8.6	1257	6.8
FDBTrack(ours)	√			63.2	79.9	77.7	62.3	2.8	8.3	3045	18.7

表 3-7 MOT20 测试集基准评估结果(最优数据用粗体标出)

方法	App	JDE	CMC	HOTA↑	MOTA↑	IDF1↑	AssA↑	FP↓(10 ⁴)	FN↓(10 ⁴)	IDs↓	FPS↑
TransTrack ^[48]	√			48.5	65.0	59.4	45.2	2.7	15.0	3608	14.9
CSTrack ^[25]	√	√		54.0	66.6	68.6	54.0	2.5	14.4	3198	4.5
FairMOT ^[14]	√	√		54.6	61.8	67.3	54.7	10.3	8.9	5243	13.2
CrowdTrack ^[83]			√	55.0	70.7	68.2	52.6	2.2	12.7	3198	9.5
RelationTrack ^[26]	√	√		56.5	60.6	70.5	55.8	11.3	8.5	4243	3.0
ByteTrack ^[27]				61.3	77.8	75.2	59.6	2.6	8.8	1223	17.5
OC-SORT ^[28]				62.1	75.5	75.9	62.0	1.8	10.8	913	18.7
Deep OC-SORT ^[30]	√		√	63.3	77.8	77.5	62.9	2.5	8.9	1313	5.6
BOT-SORT ^[11]	√		√	63.9	75.6	79.2	65.7	1.7	10.8	779	5.1
FDBTrack(ours)	√			62.0	75.9	75.9	61.5	2.1	10.2	1452	10.1

如表 3-6 和 3-7 所示，所提方法在 2 个数据集上都取得了有竞争性的表现。无论是在 MOT17 还是在更具挑战性的 MOT20 上，所提方法的跟踪性能均优于近期流行的 JDE 类方法。例如，与 RelationTrack 相比，在 MOT17 上，所提方法的 HOTA、MOTA、IDF1 和 AssA 分别领先 2.2、6.1、3.0 和 0.8。在 MOT20 上，这种优势进一步扩大到了 5.5、15.3、5.4 和 5.7，而且 IDs 仅为 RelationTrack 的 34.2%。这说明较之 JDE 类方法，

所提方法拥有更高的跟踪准确度和鲁棒性，且更能应付复杂场景。

与所提方法的基线之一 ByteTrack 相比，其在 HOTA、IDF1 和 AssA 指标上有改善。例如，在 MOT17 上，所提方法的 HOTA、IDF1 和 AssA 分别领先了 0.1、0.4 和 0.3。在 MOT20 上，则分别为 0.7、0.7 和 1.6，改善更为明显。表明所提轻量级深度外观提取方案能有效提升 TBD 类方法在复杂场景下的跟踪性能。

与近期流行的基于 CMC 的方法 BOT-SORT 和 Deep OC-SORT 相比，所提方法在跟踪性能方面还存在一定差距。但是，由于不涉及 CMC 过程，所提方法具备更快的关联速度。这使其对时间敏感的下游任务更为友好。此外，所提方案与 CMC 并不冲突，可以根据实际应用需求进行灵活组合。

3.4.8 定性分析

图 3-4 展示了所提 FDBTrack、DeepSORT 和 ByteTrack 在 MOT17-05 上的部分跟踪结果。该视频为手持运动拍摄，存在大量的视角运动与长期遮挡。

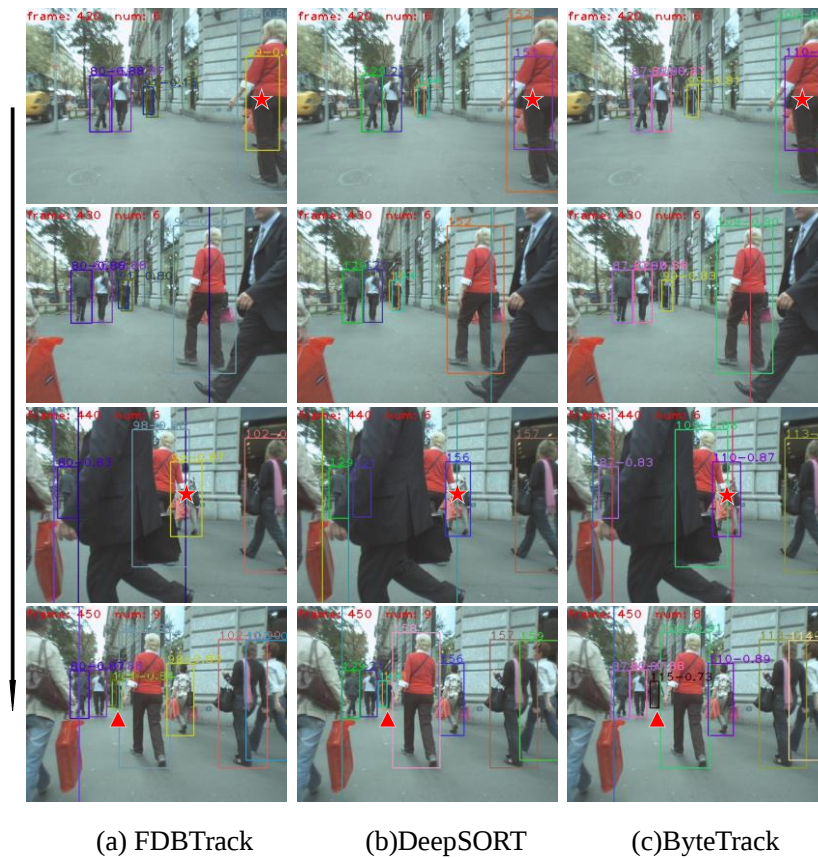


图 3-4 定性实验运行结果(MOT17-05: # 420、#430、#440 和 #450)

由图 3-4 可见，在第 420 帧所有方法均成功跟踪了红色五角星处的高遮挡目标。

在第 430 帧，该目标被红衣女子完全遮挡。当第 440 帧该目标重新出现时，所提方法与 ByteTrack 均成功重连了轨迹，但 DeepSORT 重连失败，目标 ID 由 151 变为 156，发生了轨迹切换。此外，在第 450 帧，所提方法与 DeepSORT 均成功跟踪了红色三角形所指的小目标，而 ByteTrack 则没有跟踪到。由此可见，相较于另外 2 种方法，所提方法在复杂场景下展现了更加鲁棒的跟踪效果。

3.5 本章小结

本章提出了一种轻量级深度外观提取方案，其由轻量级 Re-ID 网络 Fast OSNet 和低代价轨迹外观更新方法 HAEMA 组成。具体来说，Fast OSNet 在 OSNet 的基础上简化了结构、添加了通道和空间注意力机制，引入了全局与局部特征融合机制。在 Market1501 数据集上的消融实验显示，较其基线 OSNet，Rank-1 和 mAP 分别提升 1.4 和 3.7，推理速度增加 69.0%，内存占用减少 19.4%。在 EMA 基础上，加入轨迹外观分数更新过程，通过二段线性变化构建自适应外观更新权重，提出 HAEMA。消融实验显示，其能以较低的代价，提升 MOT 算法的跟踪准确度和鲁棒性。此外，所提 MOT 方法 FDBTrack，关联速度是其基线 DeepSORT 的 1.6 倍，且跟踪准确度和鲁棒性提升显著；与 ByteTrack 相比，虽速度较慢，但在复杂场景下的跟踪效果更为鲁棒。另一方面，其在 MOT17 测试集上实现了 63.2 的 HOTA 和 77.7 的 IDF1；在 MOT20 测试集上则为 62.0 和 75.9，较近期的 JDE 类方法更有优势。虽跟踪性能落后于使用了 CMC 的 SOTA 方法，但更适用于时间敏感场景。

第 4 章 置信度优化全检测关联在线多目标跟踪研究

MOT 旨在得出视频流中多个对象的轨迹与身份标识符。近期的 SOTA 方法们^[11, 27, 28, 30]利用置信度门限阈值实现多阶段的数据关联。然而，人为设置的阈值会造成算法面对不同场景时的不稳定性。而且，在实际算法部署时，这些阈值需根据应用场景的具体需求人为调整，过多的阈值设置无疑为 MOT 算法部署带来挑战。

针对以上局限性，摒弃了依赖置信度门限硬性划分匹配成本矩阵的思路，创新地引入 EMA 以迭代的建模轨迹置信度，并将结合其与检测置信度构建加权矩阵，为匹配过程引入轨迹与检测的置信度信息，以优化匹配成本矩阵，减少模糊的匹配关系，提升关联准确率。以此为基础，引入考察期隔离的关联策略，提出了一种二阶段的全检测关联跟踪器，CWTracker，其能直接处理所有检测器输出，实现了全检测关联，避免了检测集合划分步骤。此外，为探索其改进潜力，引入了近期流行的 CMC 和外观线索，提出了进阶框架 CWTracker++。为验证所提跟踪器的可行性与有效性，在 MOT17 和 MOT20 数据集上开展了一系列实验。首先，在验证集上进行了消融实验，可视化了各模块的跟踪结果，解释了所提置信度优化方法的有效性。其次，与 SOTA 方法在相同设备上开展了对比实验，验证了全检测关联的价值。最后，在测试集上进行了基准评估，并通过定性分析以显示所提方法在多种场景下的跟踪效果。

4.1 轨迹置信度

轨迹置信度能量化的衡量一条轨迹的可靠性，主要与两个因素有关，即与对应检测间的匹配成本和目标遮挡程度^[15]。匹配成本越低说明轨迹越符合运动模型的预测，目标遮挡程度越低说明轨迹受到的遮挡噪声干扰越少。匹配成本可以由轨迹与检测间的各类距离表示，而遮挡程度与检测的置信度分数负相关^[15]。此外，轨迹置信度需要综合考量轨迹在一段时间窗口内的所有状态，且更倾向于近期的表现。因此，以轨迹与对应检测间的 IOU 距离和检测置信度作为参量，基于 EMA 方法，建模轨迹置信度如式(4-1)所示。

$$\begin{cases} \vartheta_j^k = s_i^k & k = t_j^s \\ \vartheta_j^k = (\vartheta_j^{k-1} + (\text{IOU}(\mathbf{L}_i^k, \hat{\mathbf{L}}_j^k) + s_i^k)/2)/2 & \text{otherwise} \end{cases} \quad (4-1)$$

式中, $\text{IOU}(\mathbf{L}_i^k, \hat{\mathbf{L}}_j^k)$ 表示第 k 帧检测框 i 与轨迹 j 的预测框间的 IOU 距离。不同于参考文献^[15]提出的基于历史状态池的轨迹置信度模型, 所提模型是迭代更新的, 计算复杂度低且参数无需额外求解, 于在线任务而言是有利的。图 4-1 展示了面对目标遮挡时 ϑ 的变化。

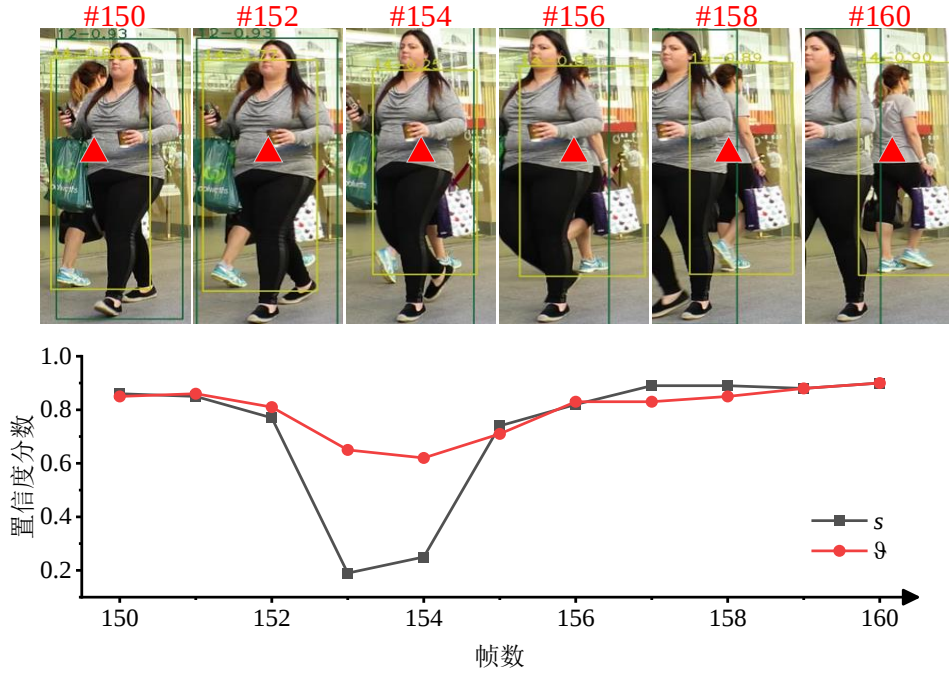


图 4-1 在 MOT17-09 序列中, 目标 14(红色三角)在遮挡情况下 ϑ_{14}^k 的变化

由图 4-1 可见, 目标 14 先后历经部分遮挡、完全遮挡和无遮挡 3 个阶段。在第 152 帧至第 156 帧期间, 因同类目标的遮挡, 导致检测框失真, 进而影响到轨迹精度, 相应的 s_{14}^k 和 ϑ_{14}^k 均出现下降。较之 s_{14}^k , ϑ_{14}^k 的曲线更为平滑。随着目标摆脱遮挡, s_{14}^k 和 ϑ_{14}^k 均逐渐增加。因此, ϑ_{14}^k 能够及时反映轨迹的可靠性变化, 且较之 s_{14}^k 鲁棒性更好。

4.2 CWTracker

所提方法与 ByteTrack 的流程如图 4-2 所示。由图 4-2 可见, 与 ByteTrack 不同, 所提方法未通过人为设置的置信度阈值来划分检测框, 而是通过 FCW 和 SCW 引入额外的线索, 从而降低匹配对间的模糊性, 避免低置信度检测的干扰检测。而且, 所提方法没有独立的新轨迹关联阶段, 二次关联起到了激活新轨迹的作用。此外, 通过引

入 CMC 和外观模块，CWTracker 可以方便的拓展为其进阶版 CWTracker++。CWTracker 的具体框架细节如图 4-3 所示。

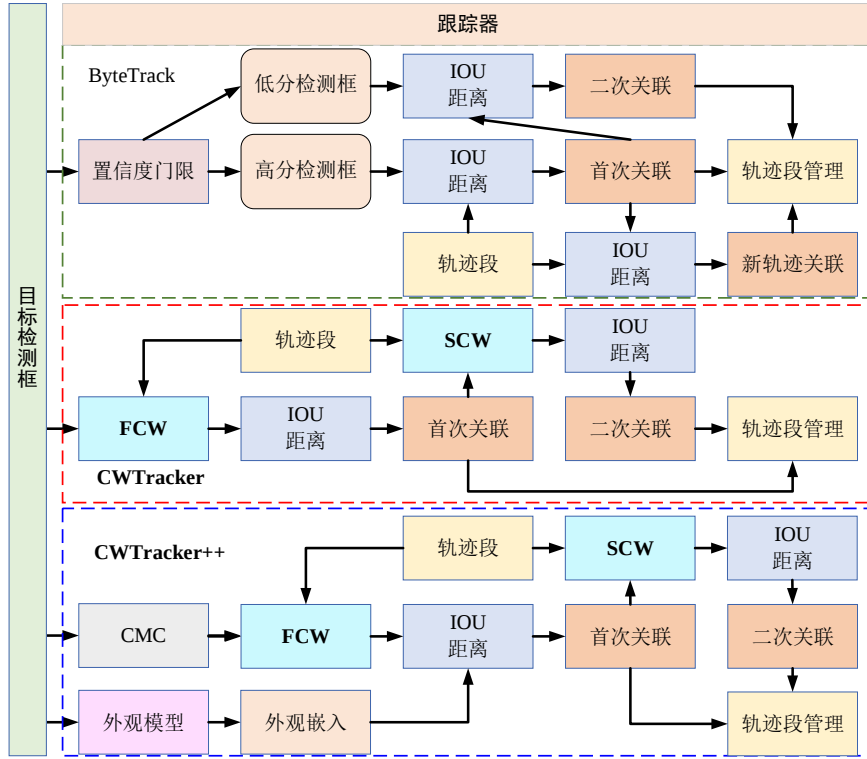


图 4-2 CWTracker、CWTracker++ 与 ByteTrack 的流程

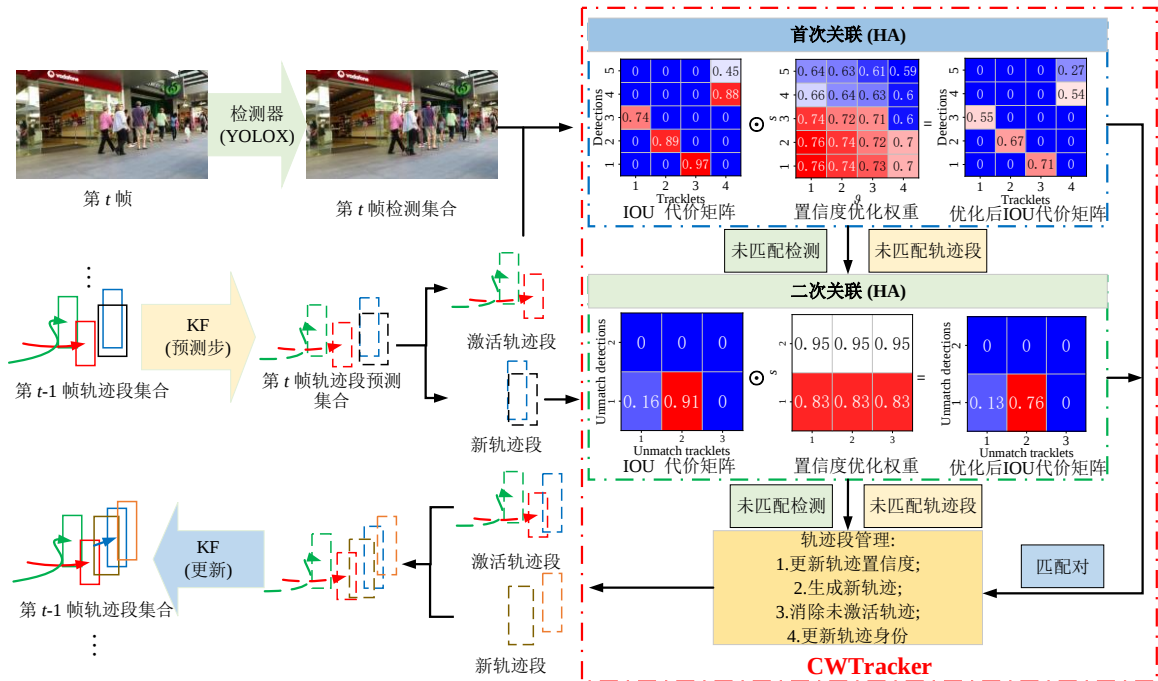


图 4-3 所提跟踪器框架

如图 4-3 红点划线框内所示，所提 CWTracker 可以分为 3 个主要部分，即首次关

联、二次关联和轨迹管理。

首次关联：在首次关联时，它允许所有检测器的输出和激活轨迹(包括正常匹配轨迹和丢失轨迹)参与关联。这将有助于从低分检测中恢复中断的轨迹，并省去阈值优化过程，方便实际部署。所提方法遵循与先前跟踪器相同的匹配成本矩阵构建方法。它采用线性 KF 作为运动模型，以预测所有激活轨迹在当前帧的位置，进而构建 IOU 相似度矩阵 $\mathbf{U}_{\text{first}}$ 。除此之外，所提方法构建了所有检测的置信度向量 $\mathbf{s}$ 和所有激活轨迹的置信度向量 $\mathbf{\vartheta}$ ，如式(4-2)和(4-3)所示。

$$\mathbf{s} = [s_1^t, s_2^t, s_3^t, \dots, s_n^t]^T \quad (4-2)$$

$$\mathbf{\vartheta} = [\vartheta_1^{t-1}, \vartheta_2^{t-1}, \vartheta_3^{t-1}, \dots, \vartheta_m^{t-1}] \quad (4-3)$$

通过计算 $\mathbf{s}$ 和 $\mathbf{\vartheta}$ 的外积得出置信度矩阵 $\mathbf{W}_{\text{first}}$ ，然后将其与 $\mathbf{U}_{\text{first}}$ 求范德蒙德积，得出优化后的首次关联成本矩阵 $\mathbf{C}_{\text{first}}$ ，如式(4-4)所示。

$$\mathbf{C}_{\text{first}} = \mathbf{1} - (\mathbf{U}_{\text{first}} \odot (\mathbf{s} \otimes \mathbf{\vartheta})) \quad (4-5)$$

实际上，首次关联问题模型如式(4-6)所示，为成本矩阵的各元素引入额外的轨迹与检测置信度线索，从而降低匹配对间的模糊性，提升关联准确性。

$$\begin{aligned} \mathbf{A}_{\text{first}}^* = \underset{\mathbf{A}_{\text{first}} \in \{0,1\}^{n \times m}}{\operatorname{argmin}} \quad & \sum_{i=1}^n \sum_{j=1}^m a_{ij} g_{ij} \left\{ 1 - \left(s_i^t \vartheta_j^{t-1} \text{IOU}(\mathbf{L}_i^t, \hat{\mathbf{L}}_j^t) \right) \right\} \\ \text{s. t.} \quad & \begin{cases} \sum_{i=1}^n a_{ij} \leq 1, \forall m \\ \sum_{j=1}^m a_{ij} \leq 1, \forall n \\ \sum_{i=1}^n \sum_{j=1}^m a_{ij} \leq \min\{n, m\} \end{cases} \end{aligned} \quad (4-6)$$

式中 g_{ij} 为门限控制系数，用以过滤不可能的匹配关系，如式(4-7)所示。

$$g_{ij} = \begin{cases} 1 & 1 - \left(s_i^t \vartheta_j^{t-1} \text{IOU}(\mathbf{L}_i^t, \hat{\mathbf{L}}_j^t) \right) \leq \varepsilon_{\text{first}} \\ +\infty & \text{otherwise} \end{cases} \quad (4-7)$$

式中 $\varepsilon_{\text{first}}$ 为首次关联的最大成本阈值。通过 HA 求解式(4-6)，得出首次关联的近似最优匹配矩阵 $\mathbf{A}_{\text{first}}^*$ ，并输出匹配轨迹、未匹配轨迹和未匹配检测集合。

二次关联：所提方法与 ByteTrack 一致，均对新轨迹采取了隔离匹配的策略。因为这些轨迹很可能是检测器误检造成的，如果同正常轨迹一同参与匹配，则可能引起大量的关联错误。此外，对于首次关联未匹配的正常轨迹，二次关联也提供了一次重连的机会。值得注意的是，丢失轨迹不能参与二次关联，其只能在首次关联中恢复身份。综上，二次关联的参与对象包括新轨迹、首次关联中未匹配的正常轨迹和首次关

联中未匹配的检测。

与首次关联相同，先构建轨迹与检测间的 IOU 相似度矩阵 $\mathbf{U}_{\text{second}}$ ；其次得出二次关联的置信度权重矩阵 $\mathbf{W}_{\text{second}}$ ，并得出二次关联成本矩阵 $\mathbf{C}_{\text{second}}$ ，如式(4-8)所示。值得注意的是，因为新轨迹诞生时间较短，其轨迹置信度难以可靠的反映轨迹的可信度，因此 $\mathbf{W}_{\text{second}}$ 仅与检测的置信度有关。

$$\mathbf{C}_{\text{second}} = \mathbf{1} - (\mathbf{U}_{\text{second}} \odot (\mathbf{s} \otimes \mathbf{t})) \quad (4-8)$$

式中， $\mathbf{t}$ 为一个 $1 \times m$ 的行向量。二次关联的问题模型如式(4-9)所示。

$$\begin{aligned} \mathbf{A}_{\text{second}}^* = \underset{\mathbf{A}_{\text{second}} \in \{0,1\}^{n \times m}}{\operatorname{argmin}} \quad & \sum_{i=1}^n \sum_{j=1}^m a_{ij} g_{ij} \{1 - (s_i^t \text{IOU}(\mathbf{L}_i^t, \hat{\mathbf{L}}_j^t))\} \\ \text{s. t.} \quad & \begin{cases} \sum_{i=1}^n a_{ij} \leq 1, \forall m \\ \sum_{j=1}^m a_{ij} \leq 1, \forall n \\ \sum_{i=1}^n \sum_{j=1}^m a_{ij} \leq \min\{n, m\} \end{cases} \end{aligned} \quad (4-9)$$

其中， g_{ij} 为门限控制系数，如式(4-10)所示。

$$g_{ij} = \begin{cases} 1 & 1 - (s_i^t \text{IOU}(\mathbf{L}_i^t, \hat{\mathbf{L}}_j^t)) \leq \varepsilon_{\text{second}} \\ +\infty & \text{otherwise} \end{cases} \quad (4-10)$$

式中 $\varepsilon_{\text{second}}$ 为二次关联的最大成本阈值。通过 HA 求解式(4-9)，得二次关联的近似最优匹配矩阵 $\mathbf{A}_{\text{second}}^*$ ，最终输出匹配轨迹、未匹配轨迹和未匹配检测。

轨迹管理：对于两次关联后的轨迹与检测需进行一系列状态管理，如下：

(1)生成新轨迹：CWTracker 采取了与 ByteTrack 相同的轨迹生成策略，即将经历二次关联的未匹配检测，高于置信度门限 φ 的检测全部转为新轨迹($I = 0$)，低于该阈值的删除。

(2)消除非激活轨迹：CWTracker 为丢失轨迹设置了一段缓冲期，位于缓冲期内的轨迹可以参与首次关联，以从丢失状态中恢复 ID。若丢失轨迹在超过缓冲期后，仍未恢复 ID，则状态转为非激活($I = 4$)，不再参与关联。

(3)更新轨迹状态：对于参与关联的轨迹，需要根据 $\mathbf{A}_{\text{first}}^*$ 和 $\mathbf{A}_{\text{second}}^*$ 更新所有轨迹的 I 。具体而言，对于连续关联的正常轨迹，继续保持 $I = 1$ ；对于连续 2 帧匹配成功的新轨迹，将其转为正常轨迹($I = 1$)；对于位于缓冲期内的未匹配轨迹，将其转为丢失轨迹($I = 2$)；对于从丢失状态中恢复过来的轨迹，将其转为重新找回轨迹($I = 3$)。

(4)更新轨迹置信度：在完成以上 3 个步骤后，依式(4-1)更新轨迹置信度。

4.3 CWTracker++

CWTracker 是一个基于运动的跟踪器，与同期方法 ByteTrack 和 OC-SORT 一样，其同样可以附加额外的模块。通过引入 CMC 模块^[11]与外观模型 BOT(SBS)^[29]，提出其进阶版 CWTracker++，流程图见图 4-2 中蓝色虚线框，如算法 4-1 所示。输入为视频序列 V ，目标检测器 Det ，新轨迹生成门限 φ ，深度外观提取器 BOT；输出为视频中所有感兴趣目标的轨迹，即每个目标在每帧的包围框与 ID 号。

算法 4-1: CWTracker++

输入: 视频序列 V ；目标检测器 Det ； φ ；深度外观模型 BOT

输出: 视频的多目标轨迹 $\mathbb{T}_{\text{finish}}$

```

1:   初始化:  $\mathbb{T}_{\text{start}} \leftarrow \emptyset$ ;
2:   for frame in  $V$  do
3:        $\mathbb{D}_t \leftarrow \text{Det}(\text{frame})$ 
4:       由 KF 预测  $\mathbb{T}_{t-1}$  的  $\hat{L}_j^t$ 
5:        $\hat{L}_j^t \leftarrow \text{CMC}(\hat{L}_j^t, \text{frame})$  // 细节见参考文献[11]//
6:        $\mathbb{T}_{\text{activate}}, \mathbb{T}_{\text{new}} \leftarrow \text{Divide}(\mathbb{T}_{t-1})$ 
7:        $f\text{set} \leftarrow \text{BOT}(\mathbb{D}_t, \text{frame})$ 
8:       由式(4-11)计算  $C_{\text{first}}$ 
9:        $A_{\text{first}}^* \leftarrow \text{第一次关联}(\mathbb{D}_t, \mathbb{T}_{\text{activate}})$ 
10:      得出  $\mathbb{D}_{\text{unmatch}}$ 、 $\mathbb{T}_{\text{unmatch}}$  和  $\mathbb{T}_{\text{first-match}}$ 
11:       $\mathbb{T}_{\text{second}} \leftarrow \mathbb{T}_{\text{new}} \cup (\mathbb{T}_{\text{unmatch}} \mid I \neq 2)$ 
12:      由式(4-8)计算  $C_{\text{second}}$ 
13:       $A_{\text{second}}^* \leftarrow \text{第二次关联}(\mathbb{D}_{\text{unmatch}}, \mathbb{T}_{\text{second}})$ 
14:      得出  $\mathbb{D}_{\text{unmatch}}$ 、 $\mathbb{T}_{\text{unmatch}}$  和  $\mathbb{T}_{\text{second-match}}$ 
15:       $\mathbb{T}_{\text{match}} \leftarrow \mathbb{T}_{\text{first-match}} \cup \mathbb{T}_{\text{second-match}}$ 
16:      由式(4-12)更新  $e$ 
17:       $\mathbb{T}_{\text{new}} \leftarrow \text{Create}(\mathbb{D}_{\text{unmatch}}, \varphi)$ 
18:       $\mathbb{T}_{\text{delete}} \leftarrow \text{Delete}(\mathbb{T}_{\text{unmatch}})$ 
19:      更新  $\mathbb{T}_{\text{match}}$  和  $\mathbb{T}_{\text{new}}$  的  $\vartheta$ 
20:      由 KF 更新  $\mathbb{T}_{\text{match}}$  和  $\mathbb{T}_{\text{new}}$  的  $L_j^t$ 
21:       $\mathbb{T}_t \leftarrow \mathbb{T}_{\text{match}} \cup \mathbb{T}_{\text{unmatch}} \cup \mathbb{T}_{\text{new}}$ 
22:       $\mathbb{T}_t \leftarrow \mathbb{T}_t \setminus \mathbb{T}_{\text{delete}}$ 
23:   end
24:   返回:  $\mathbb{T}_{\text{finish}}$ 

```

在视频序列开始，初始化空轨迹集合 $\mathbb{T}_{\text{start}}$ 。在第 t 帧，首先，通过 Det 得出所有目

标的包围框与置信度分数，用 KF 预测 $\mathbb{T}_{t-1}$ 在 t 时刻的包围框位置，利用 CMC 模块对预测进行补偿，划分出激活轨迹与新轨迹，并对所有检测进行深度外观特征提取，得出外观嵌入向量 $\mathbf{f}$ 的集合(1~7 行)；其次，融合 IOU 与外观线索，生成置信度权重矩阵，得出 $\mathbf{C}_{\text{first}}$ ，如式(4-11)所示，并进行首次关联得出 $\mathbf{A}_{\text{first}}^*$ (8~11 行)；再次，进行二次关联得出 $\mathbf{A}_{\text{second}}^*$ ，此处与 CWTracker 完全相同(12~15 行)；最终，依照两次关联求得的匹配关系，管理所有轨迹，与基础框架不同的是，增加了对轨迹外观状态 $\mathbf{e}$ 的更新步骤，如式(4-12)所示。如此，遍历循环整个序列，得出所有轨迹集合 $\mathbb{T}_{\text{finish}}$ 。

$$\mathbf{C}_{\text{first}} = \mathbf{1} - \{(\lambda \mathbf{U}_{\text{IOU}}^{\text{first}} + (1 - \lambda) \mathbf{U}_{\text{app}}^{\text{first}}) \odot (\mathbf{s} \otimes \boldsymbol{\vartheta})\} \quad (4-11)$$

式(4-11)是基于式(4-6)得出的，区别在于式(4-11)加入了轨迹与检测间外观嵌入向量 $\mathbf{f}$ 的余弦相似度矩阵 $\mathbf{U}_{\text{app}}^{\text{first}}$ ， λ 为融合权重。式(4-12)为基于 EMA 模型的轨迹外观状态更新方法^[24]， r 为平滑系数。

$$\mathbf{e}_j^k = \begin{cases} \mathbf{f}_i^k, & t = t_j^s \\ (1 - r) \mathbf{e}_j^{k-1} + r \mathbf{f}_i^k, & \text{otherwise} \end{cases} \quad (4-12)$$

4.4 实验与分析

4.4.1 实验设置

为了公平地与 SOTA 方法们比较，在消融和对比实验中，所有方法均使用了 ByteTrack 提供的 YOLOX 检测器及对应的网络权重。CWTracker 的 φ 设置与 ByteTrack 相同， $\varepsilon_{\text{first}}$ 和 $\varepsilon_{\text{second}}$ 分别为 0.9 和 0.7。此外，CWTracker++ 的外观模型 BOT 和 CMC 模块均来自 BOT-SORT，且 KF 设置也与其一致。轨迹外观状态更新方法采用了常见的 EMA 更新方法， r 为 0.9。实验设备配置如下：CPU 型号为 Intel(R) Core(TM) i9-12900K @ 5.0GHz，内存大小为 128GB，显卡型号为 Nvidia RTX3090。所有程序均通过 Python 3.8 和 PyTorch 1.10.1 实现。

4.4.2 消融实验

为验证所提跟踪器性能，并量化每个模块的贡献程度，包括 FCW、SCW、BOT 和 CMC，在 MOT17 和 MOT20 验证集上开展了消融实验。基线为 CWTracker 不包含

FCW 和 SCW 的状态，即不采用置信度优化。表 4-1 和表 4-2 分别报告了所提跟踪器在 MOT17 和 MOT20 验证集上的消融实验结果。

表 4-1 所提方法在 MOT17 半验证集上的消融实验结果(最优数据用粗体标出)

方案	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
基线	50.1	22.6	56.1	56.0	76.1	5.0	803	636.8
+ FCW	64.4	72.0	74.6	65.2	68.1	10.3	305	627.7
+ SCW (CWTracker)	68.0	78.0	79.5	69.4	62.2	10.9	311	624.1
+BOT	68.3	78.0	80.1	69.9	62.5	11.2	285	20.1
+CMC (CWTracker++)	68.9	78.2	81.2	71.0	63.4	10.6	168	15.5

表 4-2 所提方法在 MOT20 半验证集上的消融实验结果(最优数据用粗体标出)

方案	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
基线	46.9	46	59.1	43.9	72.1	8.0	3612	150.9
+ FCW	57.7	71.9	72.6	54.6	66.6	9.0	1421	146.3
+ SCW (CWTracker)	59.6	74.4	76.1	57.9	56.8	9.9	1317	144.2
+BOT	59.8	74.4	76.3	58.2	56.8	9.9	1293	3.7
+CMC (CWTracker++)	59.9	74.4	76.4	58.4	56.8	9.9	1277	3.4

由表 4-1 可知，引入 FCW 后，显著提升了 HOTA、MOTA、IDF1 和 AssA，分别增加了 14.3%、49.4%、18.5%和 9.2%，IDs 降低至 305。展现了 FCW 对跟踪和关联准确性的积极影响。关联速度轻微下降，但仍保持在可接受范围。继续加入 SCW 模块后，HOTA、MOTA、IDF1 和 AssA 分别提升 3.6%、6.0%、4.9%和 4.2%，IDs 略微上升。虽然，提升幅度不如 FCW 显著，但仍有效改善了跟踪性能。此外，在 MOT20 验证集上，FCW 和 SCW 也带来了类似的性能改观。这说明 FCW 和 SCW 对于极端拥挤场景是同样有效的。

如表 4-1 所示，BOT 的整合使得 HOTA、IDF1、AssA 和 MT 略有改善，增量分别为 0.3%、0.6%、0.5%和 0.3%。然而，这种改进是以显著降低关联速度为代价的，关联速度降至 20.1。随后添加的 CMC 在 HOTA、MOTA、IDF1、AssA 和 MT 方面产生了进一步的改进。尽管关联速度降至 15.5 FPS，但它仍然满足实时要求。在 MOT20 验证集上，BOT 和 CMC 都有助于提高跟踪性能。然而，它们的效果不太明显，关联速度降至 3.4 FPS。这可以归因于 MOT20 数据集中的高度拥挤，这影响了外观线索的可

靠性。此外，固定的拍摄视角会降低 CMC 的有效性。

综上，各个模块均能有效提升跟踪性能，但代价各不相同。FCW 和 SCW 模块在略微牺牲关联速度的情况下，实现了显著的性能提升。BOT 和 CMC 则会造成严重的关联速度下降，且在类似 MOT20 数据集的应用场景上作用有限。因此，在实际应用中需权衡使用各个模块。此外，基线较低的性能也是导致 FCW 模块能取得巨大性能提升的原因之一。为了探究基线性能较低的原因，图 4-4 可视化的显示了基线、+FCW 和+SCW 在多个视频序列上的跟踪结果。

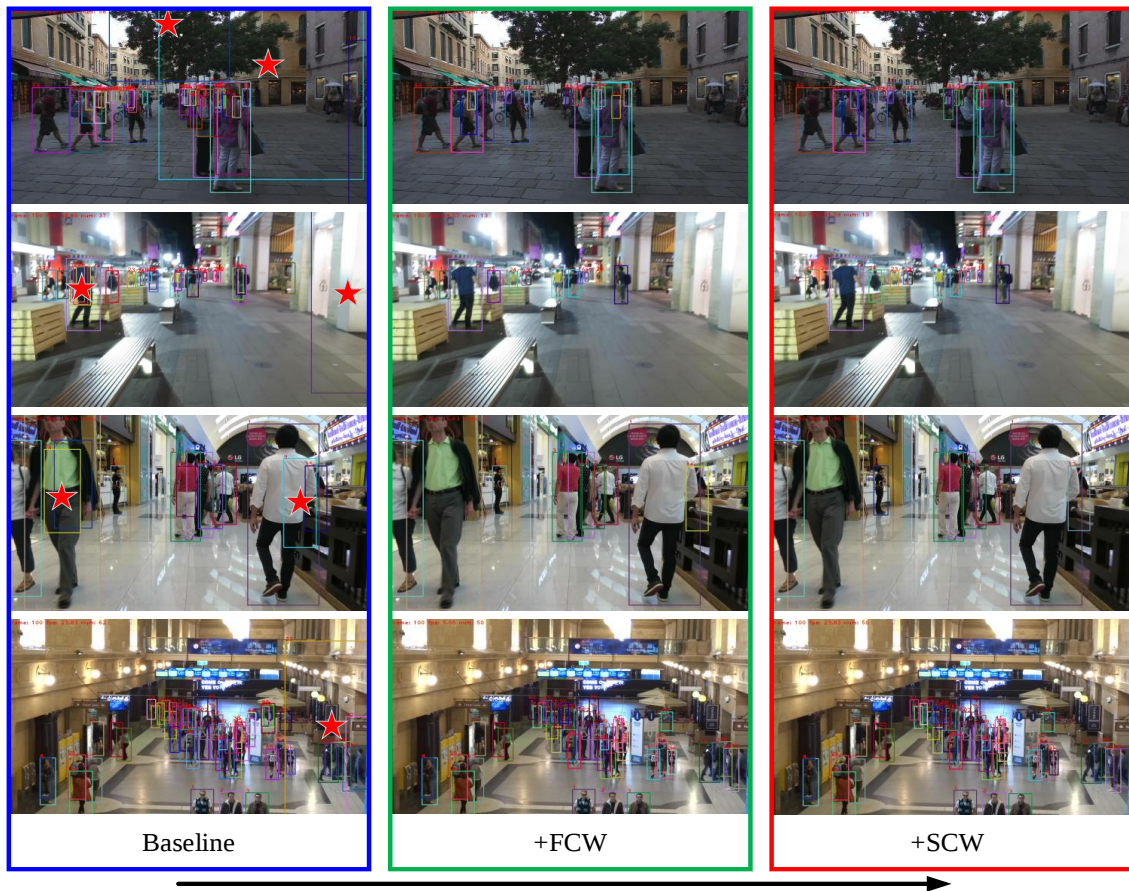


图 4-4 可视化跟踪结果

由图 4-4 可见，基线的跟踪结果中(蓝框)，存在大量明显的 FP 轨迹(红色五角星)，产生这些轨迹的主要原因是面对复杂场景时检测器容易产生大量的低置信度误检。在传统跟踪器中，如 SORT 和 ByteTrack，这些检测不会参与到高置信度检测的匹配过程中。然而，所提框架并没有隔离二者，而是让全检测都参与了关联过程，这导致了较低的性能表现。在添加所提 FCW 后(绿框)，为首次关联引入了置信度线索，低置信度的检测和轨迹被赋予了更高的匹配成本，有效抑制了 FP 轨迹。继续引入 SCW 后，

参与二次关联的低置信度检测同样被赋予了严格的限制，进一步减少了 FP 轨迹的产生。综上，FCW 和 SCW 对于 CWTracker 都是极其重要的，它们能够有效抑制低置信度检测对关联的干扰，并从高遮挡的低置信度检测中恢复被遮挡轨迹。

4.4.3 对比实验

为比较其他在线 SOTA 方法与所提方法在同一运行环境下的性能，在 MOT17 和 MOT20 验证集上开展了对比实验，表 4-3 和表 4-4 分别报告了复现的各个 SOTA 方法在两个数据集上的运行结果。注意，所有方法均未使用后处理技术。

表 4-3 所提方法与对比方法在 MOT17 验证集上结果(最优数据用粗体标出)

方法	App	CMC	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
OC-SORT ^[28]			66.1	74.5	77.6	68.6	48.7	16.5	233	498.7
ByteTrack ^[27]			67.7	77.7	79.3	69.5	58.1	12.7	213	612.3
Deep OC-SORT ^[30]	√	√	68.3	75.1	81.0	72.4	50.2	15.3	138	14.2
BOT-SORT ^[11]	√	√	69.1	78.5	81.9	74.4	60.5	11.2	162	17.0
CWTracker			68.0	78.0	79.5	69.4	56.8	9.9	311	624.1
CWTracker++	√	√	68.9	78.2	81.2	71.0	63.4	10.6	168	15.5

表 4-4 所提方法与对比方法在 MOT20 验证集上结果(最优数据用粗体标出)

方法	App	CMC	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
OC-SORT ^[28]			56.4	69.9	73.4	55.2	44.1	11.6	1275	78.8
ByteTrack ^[27]			57.4	72.7	72.7	55.1	52.5	11.1	1225	143.7
Deep OC-SORT ^[30]	√	√	57.4	69.9	74.7	56.9	44.2	11.7	1125	3.3
BOT-SORT ^[11]	√	√	58.5	74.1	74.1	56.1	57.2	10.2	1632	3.8
CWTracker			59.6	74.4	76.1	57.9	56.8	9.9	1317	144.2
CWTracker++	√	√	59.9	74.4	76.4	58.4	56.8	9.9	1277	3.4

如表 4-3 所示，在相同的运行环境下，CWTracker 展现了最快的关联速度，超过了同样基于运动的 ByteTrack 和 OC-SORT。这是由于所提方法未隔离高、低置信度检测，减少了一次关联过程，从而提升了速度。就跟踪性能而言，与未使用插值的 OC-SORT 相比，其在 HOTA、MOTA、IDF1 和 AssA 方面分别提升 1.9%、3.5%、1.9%和 0.8%。与同样使用线性 KF 的 ByteTrack 相比，虽然增长了 98 个 IDs，但在 HOTA、MOTA 和 IDF1 等多个指标上表现优异。另一方面，较之 OC-SORT 的进阶版 Deep OC-SORT，所提 CWTracker++在 HOTA、MOTA、IDF1 和 MT 方面分别提升 0.6%、

3.1%、0.2%和 13.4%，具有更好的跟踪准确度和关联鲁棒性。与 BOT-SORT 相比，所提方法在 HOTA、MOTA、IDF1 和 AssA 等主要指标上稍逊一筹。就关联速度而言，三者间差距并不明显。

如表 4-4 所示，在 MOT20 验证集上，较之 OC-SORT，CWTracker 在 HOTA、MOTA、IDF1 和 AssA 方面分别提升了 3.2%、4.5%、2.7%和 2.7%。与 ByteTrack 相比，也取得了显著的优势(+2.2% HOTA、+1.7% MOTA、+3.4% IDF1、+2.8 %AssA)。甚至与使用了外观线索和 CMC 模块的 BOT-SORT 相比，其也展现了更好的跟踪性能。这是由于 MOT20 数据集的目标遮挡问题更为严重，同一目标的检测置信度波动较大，硬性划分检测集合会导致对应的检测和轨迹不能同时参与关联，增加了关联错误的风险。所提方法通过置信度优化成本矩阵，避免了上述问题。

综上，所提 CWTracker 和 CWTracker++方法在关联速度、跟踪准确度和关联鲁棒性方面展现出了一定的价值，且较之其他 SOTA 方法能更好地应对复杂遮挡的拥挤场景。

与工业界流行的传统 TBD 框架(ByteTrack 和 SORT)相比，所提方法的一个关键创新在于实现了真正的全检测关联。对于传统框架而言，低置信度阈值是敏感的，需要根据场景调整。图 4-5 报告了 CWTracker、ByteTrack 和 SORT 的跟踪性能随低置信度阈值的变化情况，所提方法中的阈值是额外添加的。这 3 种方法均为基于运动的在线跟踪方法，且运动模型均为线性 KF。

由图 4-5 可见，在所有性能指标上，SORT 的性能曲线始终位于其他方法之下。它的各项指标在阈值范围为 0-0.1 时表现最差，在 0.1-0.4 的范围内逐渐上升。在 0.4-0.6 的范围内，HOTA、IDF1 和 AssA 呈上升趋势。这表明单阶段的 SORT 难以应对低置信度检测，对于阈值变化敏感。另一方面，ByteTrack 的性能曲线呈现出先升后降的趋势。在 0-0.1 的置信度阈值范围内，各项性能指标快速上升，在 0.1-0.6 的范围内，整体性能呈现平稳下降的趋势。这表明多阶段的 ByteTrack 能够有效的从 0.1-0.5 的低置信度检测中恢复轨迹信息，但还难以处理 0~0.1 范围的超低置信度检测。与之相比，所提方法的性能曲线大部分位于 ByteTrack 之上，且随着阈值的增加而平稳降低，未在 0-0.1 范围内的发生剧烈变化。这说明基于置信度优化的所提方法能够有效应对高遮

挡场景中常见的超低置信度检测，且在实际部署中无需进行阈值优化。

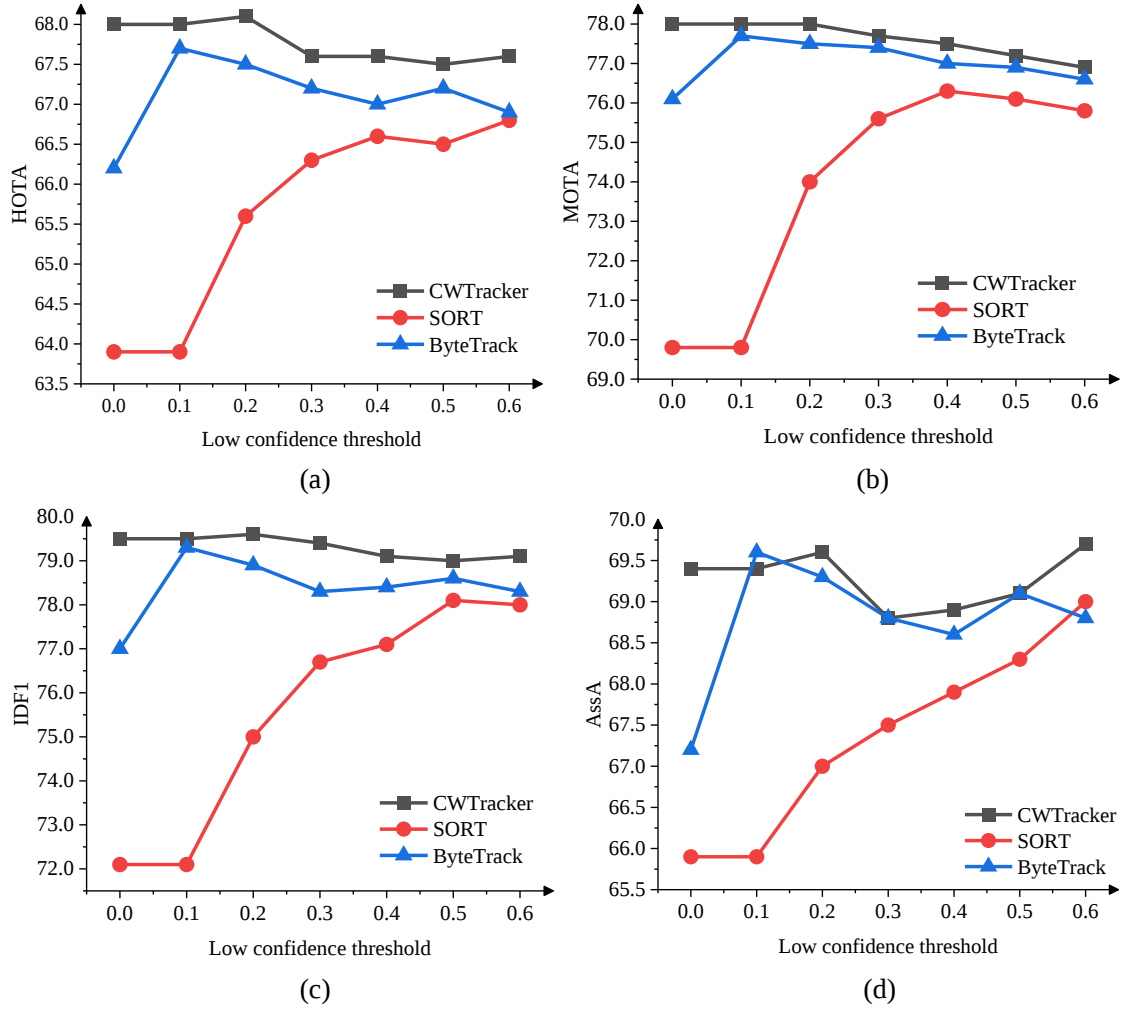


图 4-5 CWTracker、ByteTrack 和 SORT 在 MOT17 验证集上的低置信度阈值对比实验

4.4.4 基准评估

表 4-5 和 4-6 分别报告了在 MOT17 和 MOT20 测试集上，CWTracker、CWTracker++与近期的 SOTA 方法们的公开评估结果。值得注意的是，为了进行公平的比较，所提方法在提交时使用了与 ByteTrack 相同的简单线性插值。

由表 4-5 和表 4-6 的结果可知，与近期流行的方法 ByteTrack 和 OC-SORT 相比，CWTracker 在 MOT17 数据集上具备相近的 HOTA。但在 IDF1 和 AssA 指标上，与 OC-SORT 相比分别落后了 0.4%和 1.4%，这是由于 OC-SORT 在提交结果时使用了非线性的差值技术。而在 MOT20 数据集上，所提方法显著优于 ByteTrack，在 HOTA、IDF1 和 AssA 三个指标上分别领先了 1.1%、0.9%和 2.1%。与 OC-SORT 相比，在其他指标上性能相近的同时，其 MOTA 领先了 1.2%。

表 4-5 MOT17 测试集基准评估结果(最优数据用粗体标出)

方法	App	CMC	HOTA↑	MOTA↑	IDF1↑	AssA↑	FP↓(10 ⁴)	FN↓(10 ⁴)	IDs↓	FPS↑
TransTrack	√		54.1	75.2	63.5	47.9	5.0	8.6	3603	59.2
CSTrack	√		59.3	74.9	72.6	57.9	2.4	11.4	3567	15.8
FairMOT	√		59.3	73.7	72.3	58.0	2.8	11.7	3303	25.9
CrowdTrack		√	60.3	75.6	73.6	59.3	2.6	10.9	2544	140.8
RelationTrack	√		61.0	73.8	74.7	61.5	2.8	11.9	1374	8.5
ByteTrack			63.1	80.3	77.3	62.0	2.5	8.4	2196	29.6
OC-SORT			63.2	78.0	77.5	63.2	1.5	10.8	1950	29.0
Deep OC-SORT	√	√	64.9	79.4	80.6	65.9	1.7	9.9	1023	28.1
BOT-SORT	√	√	65.0	80.5	80.2	65.5	2.3	8.6	1257	6.8
CWTracker			63.0	79.6	77.1	61.8	3.1	8.0	3894	29.1
CWTracker++	√	√	64.2	80.9	78.8	63.4	2.7	7.9	1215	8.2

表 4-6 MOT20 测试集基准评估结果(最优数据用粗体标出)

方法	App	CMC	HOTA↑	MOTA↑	IDF1↑	AssA↑	FP↓(10 ⁴)	FN↓(10 ⁴)	IDs↓	FPS↑
TransTrack	√		48.5	65.0	59.4	45.2	2.7	15.0	3608	14.9
CSTrack	√		54.0	66.6	68.6	54.0	2.5	14.4	3198	4.5
FairMOT	√		54.6	61.8	67.3	54.7	10.3	8.9	5243	13.2
CrowdTrack		√	55.0	70.7	68.2	52.6	2.2	12.7	3198	9.5
RelationTrack	√		56.5	60.6	70.5	55.8	11.3	8.5	4243	3.0
ByteTrack			61.3	77.8	75.2	59.6	2.6	8.8	1223	17.5
OC-SORT			62.1	75.5	75.9	62.0	1.8	10.8	913	18.7
Deep OC-SORT	√	√	63.9	75.6	79.2	65.7	1.7	10.8	779	5.1
BOT-SORT	√	√	63.3	77.8	77.5	62.9	2.5	8.9	1313	5.6
CWTracker			62.4	76.7	76.1	61.7	3.4	8.5	1300	19.2
CWTracker++	√	√	62.7	76.8	76.4	62.1	3.4	8.5	1245	5.2

另一方面，CWTracker++在 MOT17 数据集上取得了最优的 MOTA 和 FN 指标，但在 HOTA 和 IDF1 指标上逊于 Deep OC-SORT 和 BOT-SORT。尽管如此，CWTracker++ 仍然展现出了有希望的结果。例如，在 MOT17 数据集上，相较于 CWTracker，其 HOTA、MOTA、IDF1 和 AssA 分别领先了 1.2%、1.3%、1.7%和 1.6%，并且成功降低了 68.8%的 IDs 数量。总之，CWTracker 在 2 个数据集上都展现出了有竞争力的性能，且无需精心优化置信度阈值，这对于实际部署是有价值的。尽管与最先进的方法相比，CWTracker++仍有一些差距，但它通过结合额外的线索和运动补偿，验证了进一步增强 CWTracker 的潜在途径。

4.4.5 定性分析

图 4-6 展示了在 MOT17 与 MOT20 数据集中的一些困难跟踪案例。红色五角星表示值得关注的目标，MOT20-05 中关注的是镜头中拥挤的目标群体。



图 4-6 CWTracker 跟踪结果可视化

由图 4-6 可见, 在低光照条件下, CWTracker 成功跟踪了被完全遮挡且存在运动模糊的小目标(MOT17-02 和 MOT17-10)。而且, 面对长期处于部分遮挡状态的低置信度目标(MOT17-05), 所提方法也能成功跟踪, 即使是在低对比度的情型下(MOT20-02), 依旧如此。这意味着无需依靠人为设置的置信度阈值, 采用所提置信度优化方法同样可以从低置信度检测中恢复轨迹身份。另一方面, 在面对相机快速运动(MOT17-13)和极端拥挤场景(MOT20-05)时, 所提方法也能够保持良好的轨迹跟踪性能。由此可见, 所提方法在复杂场景下鲁棒的跟踪效果。

4.5 本章小结

本章提出一种基于置信度优化的在线 MOT 方法 CWTracker。具体而言, 其以轨迹对应的检测置信度和匹配成本为参量, 基于 EMA 逐帧迭代的更新轨迹置信度, 并结合检测置信度逐元素的优化匹配成本矩阵。基于此, 构建了两阶段的数据关联框架, 实现了真正的全检测关联跟踪器, 避免了因人为阈值导致的轨迹中断。还引入近期流行的 CMC 与外观模型, 提出 CWTracker++。与 SOTA 方法在相同设备上的对比实验显示, CWTracker 在保持高关联速度的同时, 改善了跟踪准确性和关联鲁棒性, 尤其是在复杂拥挤环境中。在 MOT17 测试集上, 其实现了 63.0%的 HOTA 和 77.1%的 HOTA, 且满足实时需求。在 MOT20 上则为 62.4% HOTA 和 76.1% HOTA, 优于类似的方法 ByteTrack 和 OC-SORT, 避免了繁琐的阈值优化过程, 使其在实际应用中更具吸引力。

第 5 章 综合应用与实验验证

5.1 综合应用

第 2、3 和 4 章分别针对多目标跟踪算法的运动预测模型、外观特征提取模型与数据关联框架展开研究，以上模块间不存在冲突，故所提 PAKF、Fast OSNet 能够以插件形式与 CWTracker 组合，以构成所提综合应用方案。其流程图如图 5-1 所示，运行流程见算法 5-1。

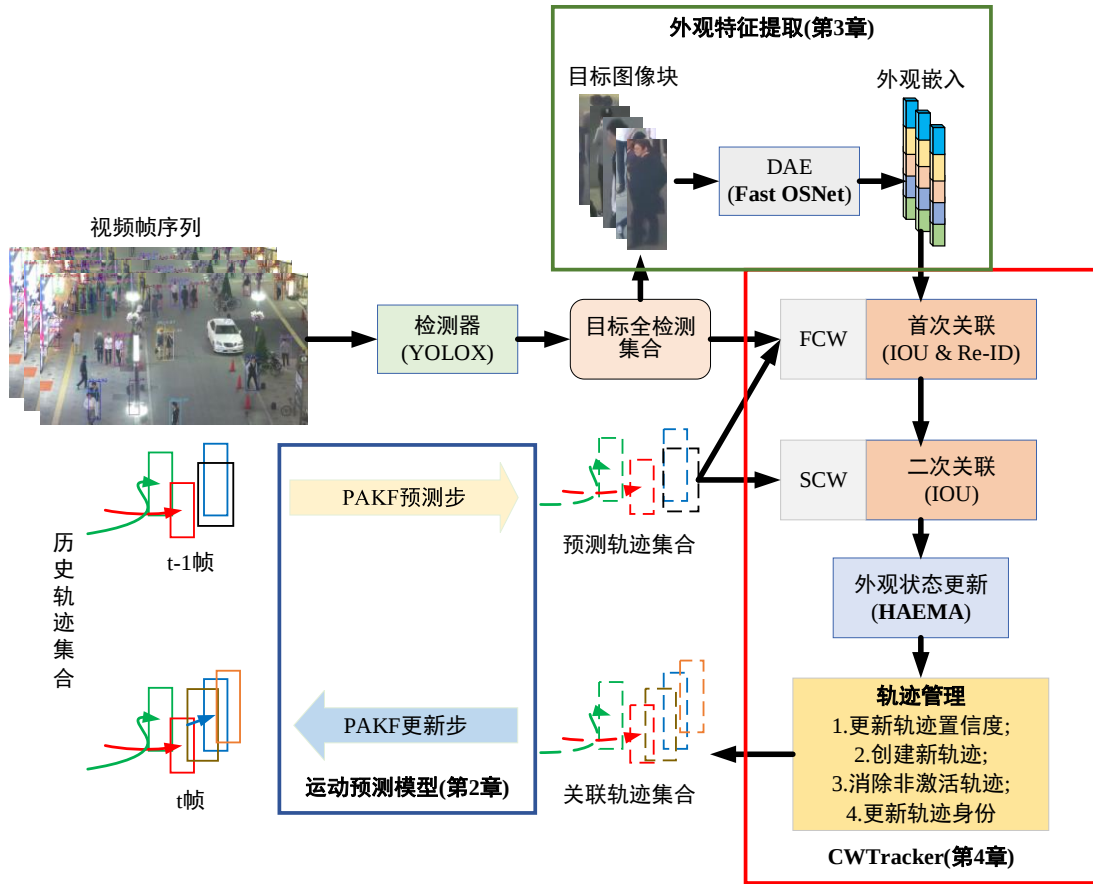


图 5-1 综合应用方案流程图

综合应用方案由目标检测器 YOLOX、所提 Fast OSNet、PAKF 与 CWTracker 组成。输入视频帧序列，输出视频中所有感兴趣目标的轨迹。

在跟踪开始时，初始化空轨迹集合 $\mathbb{T}_{\text{start}}$ 。在第 t 帧，先通过 YOLOX 得出 $\mathbf{L}$ 和 $\mathbf{s}_t$ ，用 PAKF 预测 $\mathbb{T}_{t-1}$ 在 t 时刻 $\hat{\mathbf{L}}_j^t$ ，划分出激活轨迹与新轨迹，并利用 Fast OSNet 对所有检

测目标进行深度外观特征提取，得出外观嵌入向量 $\mathbf{f}$ 的集合 (1~6 行)；其次，生成 FCW，融合 IOU 与外观成本，得出 $\mathbf{C}_{\text{first}}$ ，并进行首次关联得出 $\mathbf{A}_{\text{first}}^*$ (7~10 行)；再次，生成 SCW，进行二次关联得出 $\mathbf{A}_{\text{second}}^*$ (11~14 行)；最终，利用 HAEMA 更新 $\mathbf{e}_j^t$ ，并依照两次关联求得的匹配关系，管理所有轨迹，包括生成 $\mathbb{T}_{\text{new}}$ 、删除 $\mathbb{T}_{\text{delete}}$ 和更新 $\mathbb{T}_{\text{match}}$ ，并利用 PAKF 更新 $\mathbb{T}_{\text{match}}$ 和 $\mathbb{T}_{\text{new}}$ 的 $\mathbf{L}_j^t$ 。遍历循环序列，得出所有轨迹集合 $\mathbb{T}_{\text{finish}}$ 。

算法 5-1: 综合应用方案

输入: 视频序列 V ；目标检测器 YOLOX; Fast OSNet; PAKF

输出: 视频的多目标轨迹 $\mathbb{T}_{\text{finish}}$

```

1:   初始化:  $\mathbb{T}_{\text{start}} \leftarrow \emptyset$ ;
2:   for frame in  $V$  do
3:        $\mathbb{D}_t \leftarrow \text{YOLOX}(\text{frame})$ 
4:       利用 PAKF 预测  $\mathbb{T}_{t-1}$  的  $\mathbf{L}_j^t$ 
5:       划分  $\mathbb{T}_{\text{activate}}$ ,  $\mathbb{T}_{\text{new}}$ 
6:       外观嵌入集合  $\mathbf{f} \leftarrow \text{DAE}(\mathbb{D}_t, \text{视频帧})$ 
7:       计算 FCW 与首次关联成本  $\mathbf{C}_{\text{first}}$ 
8:        $\mathbf{A}_{\text{first}}^* \leftarrow \text{首次关联}(\mathbb{D}_t, \mathbb{T}_{\text{activate}})$ 
9:       得出  $\mathbb{D}_{\text{unmatch}}$ 、 $\mathbb{T}_{\text{unmatch}}$  与  $\mathbb{T}_{\text{first-match}}$ 
10:       $\mathbb{T}_{\text{second}} \leftarrow \mathbb{T}_{\text{new}} \cup (\mathbb{T}_{\text{unmatch}} \mid I \neq 2)$ 
11:      计算 SCW 与二次关联成本  $\mathbf{C}_{\text{second}}$ 
12:       $\mathbf{A}_{\text{second}}^* \leftarrow \text{二次关联}(\mathbb{D}_{\text{unmatch}}, \mathbb{T}_{\text{second}})$ 
13:      得出  $\mathbb{D}_{\text{unmatch}}$ 、 $\mathbb{T}_{\text{unmatch}}$  与  $\mathbb{T}_{\text{second-match}}$ 
14:       $\mathbb{T}_{\text{match}} \leftarrow \mathbb{T}_{\text{first-match}} \cup \mathbb{T}_{\text{second-match}}$ 
15:      利用 HAEMA 更新  $\mathbf{e}_j^t$ 
16:       $\mathbb{T}_{\text{new}} \leftarrow \text{创建}(\mathbb{D}_{\text{unmatch}}, \varphi, (\mathbf{L}_j^k, \mathbf{L}_j^k, \mathbf{P}_j^k, \kappa_j^{k+1}, l_j^k))$ 
17:       $\mathbb{T}_{\text{delete}} \leftarrow \text{删除}(\mathbb{T}_{\text{unmatch}})$ 
18:      更新轨迹置信度  $\vartheta$ 
19:      利用 PAKF 更新  $\mathbb{T}_{\text{match}}$  &  $\mathbb{T}_{\text{new}}$  的  $\mathbf{L}_j^t$ 
20:       $\mathbb{T}_t \leftarrow \mathbb{T}_{\text{match}} \cup \mathbb{T}_{\text{unmatch}} \cup \mathbb{T}_{\text{new}}$ 
21:       $\mathbb{T}_t \leftarrow \mathbb{T}_t \setminus \mathbb{T}_{\text{delete}}$ 
22:   end
23:   返回:  $\mathbb{T}_{\text{finish}}$ 

```

5.2 实验验证

5.2.1 实验设置

实验过程中，所有算法均采用检测器 YOLOX-X 为检测器，详情见第 2.4 节。Deep OC-SORT 与 BOT-SORT 所采用外观特征提取模型均其原论文所述模型。所有对照组算法超参数设置也与原论文中保持一致，所有算法均未使用后处理。所提综合应用方案采用的 PAKF、Fast OSNet 与 HAEMA 参数与权重设置均与第 2、3 章保持一致，CWTracker 框架参数设置见第 4.4 节。

实验环境如下：CPU 型号为 Intel(R) Core(TM) i9-12900K @ 5.0GHz，内存大小为 128GB，显卡型号为 Nvidia RTX3090。所有程序均通过 Python 3.8 和 PyTorch 1.10.1 实现。实际场景视频由 Redmi Note12 Turbo 手机拍摄，视频分辨率为 2000×1080 ，帧率为 30FPS，未使用图像增强。

5.2.2 消融实验

为量化比较 PAKF、Fast OSNet 与 CWTracker 对综合应用方案跟踪性能的贡献程度，在 MOT17 与 MOT20 验证集上开展消融实验，结果分别如表 5-1 与表 5-2 所示。

表 5-1 综合应用方案在 MOT17 半验证集上的消融实验结果(最优数据用粗体标出)

方案	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
基线(CWTracker)	68.0	78.0	79.5	69.4	62.2	10.9	311	624.1
+ PAKF	68.2	78.1	79.8	69.9	62.7	11.4	298	611.3
综合应用方案	69.0	78.4	81.5	72.2	63.5	10.8	179	81.5

表 5-2 综合应用方案在 MOT20 半验证集上的消融实验结果(最优数据用粗体标出)

方案	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
基线(CWTracker)	59.6	74.4	76.1	57.9	56.8	9.9	1317	144.2
+ PAKF	59.7	74.3	76.5	58.3	56.8	9.8	1425	138.6
综合应用方案	59.9	74.4	76.7	58.4	56.8	9.9	1215	15.9

由表 5-1 可知，在加入 PAKF 后，HOTA、MOTA、IDF1 和 AssA 分别提升 0.2%、0.1%、0.3%和 0.5%，IDs 下降 13，关联速度下降 12.8FPS。由表 5-2 可知，在加入 PAKF 后，即使面对 MOT20 数据集的高遮挡场景，HOTA、MOTA、IDF1 和 AssA 同

样取得了一定程度的提升。因此，所提 PAKF 对于提升 CWTracker 在高遮挡场景下的跟踪性能而言是有益的，且造成的速度下降较小。

由表 5-1 可知，加入 Fast OSNet 后，HOTA、MOTA、IDF1 和 AssA 分别提升 0.8%、0.3%、1.7%和 0.3%，IDs 下降 119，关联速度降至 81.5FPS。在面对高遮挡的 MOT20 数据集时，HOTA、MOTA、IDF1 和 AssA 分别提升 0.2%、0.1%、0.2%和 0.1%，效果不如面对 MOT17 数据集时显著。这是由于 MOT20 数据集为固定视角拍摄，虽目标间相互遮挡严重，但不存在因拍摄视角移动造成的运动预测失真。同时，大量的夜晚场景，使得许多目标缺失纹理细节，拉近了各目标外观嵌入间的余弦距离。因此，所提 Fast OSNet 能够有效提升 CWTracker 的跟踪性能，但会造成关联速度大幅下降，且面对暗光、固定视角场景时，其性能提升不如 PAKF 明显。

5.2.3 对比实验

以近期 SOTA 方法为对照组，在相同设备环境下，与所提综合应用方案开展对比实验。MOT17 与 MOT20 验证集上的运行结果分别见表 5-3 与表 5-4。为公平比较所涉及方法们的在线 MOT 性能，所有方法均未采用后处理步骤。

表 5-3 所提方法与对照组方法在 MOT17 验证集上结果(最优数据用粗体标出)

方法	App	CMC	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
OC-SORT ^[28]			66.1	74.5	77.6	68.6	48.7	16.5	233	498.7
ByteTrack ^[27]			67.7	77.7	79.3	69.5	58.1	12.7	213	612.3
Deep OC-SORT ^[30]	√	√	68.3	75.1	81.0	72.4	50.2	15.3	138	14.2
BOT-SORT ^[11]	√	√	69.1	78.5	81.9	74.4	60.5	11.2	162	17.0
综合应用方案	√		69.0	78.4	81.5	72.2	63.5	10.8	179	81.5

表 5-4 所提方法与对照组方法在 MOT20 验证集上结果(最优数据用粗体标出)

方法	App	CMC	HOTA↑	MOTA↑	IDF1↑	AssA↑	MT↑	ML↓	IDs↓	FPS↑
OC-SORT ^[28]			56.4	69.9	73.4	55.2	44.1	11.6	1275	78.8
ByteTrack ^[27]			57.4	72.7	72.7	55.1	52.5	11.1	1225	143.7
Deep OC-SORT ^[30]	√	√	57.4	69.9	74.7	56.9	44.2	11.7	1125	3.3
BOT-SORT ^[11]	√	√	58.5	74.1	74.1	56.1	57.2	10.2	1632	3.8
综合应用方案	√		59.9	74.4	76.7	58.4	56.8	9.9	1215	15.9

由表 5-3 可知，在 MOT17 验证集上，BOT-SORT 跟踪性能最优，所提方案次之，

二者跟踪性能接近，所提方案在 HOTA、MOTA、IDF1 与 AssA 上分别落后 0.1%、0.1%、0.4%和 0.2%，但其关联速度是 BOT-SORT 的 4.8 倍。这得益于所提方案未使用复杂且耗时的 CMC 前处理过程和 BOT 外观模型，而是采用自适应 PAKF 和轻量级 Fast OSNet，简化了运行步骤，降低了算力需求，这使其更能适应边缘部署场景。此外，较之其他 SOTA 方法，所提方案在跟踪性能方面优势更为明显。例如，较之使用了外观模型与 CMC 的 Deep OC-SORT，所提方案在 HOTA、MOTA 和 IDF1 上分别领先 0.7%、3.3%和 0.5%，且速度为前者的 5.7 倍。

另一方面，由表 5-4 可知，在更为拥挤、遮挡频繁的 MOT20 验证集上，所提方案取得了最优的跟踪性能，较次优方法 BOT-SORT 提升 0.4% HOTA、0.3% MOTA、2.6% IDF1 和 2.3% AssA，速度为其 4.2 倍，满足实时需求。由第 4.4.4 节分析可知，这得益于所提方案中的 CWTracker 关联框架采用置信度优化的设计，更能应对高遮挡场景下目标置信度与遮挡程度的随机变化。

5.2.4 定性分析

(1) 公用数据集场景

图 5-2、5-3 分别展示了 ByteTrack 与所提方法在 MOT20-03 与 MOT17-02 视频序列上的可视化跟踪结果。图 5-2 所示视频为监控摄像头拍摄的夜晚极端拥挤场景。图 5-3 所示视频为手持固定拍摄视角，低光照条件，存在大量小目标与频繁的完全遮挡。

如图 5-2 中红色三角形所示，在#1330 帧，131 号目标开始与 117 号小目标交汇。由于 117 号目标处于暗光场景下，纹理特征不明显且部分遮挡，其置信度 $s = 0.14$ ，属于低置信度目标，并且周围存在多个位置相近的同类目标。在#1340 帧，131 号目标几乎完全遮挡了 117 号目标，此时 ByteTrack 中断了对 117 号目标的跟踪，而所提出的方法均成功识别并继续跟踪了该目标。在#1350 帧遮挡结束后，ByteTrack 中的原 117 号目标转变为 131 号目标，而原来的 131 号目标则成为新的目标，而所提出的方案都保持了目标的原始身份。

类似的情况也出现在图 5-3 中红色三角形处，即在#340 帧，所有四种方案均成功识别并跟踪了 5 号目标小目标。然而，在#350 帧，5 号目标被 27 号目标完全遮挡，随后在#360 帧遮挡结束后，ByteTrack 方法的跟踪结果中，原先的 5 号目标与原先的 26

号目标发生了身份切换，而所提出的方案都未出现此情况。

综上所述，所提出的方法在面对高遮挡场景时表现出更强的跟踪鲁棒性，这得益于 CWTracker 未对高、低分检测采用硬性划分，使得整体方案能更好地保持低分目标的身份。另一方面，在陆续引入 PAKF 和 Fast OSNet 后，所提出的方案的跟踪结果并未出现显著变化。这是由于这两段视频均为固定视角拍摄，PAKF 未能根据相机坐标系的变化自适应调整观测和过程噪声，且两段视频均为暗光场景，目标的纹理特征不明显，外观模型作用不明显。

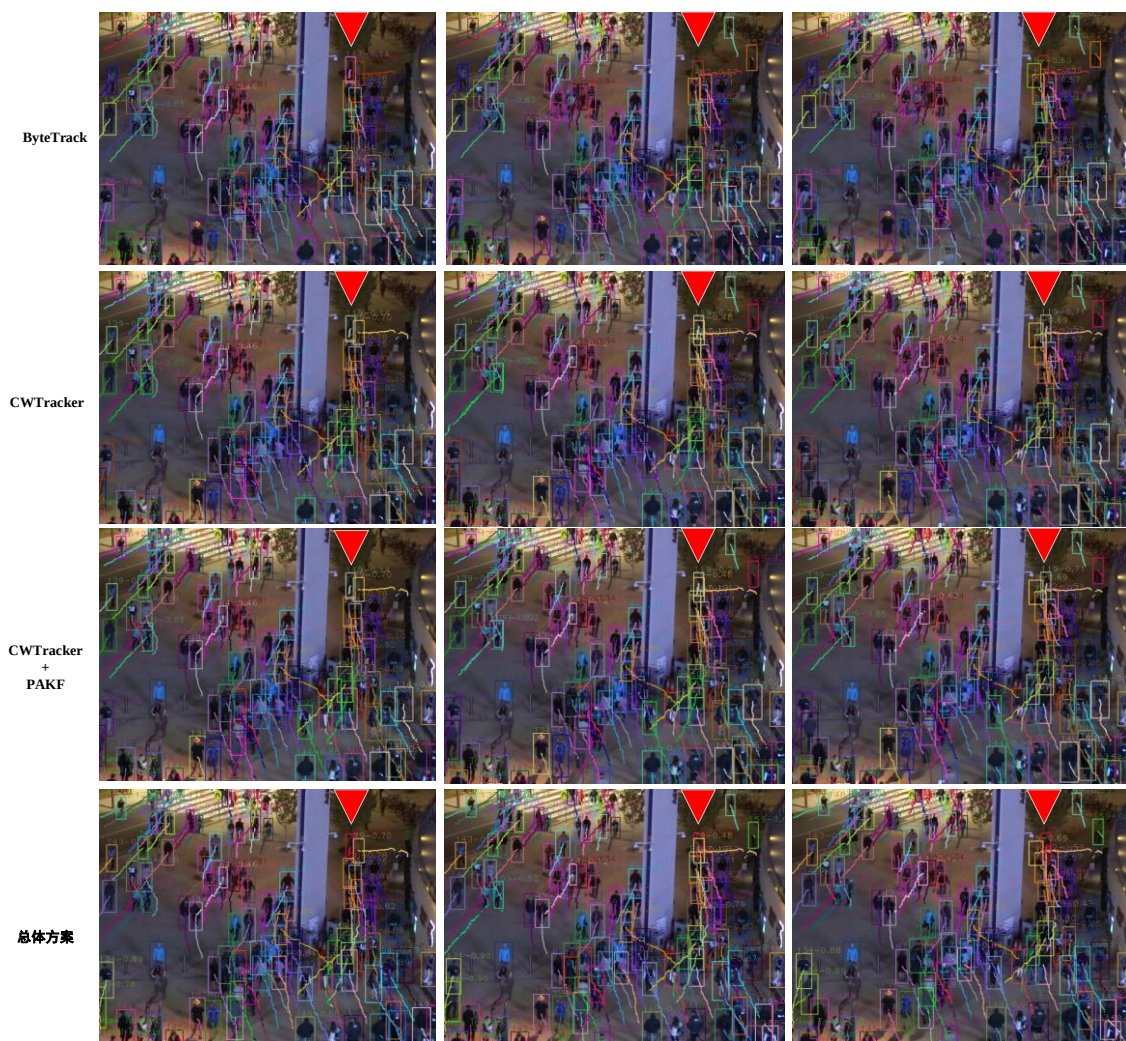


图 5-2 MOT20-03 序列跟踪结果(#1330、#1340、#1350)



图 5-3 MOT17-02 序列跟踪结果(#340、#350、#360)

(2) 实际场景

图 5-4 和 5-5 分别展示了 ByteTrack 与所提方法在面对存在高遮挡、动态实际场景时的跟踪结果，2 段视频均拍摄于校园内。图 5-4 所示视频为室内手持移动拍摄。图 5-5 所示视频为室外手持移动拍摄，光照条件充足，存在大量小目标、频繁的类型间、类内遮挡与非线性运动目标。

如图 5-4 中红色三角形所示的被遮挡目标群，在#20 帧时，ByteTrack 漏检了高度遮挡的 8 号目标，而所提出的方案成功检测并跟踪了该目标。在#30 帧时，4 号目标完全遮挡了 2 号和 5 号目标，同时 14 号目标的部分肢体出现在图像右侧。ByteTrack 仍未检测到该目标，而所提方案则均成功检出并跟踪。在#40 帧时，遮挡结束时，ByteTrack 中原来的 2 号目标变成了 16 号目标，跟踪中断，而所提方案则成功保持了轨迹一致性。这得益于所提 PAKF，它缓解了由于相机坐标系变化而导致的在图像坐标系中的非线性运动。

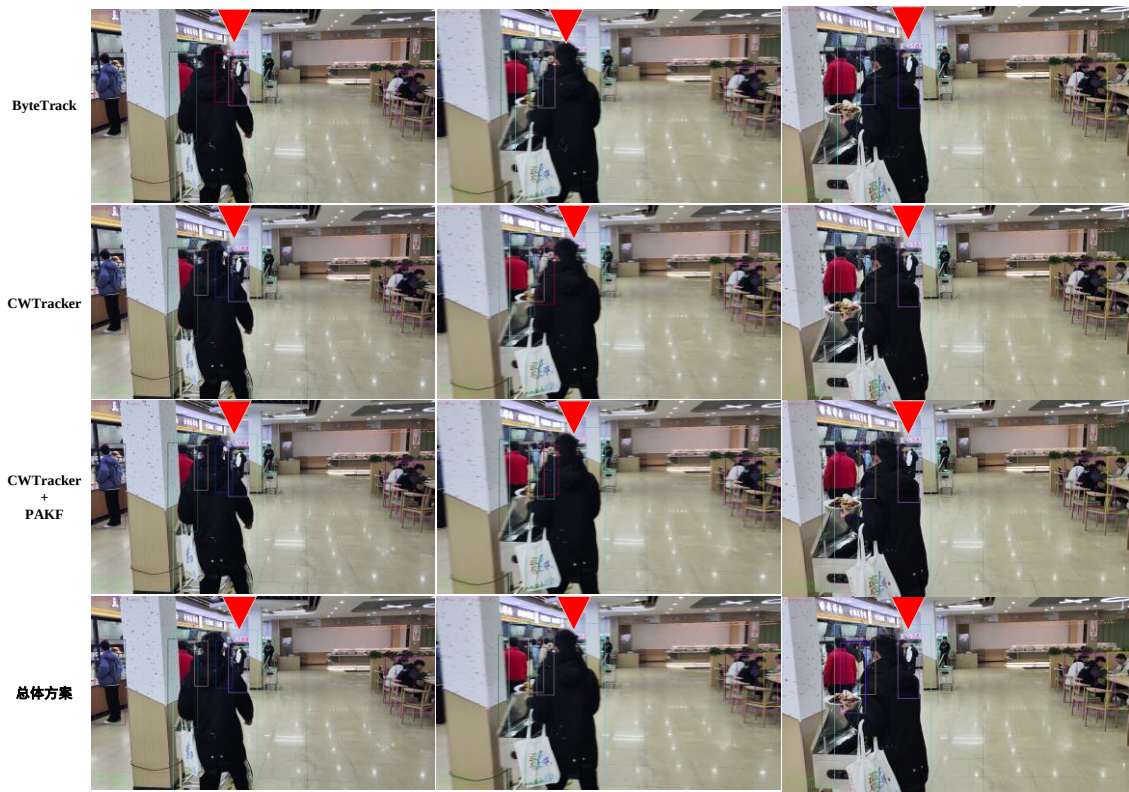


图 5-4 室内高遮挡场景跟踪结果

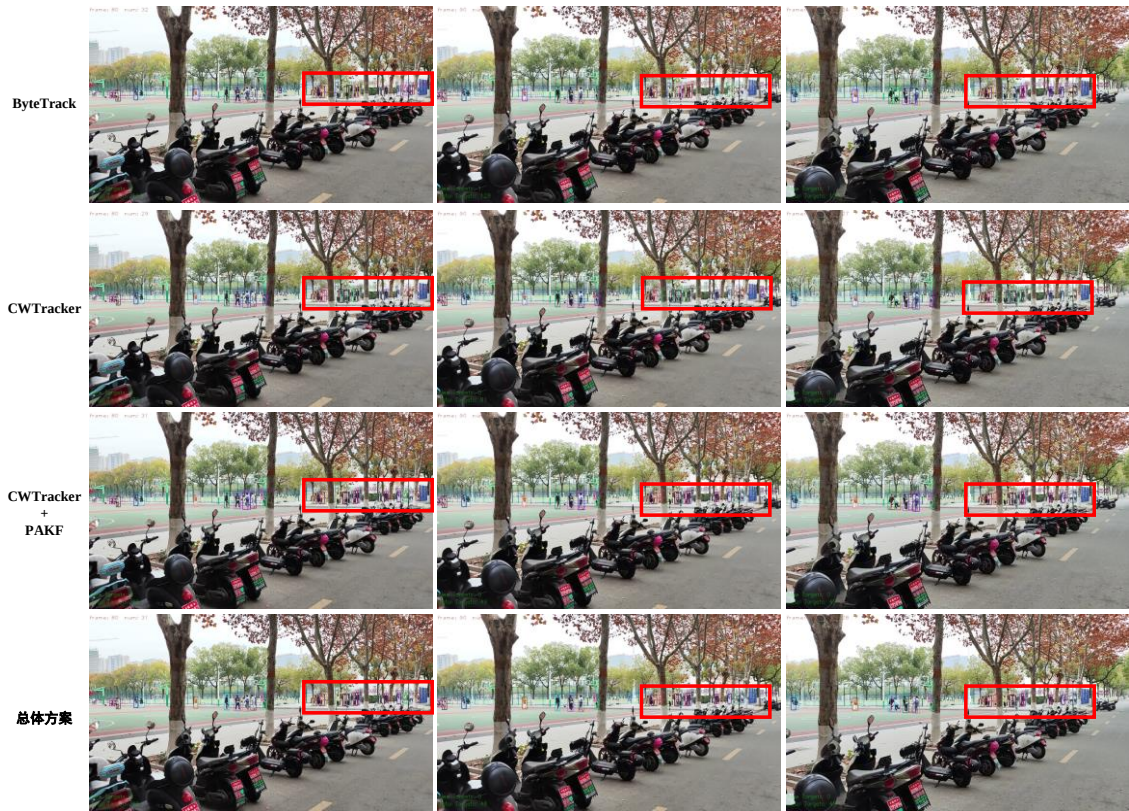


图 5-5 室外高遮挡场景跟踪结果

类似的情况也出现在图 5-5 中红色框所示的小目标集群中。在这个区域中存在多个远处的相互遮挡的小目标以及树干、电动车等障碍物。随着拍摄视角的移动和障碍物的遮挡，在#80 到#100 帧之间，ByteTrack 的跟踪结果出现了大量身份交换，CWTracker 也出现了一些身份交换，但数量较少。而引入了 PAKF 的方案，只出现了少量交换，整体方案则没有发生交换。这表明较之 ByteTrack，所提方案更能适应存在高遮挡小目标和视角运动的复杂场景。

此外，与 CWTracker 相比，引入 PAKF 和 Fast OSNet 能够提升综合应用方案在高遮挡动态场景下的跟踪鲁棒性。

5.3 本章小结

本章结合前文提出的 PAKF、轻量级深度外观特征提取方案和 CWTracker，组成了所提综合应用方案，在 MOT17 和 MOT20 验证集上开展了消融实验和对比实验，并在数据集场景和实际拍摄场景上开展了可视化的定性分析。实验结果表明：所提方案在 MOT17 验证集上取得了次优性能，与最优方法相比在 HOTA、MOTA、IDF1 与 AssA 上分别落后 0.1%、0.1%、0.4%和 0.2%，但其关联速度是后者的 4.8 倍。而在高遮挡的 MOT20 验证集上则取得了最优性能，较次优方法提升了 0.4% HOTA、0.3% MOTA、2.6% IDF1 和 2.3% AssA，且速度为其 4.2 倍，满足实时需求。此外，消融实验与定性分析的结果均表明，引入 PAKF 和 Fast OSNet 能提升 CWTracker 的跟踪准确度与鲁棒性。因此，所提综合应用方案能有效应对动态高遮挡场景下的 MOT 任务，且权衡了跟踪性能与时效性，满足了时间关键型任务的实时需求。

第 6 章 结论与展望

6.1 结论

针对传统 TBD 类 MOT 方法在高遮挡、动态场景下的跟踪局限性，从运动预测、外观特征提取和数据关联入手。基于噪声尺度自适应策略，以运动匹配成本和置信度分数分别作为过程和观测噪声的尺度参数，并通过幂次变换构建非线性自适应因子，对不同轨迹的过程和观测协方差矩阵进行差异化调整，提出 PAKF。根据 MOT 任务实际需求，简化 OSBR 块结构，添加空间与信道注意力模块，引入多尺度特征融合机制，提出 Fast OSNet，还结合表征与度量学习，为其设计联合损失函数。此外，结合基于 EMA 和二段线性变化构建的自适应轨迹外观状态更新方法 HAEMA，得出所提用于 MOT 任务的轻量级深度外观特征提取方案。基于 EMA 建模轨迹置信度，并结合检测置信度构成置信度优化权重，为代价矩阵引入置信度线索，基于此，构建一个两阶段的全检测数据关联算法 CWTracker。最终，结合所提 PAKF、外观特征提取方案与 CWTracker 得出用于高遮挡、动态环境下的综合应用方案。

本文的主要结论如下：

(1) 在 MOT17 验证集上的实验结果表明，引入 PAKF 后 ByteTrack 的各项性能均获得更显著的提升，且额外计算导致的速度损失是可接受的。而在更为复杂的 HiEve 测试集上，PA-ByteTrack 的 MOTA、MOTP 和 MT 分别领先原版 0.5%、0.3%和 0.5%。因此，所提 PAKF 通过引入对过程与观测噪声尺度的自适应，可提升了 TBD 类方法的高遮挡、动态场景下的运动预测准确性，能有效应对非线性运动，且不会造成严重的速度损失。

(2) 在 Market1501 数据集上的消融实验显示，较其基线 OSNet，所提 Fast OSNet 的 Rank-1 和 mAP 分别提升了 1.4 和 3.7，推理速度增加 69.0%，内存占用减少 19.4%。说明 Fast OSNet 在提升推理速度的同时，进一步提升了重识别性能，这得益于多尺度特征的融合与注意力机制的作用。在 MOT17 验证集上的消融实验结果显示，HAEMA 能够有效降低遮挡噪声对轨迹外观状态的污染，从而提升跟踪准确度和鲁棒性。此外，

使用所提轻量级深度外观特征提取方案的 MOT 方法 FDBTrack，其关联速度是基线方法 DeepSORT 的 1.6 倍，且跟踪准确度和鲁棒性提升显著；与基于运动的方法 ByteTrack 相比，其虽跟踪速度较慢，但在高遮挡、动态场景下的跟踪效果更为鲁棒。另一方面，其在 MOT17 测试集上实现了 63.2% 的 HOTA% 和 77.7% 的 IDF1；在 MOT20 测试集上则为 62.0% 和 75.9%，较近期的 JDE 类方法更有优势。

(3) 与 SOTA 方法在相同设备上的对比实验显示，CWTracker 在保持高关联速度的同时，改善了跟踪准确性和关联鲁棒性，尤其是在高遮挡的复杂环境中。在 MOT17 测试集上，其实现了 63.0% 的 HOTA 和 77.1% 的 HOTA，且满足实时需求。在更为复杂的 MOT20 测试集上则为 62.4% HOTA 和 76.1% HOTA，优于类似的方法 ByteTrack 和 OC-SORT，且得益于简化的关联阶段，其在跟踪速度更快的同时，避免了繁琐的阈值优化过程，实现了真正的全检测关联，这使其在实际应用中更具吸引力。

(4) 在 MOT17 和 MOT20 数据集上的消融实验表明，引入 PAKF 和 Fast OSNet 能提升 CWTracker 的跟踪准确度和鲁棒性。与 SOTA 方法的对比实验显示，所提综合应用方案在 MOT17 验证集上取得了次优性能，与最优方法性能差距较小，且关联速度是后者的 4.8 倍。而在高遮挡的 MOT20 验证集上所提综合应用方案则取得了最优性能，较次优方法提升了 0.4% HOTA、0.3% MOTA、2.6% IDF1 和 2.3% AssA，且速度为其 4.2 倍。因此，所提综合应用方案能有效应对动态高遮挡场景下的 MOT 任务，且权衡了跟踪性能与时效性，满足了时间关键型任务的实时需求，能够适用于有算力和内存限制的边缘部署场景。

6.2 创新点

本文的创新点如下：

(1) 多尺度特征融合轻量级深度外观特征提取方案

通过剖析 Re-ID 与 MOT 任务差异，轻量化改进 OSNet 网络。通过简化其 OSRB 结构，引入信道和空间注意力机制，并添加额外的局部特征输出分支，提出了 Fast OSNet 网络。还结合表征与度量学习，联合多 ID 损失和圆损失为其设计联合损失函数。此外，为缓解遮挡噪声对轨迹外观特征的污染，基于 EMA 模型，引入轨迹外观置信度，结合二段线性变换，设计自适应外观更新权重，提出了低代价、鲁棒、自适应的

轨迹外观更新策略 HAEMA。二者组合构成了所提轻量级深度外观提取方案，其能以较低的代价，提升 MOT 算法的在高遮挡、动态场景下的跟踪准确度和鲁棒性。

(2) 基于置信度优化的全检测关联跟踪器

通过摒弃依赖置信度门限硬性划分匹配成本矩阵的传统思路，创新地采用 EMA 模型迭代更新轨迹置信度，进而与检测置信度结合，构建加权成本矩阵，在匹配过程引入额外的轨迹与检测置信度信息，减少模糊的匹配关系，提升关联准确率，实现基于置信度优化的轨迹与检测关联。以此为基础，结合轨迹管理策略，提出了一种二阶段的全检测关联跟踪器，CWTracker，其能直接处理所有检测器输出，实现了全检测关联，避免了检测集合划分步骤。较之依赖门限阈值的方法，CWTracker 无需调整置信度阈值即可面对多种复杂场景，在高遮挡动态场景中更具优势，且跟踪速度更快，在无人驾驶、移动端拍摄应用上具备潜在价值。

6.3 未来展望

本文研究了基于幂次-自适应卡尔曼滤波的运动预测，提出了融合轻量级深度外观特征提取方案和基于置信度优化全检测关联算法，并结合以上三者，提出了综合应用方案。由于时间关系，还存在以下问题值得进一步研究：

(1) JDE 类方法在实际应用中展现了时效性上的巨大优势，然重识别与目标检测对特征需求的矛盾关系，使得其训练过程受损。在面对高遮挡场景时，难以取得令人满意的跟踪效果。基于本文第 3 章的研究，在未来的工作中，如何将深度外观特征提取网络与目标检测器有机结合，并设计合理的联合损失函数将成为一个有价值的研究课题。这将有助于从根本上破解 MOT 任务跟踪性能与时效性间的矛盾关系。

(2) MOT 任务作为许多智能环境感知任务的前端，如何与后续任务，例如同步定位与建图、目标位姿估计与运动预测、视频理解与分析等有机结合，并实现移动端部署，将成为未来的工作重点。

参考文献

- [1] 葛宝义,左宪章,胡永江.视觉目标跟踪方法研究综述[J].中国图象图形学报,2018,23(08):1091-1107.
- [2] 黄君君.基于改进 DLT 算法的单目标识别跟踪研究[J].太原学院学报(自然科学版),2023,41(04):52-58.
- [3] 杨洋,宋品德,钟春来等.无人机视角下基于深度学习的多目标跟踪研究进展[J].计算机工程与应用,2023,59(23):48-62.
- [4] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric[C]//2017 IEEE international conference on image processing (ICIP). IEEE, 2017: 3645-3649.
- [5] Kishi N, Shinkuma R, Oka M, et al. Multi-object tracking for road surveillance without using features of image data[C]//2021 IEEE Global Communications Conference (GLOBECOM). IEEE, 2021: 1-6.
- [6] Chen J, Wang F, Li C, et al. Online multiple object tracking using a novel discriminative module for autonomous driving[J]. Electronics, 2021, 10(20): 2479-2489.
- [7] Akkaladevi S C, Plasch M, Eitzinger C, et al. Towards a context enhanced framework for multi object tracking in human robot collaboration[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 168-173.
- [8] Yurtsever M M E, Eken S. BabyPose: real-time decoding of baby's non-verbal communication using 2D video-based pose estimation[J]. IEEE Sensors Journal, 2022, 22(14): 13776-13784.
- [9] Qin Z, Zhou S, Wang L, et al. MotionTrack: Learning Robust Short-term and Long-term Motions for Multi-Object Tracking [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 17939-17948.
- [10] Han S, Huang P, Wang H, et al. Mat: Motion-aware multi-object tracking[J].

- Neurocomputing, 2022, 476: 75-86.
- [11] Aharon N, Orfaig R, Bobrovsky B Z. BoT-SORT: Robust associations multi-pedestrian tracking[J/OL]. arXiv preprint arXiv:2206.14651, 2022.
- [12] Wen L, Du D, Cai Z, et al. DETRAC: A new benchmark and protocol for multi-object tracking[J/OL]. arXiv preprint arXiv:1511.04136, 2015.
- [13] Bewley A, Ge Z, Ott L, et al. Simple online and realtime tracking[C]//2016 IEEE international conference on image processing (ICIP). IEEE, 2016: 3464-3468.
- [14] Zhang Y, Wang C, Wang X, et al. Fairmot: On the fairness of detection and re-identification in multiple object tracking[J]. International Journal of Computer Vision, 2021, 129: 3069-3087.
- [15] Bae S H, Yoon K J. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(3): 595-610.
- [16] Mahmoudi N, Ahadi S M, Rahmati M. Multi-target tracking using CNN-based features: CNNMTT[J]. Multimedia Tools and Applications, 2019, 78: 7077-7096.
- [17] Ge Z, Liu S, Wang F, et al. YoloX: Exceeding yolo series in 2021[J/OL]. arXiv preprint arXiv: 2107.08430, 2021.
- [18] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
- [19] Kalman R E. A new approach to linear filtering and prediction problems[J]. J Fluids Eng, 1960, 82(1) : 35-45.
- [20] Hamuda E, Mc Ginley B, Glavin M, et al. Improved image processing-based crop detection using Kalman filtering and the Hungarian algorithm[J]. Computers and electronics in agriculture, 2018, 148: 37-44.
- [21] Feichtenhofer C, Pinz A, Zisserman A. Detect to track and track to detect[C]//Proceedings of the IEEE international conference on computer vision. 2017:

3038-3046.

- [22] Bergmann P, Meinhardt T, Leal-Taixe L. Tracking without bells and whistles[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 941-951.
- [23] Zhou K, Yang Y, Cavallaro A, et al. Omni-scale feature learning for person re-identification[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 3702-3712.
- [24] Wang Z, Zheng L, Liu Y, et al. Towards real-time multi-object tracking[C]//European Conference on Computer Vision. Cham: Springer International Publishing, 2020: 107-122.
- [25] Liang C, Zhang Z, Zhou X, et al. Rethinking the competition between detection and reid in multiobject tracking[J]. IEEE Transactions on Image Processing, 2022, 31: 3182-3196.
- [26] Yu E, Li Z, Han S, et al. Relationtrack: Relation-aware multiple object tracking with decoupled representation[J]. IEEE Transactions on Multimedia, 2023, 25: 2686-2697.
- [27] Zhang Y, Sun P, Jiang Y, et al. Bytetrack: Multi-object tracking by associating every detection box[C]//European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022: 1-21.
- [28] Cao J, Pang J, Weng X, et al. Observation-centric sort: Rethinking sort for robust multi-object tracking[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 9686-9696.
- [29] Luo H, Gu Y, Liao X, et al. Bag of tricks and a strong baseline for deep person re-identification[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2019: 0-0.
- [30] Maggolino G, Ahmad A, Cao J, et al. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification[J/OL]. arXiv preprint arXiv: 2302.11813, 2023.
- [31] Gómez-Silva M J, de la Escalera A, Armingol J M. Deep learning of appearance affinity for multi-object tracking and re-identification: A comparative view[J]. Electronics, 2020,

9(11): 1757.

- [32] Hadsell R, Chopra S, LeCun Y. Dimensionality reduction by learning an invariant mapping[C]//2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06). IEEE, 2006, 2: 1735-1742.
- [33] Sun Y, Zheng L, Yang Y, et al. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 480-496.
- [34] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 815-823.
- [35] Sun Y, Cheng C, Zhang Y, et al. Circle loss: A unified perspective of pair similarity optimization[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 6398-6407.
- [36] Zhang X, Luo H, Fan X, et al. Alignedreid: Surpassing human-level performance in person re-identification[J/OL]. arXiv preprint arXiv: 1711.08184, 2017.
- [37] Zhang Z, Si T, Liu S. Integration convolutional neural network for person re-identification in camera networks[J]. IEEE Access, 2018, 6: 36887-36896.
- [38] He M, Luo H, Hui B, et al. Fast online multi-pedestrian tracking via integrating motion model and deep appearance model[J]. IEEE Access, 2019, 7: 89475-89486.
- [39] Sheng H, Zhang Y, Chen J, et al. Heterogeneous association graph fusion for target association in multiple object tracking[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 29(11): 3269-3280.
- [40] Du Y, Wan J, Zhao Y, et al. Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021[C]//Proceedings of the IEEE/CVF International conference on computer vision. 2021: 2809-2819.
- [41] Du Y, Zhao Z, Song Y, et al. Strongsort: Make deepsort great again[J]. IEEE Transactions on Multimedia, 2023, 25: 8725-8737.

- [42] Zhang H, Wu C, Zhang Z, et al. Resnest: Split-attention networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 2736-2746.
- [43] Sener O, Koltun V. Multi-task learning as multi-objective optimization[J]. Advances in neural information processing systems, 2018, 31.
- [44] Zhang J, Xing M, Sun G C, et al. Multiple Statistics Contributing to Few-Sample Deep Learning for Subtle Trace Detection in High-Resolution SAR Images[J]. IEEE Transactions on Geoscience Remote Sensing, 2021, 60: 1-14.
- [45] Emami P, Pardalos P M, Elefteriadou L, et al. Machine learning methods for data association in multi-object tracking[J]. ACM Computing Surveys (CSUR), 2020, 53(4): 1-34.
- [46] Svensson L, Svensson D, Guerriero M, et al. Set JPDA filter for multitarget tracking[J]. IEEE Transactions on Signal Processing, 2011, 59(10): 4677-4691.
- [47] Sheng H, Chen J, Zhang Y, et al. Iterative multiple hypothesis tracking with tracklet-level association[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 29(12): 3660-3672.
- [48] Sun P, Cao J, Jiang Y, et al. Transtrack: Multiple object tracking with transformer[J/OL]. arXiv preprint arXiv: 2012.15460, 2020.
- [49] Yoon K, Kim D Y, Yoon Y C, et al. Data association for multi-object tracking via deep neural networks[J]. Sensors, 2019, 19(3): 559.
- [50] 齐美彬,岳周龙,疏坤等.基于广义关联聚类图的分层关联多目标跟踪[J].自动化学报,2017,43(01):152-160.
- [51] Chen L, Ai H, Zhuang Z, et al. Real-time multiple people tracking with deeply learned candidate selection and person re-identification[C]//2018 IEEE international conference on multimedia and expo (ICME). IEEE, 2018: 1-6.
- [52] Stadler D, Beyerer J. Modelling ambiguous assignments for multi-person tracking in crowds[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022: 133-142.

- [53] Hua W, Mu D, Zheng Z, et al. Online multi-person tracking assist by high-performance detection[J]. The Journal of Supercomputing, 2020, 76: 4076-4094.
- [54] Hou X, Wang Y, Chau L P. Vehicle tracking using deep sort with low confidence track filtering[C]//2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2019: 1-6.
- [55] Kim T, Park T H. Extended Kalman filter (EKF) design for vehicle position tracking using reliability function of radar and lidar[J]. Sensors, 2020, 20(15) : 4126.
- [56] Wang M, Wu X. Multi-Object Tracking Strategy of Autonomous Vehicle Using Modified Unscented Kalman Filter and Reference Point Switching[J]. Journal of Shanghai Jiaotong University (Science), 2021, 26(5) : 607-14.
- [57] Cui Y, Zhang J, He Z, et al. Multiple pedestrian tracking by combining particle filter and network flow model[J]. Neurocomputing, 2019, 351: 217-27.
- [58] Zhai G, Wu C, Wang Y. Millimeter wave radar target tracking based on adaptive Kalman filter[C]//2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 453-458.
- [59] Wang Y, Mu X. Dynamic siamese network with adaptive Kalman filter for object tracking in complex scenes[J]. IEEE Access, 2020, 8: 222918-30.
- [60] He J, Erfani S, Ma X, et al. Alpha-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression[J]. Advances in Neural Information Processing Systems, 2021, 34: 20230-42.
- [61] Sarkar K, Halder T K, Mandal A. Adaptive power-law and cdf based geometric transformation for low contrast image enhancement[J]. Multimedia Tools Applications, 2021, 80(4) : 6329-53.
- [62] Milan A, Leal-Taixe L, Reid I, et al. MOT16: A Benchmark for Multi-Object Tracking[J/OL]. arXiv preprint arXiv: 1603.00831, 2016.
- [63] Lin W, Liu H, Liu S, et al. Human in Events: A Large-Scale Benchmark for Human-centric Video Analysis in Complex Events[J/OL]. arXiv preprint arXiv: 2005.04490, 2020.

- [64] Kumar M, Singh K. Retrieval of head–neck medical images using Gabor filter based on power-law transformation method and rank BHMT[J]. *Signal, Image Video Processing*, 2018, 12(5) : 827-33.
- [65] Dendorfer P, Rezatofighi H, Milan A, et al. MOT20: A benchmark for multi object tracking in crowded scenes[J/OL]. *arXiv preprint arXiv: 2003.09003*, 2020.
- [66] Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: the clear mot metrics[J]. *EURASIP Journal on Image Video Processing*, 2008, 2008: 1-10.
- [67] Ristani E, Solera F, Zou R, et al. Performance measures and a data set for multi-target, multi-camera tracking[C]//*European conference on computer vision*. Cham: Springer International Publishing, 2016: 17-35.
- [68] Luiten J, Os Ep A A, Dendorfer P, et al. HOTA: A Higher Order Metric for Evaluating Multi-object Tracking[J]. *Int J Comput Vis*, 2021, 129(2) : 548-78.
- [69] Wu B, Nevatia R. Tracking of multiple, partially occluded humans based on static body part detection[C]//*2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, 2006: 951-8.
- [70] Shao S, Zhao Z, Li B, et al. Crowddhuman: A benchmark for detecting human in a crowd[J/OL]. *arXiv preprint arXiv: 1805.00123*, 2018.
- [71] Zhang S, Benenson R, Schiele B. Citypersons: A diverse dataset for pedestrian detection[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 3213-3221.
- [72] Ess A, Leibe B, Schindler K, et al. A mobile vision system for robust multi-person tracking[C]//*2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008: 1-8.
- [73] Shuai B, Berneshawi A, Wang M, et al. Application of multi-object tracking with Siamese track-RCNN to the human in events dataset[C]//*Proceedings of the 28th ACM International Conference on Multimedia*. 2020: 4625-4629.
- [74] Ericsson L, Gouk H, Loy C C, et al. Self-supervised representation learning: Introduction,

- advances, and challenges[J]. IEEE Signal Processing Magazine, 2022, 39(3): 42-62.
- [75] Kaya M, Bilge H Ş. Deep metric learning: A survey[J]. Symmetry, 2019, 11(9): 1066.
- [76] Xie B, Wu X, Zhang S, et al. Learning diverse features with part-level resolution for person re-identification[C]//Pattern Recognition and Computer Vision: Third Chinese Conference, PRCV 2020, Nanjing, China, October 16–18, 2020, Proceedings, Part III 3, Springer, 2020: 16-28.
- [77] Zheng L, Shen L, Tian L, et al. Scalable person re-identification: A benchmark[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1116-1124.
- [78] Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 116-131.
- [79] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.
- [80] Ding X, Zhang X, Han J, et al. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 11963-11975.
- [81] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.
- [82] Sun K H, Huh H, Tama B A, et al. Vision-based fault diagnostics using explainable deep learning with class activation maps[J]. IEEE Access, 2020, 8: 129169-129179.
- [83] Stadler D, Beyerer J. On the performance of crowd-specific detectors in multi-pedestrian tracking[C]//2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2021: 1-12.

攻读学位期间成果目录

(1) 论文发表情况

- [1] Liu Y, **Li Y**, Ma L, et al. Trajectory-Tracking Control for Manipulators Based on Fuzzy Equivalence and a Terminal Sliding Mode[J]. Strojniški vestnik-Journal of Mechanical Engineering, 2021, 67(9): 433-444. (SCI 收录)
- [2] Liu Y, **Li Y**, Zhang X, et al. Parametric design approach on high-order and multi-segment modified elliptical helical gears based on virtual gear shaping[J]. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2022, 236(9):4599-4609. (SCI 收录)
- [3] Liu Y, **Li Y**, Xu D, et al. Adaptive Kalman Filter with power transformation for online multi-object tracking[J]. Multimedia Systems, 2023, 29(3): 1231-1244. (SCI 收录, 对应第 2 章)
- [4] **Li Y**, Liu Y, Zhou C, et al. A lightweight scheme of deep appearance extraction for robust online multi-object tracking[J]. The Visual Computer, 2023: 1-17. (SCI 收录, 对应第 3 章)
- [5] Liu Y, **Li Y**, Zhang X, et al. A Full-Detection Association Tracker with Confidence-Optimization for Online Multi-Object Tracking[J]. Journal of Real-Time Image Processing. (SCI, 大修已返, 对应第 4 章)

(2) 专利申请情况

- [1] Liu Y, **Li Y**, Hongwei LI, Qijie WANG, Songsong LU. ONLINE MULTI OBJECT TRACKING METHOD COMBINED WITH THE LIGHTWEIGHT DEEP APPEARANCE EXTRACTION [P]. South Africa: 2023/08452, 2024-3-27.(国际专利, 对应第 3 章, 已授权)
- [2] **Li Y**, Youyu LIU, Guodong HU, Songsong LU, Hongwei LI. ULTRA-THIN GLASS

- FRICITION SEPARATION DEVICE AND ITS OPERATION METHOD [P]. South Africa: 2023/08247, 2024-3-27. (国际专利, 已授权)
- [3] Liu Y, **Li Y**, Guodong HU, Wanbao TAO, Hongwei LI. A GAS-LIQUID ISOLATION SYSTEM FOR A NEGATIVE PRESSURE SUCTION CUP IN A LIQUID ENVIRONMENT AND ITS OPERATIONAL METHOD [P]. South Africa: 2023/08203, 2024-3-27. (国际专利, 已授权)
- [4] 刘有余, **李艺**, 李红伟, 汪祁杰, 鹿松松. 一种结合轻量级深度外观提取的在线多目标跟踪方法: 中国, CN116385943A[P]. 2023-07-04. (发明专利, 实审中, 对应第 3 章)
- [5] **李艺**, 刘有余, 李红伟, 疏达, 汪祁杰. 基于置信度优化的在线多目标跟踪方法: 中国, CN116977374A[P]. 2023-10-31. (发明专利, 实审中, 对应第 4 章)
- [6] 刘有余, **李艺**, 胡国栋, 陶万宝, 李红伟. 一种用于液下负压吸盘的气液隔离系统及其工作方法: 中国, CN116142793A [P]. 2023-05-23. (发明专利, 实审中)
- [7] **李艺**, 刘有余, 胡国栋, 鹿松松, 李红伟. 一种超薄玻璃摩擦分片装置及其作业方法: 中国, CN116281174A[P]. 2023-06-23. (发明专利, 实审中)

(3)科研项目情况

- [1] “基于栅格-局部 PRM 方法的机械臂动态避障研究”项目, 安徽省教育厅研究生重点研究项目, 2020-2022, 项目编号: YJS20210446, 主持。
- [2] “多特征融合注意力检测在线多目标跟踪装置研究”项目, 安徽工程大学研究生创新创业项目, 2022-2023, 项目编号: YJS2023CX007, 主持。
- [3] “线激光视觉重构工件曲面系统设计”项目, 安徽省教育厅高校研究生创新项目, 2023-2024, 项目编号: 2023xscx089, 主持。

致谢

时光荏苒，岁月如梭。短短三年的求学生涯即将结束。在此，我要向所有关心、支持和帮助过我的人表示衷心的感谢。

首先，我要感谢我的导师刘有余教授。我能取得今天的研究成果离不开刘老师的悉心指导和耐心教诲，是您教会了我如何调查问题、明确方向、开展研究和撰写论文，您深厚的学识和丰富的经验使我受益匪浅，您的指导与鼓励使我在困难和挫折面前保持积极的态度，并不断追求卓越。

我还要感谢母校和实验室提供良好的实验设备与研究环境，感谢教育厅高校自然科学研究研究生项目“基于速度场的机械臂双目动态避障研究”(YJS20210446)和安徽工程大学研究生实践与创新项目“多特征融合注意力检测在线多目标跟踪装置研究”提供的资金支持，以及论文答辩组各位老师和审稿专家所提出的宝贵建议。

此外，我还要感谢各位同窗与师兄弟们在学习与生活中给与我的帮助。课题组陶万宝和鹿松松同学与我互相帮助、共同成长；师弟汪祁杰与方钊尽力辅助我完成设备调试与数据采集；邓湘琳、耿郑一、高众和柯启迪等同学在生活中给予我的很多帮助与理解。

最后，我要感谢我所有的家人和朋友们。感谢你们一直以来对我的理解、支持和鼓励。在我遇到困难和挫折时，你们的陪伴和鼓励给予了我巨大的力量和勇气。

再次向所有关心、支持和帮助过我的人表示衷心的感谢！祝各位在将来的人生中事业有成、一帆风顺！

尚德敏学 唯实惟新

