小 型 微 型 计 算 机 系 统
Journal of Chinese Computer Systems
DOI:10.20009/j.cnki.21-1106/TP.2025-0030
2026 年 2 月 第 2 期
Vol.47 No.2 2026

多相似度矩阵融合的多目标跟踪算法

陈 涛,肖 杰,张雨飞,周德龙

(浙江工业大学 计算机科学与技术学院,杭州 310023)
E-mail:846384435@qq.com

摘 要: 多目标跟踪是计算机视觉领域的核心研究方向,面临目标模糊、尺寸小和频繁遮挡等难题.提出一种基于深度学习和尺度不变特征转换(SIFT)的多目标跟踪算法,旨在解决目标模糊、尺寸小、遮挡等情况下跟踪精度不稳定的问题.通过引入SIFT特征改进DeepSORT算法,将DeepSORT算法中的重识别网络提取的描述目标间外观相似度的相似度矩阵与SIFT提取的目标间相似度矩阵融合.同时还采用与外观无关的连接模型与高斯平滑插值法对目标轨迹进行优化,提高了算法在复杂场景下的跟踪精度.采用MOT16数据集进行实验验证,结果表明,该跟踪算法在目标密集、遮挡等复杂情况下表现良好,多目标跟踪算法跟踪精度达到54.26%,IDF1分数达到61.24%,身份编号转换次数仅为406,相比DeepSORT算法具有较好的鲁棒性和精度.

关 键 词: 多目标跟踪;尺度不变特征转换;DeepSORT算法;特征融合;轨迹优化
中图分类号: TP391 **文献标识码:** A **文 章 编 号:** 1000-1220(2026)02-0428-07

Multi-object Tracking Algorithm Based on Multi-similarity Matrix Fusion

CHEN Tao, XIAO Jie, ZHANG Yufei, ZHOU Delong
(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: Multi-object tracking is a core research in computer vision, facing challenges such as blurred objects, small sizes, and frequent occlusions. A multi-object tracking algorithm based on deep learning and scale invariant feature transform (SIFT) is proposed, aiming to address unstable tracking accuracy in scenes with blurred objects, small sizes, and occlusions. Introducing SIFT features to improve DeepSORT, the similarity matrix of object appearance extracted by the re-identification network in DeepSORT is fused with the similarity matrix of object appearance extracted by SIFT. Additionally, a appearance-free link model and Gaussian-smoothed interpolation are employed to optimize object trajectories, enhancing the tracking accuracy of the algorithm in complex scenes. Experimental validation using the MOT16 dataset shows that the tracking algorithm performs well in complex scenes such as dense objects and occlusions, achieving the MOTA of 54.26%, the IDF1 of 61.24%, and only 406 identity switches, demonstrating bet-ter robustness and accuracy compared to the DeepSORT.

Keywords: muti-object tracking; scale invariant feature transformation; DeepSORT; feature fusion; trajectory optimization

0 引 言

在计算机视觉领域,多目标跟踪是一项重要的研究内容,其主要目标是将视频序列中感兴趣的目标检测出来,给每个目标分配不同的编号,在整个序列中形成目标的轨迹,实现对目标的跟踪,现已广泛应用在自动驾驶、智能监控、行为识别等领域.

随着深度学习在各种计算机视觉任务上的广泛应用,基于深度学习和检测的多目标跟踪算法可划分为基于检测的跟踪(Tracking-By-Detection,TBD)、联合检测跟踪(Joint Detection and Tracking,JDT)[1].基于检测的跟踪方法首先通过目标检测算法定位目标,然后利用外观、运动等信息进行关联.这种方法受益于目标检测技术的快速发展,长期以来在多目标跟踪任务中占据主导地位.后来,联合检测跟踪算法被提出,该算法联合训练检测模型和其他组件(如运动、嵌入和关联模型),这类追踪器的主要优势在于低计算成本和相当的性能[2,3].然而,联合追踪器面临两个主要问题:不同组件之间的竞争和共同训练组件的数据受限.这两个问题限制了追踪精度的上限.因此,基于检测的跟踪仍然是实现最佳追踪性能的解决方案.

基于检测的跟踪结合了目标检测和目标跟踪两个任务.这种方法的核心思想是在每一帧中首先使用目标检测算法来识别并定位目标对象,然后利用跟踪算法将这些目标对象在连续的帧之间进行关联,从而实现对目标的持续跟踪.在基于检测的跟踪中,目标检测算法负责在每一帧中准确地识别出目标对象的位置和范围.常见的目标检测算法包括一阶段和二阶段两种类型.一阶段算法如SSD算法[4]和YOLO系列算法[5,6],二阶段算法如R-CNN系列算法[7-9].检测算法在某一

收稿日期:2025-01-13 收修改稿日期:2025-02-25 基金项目:浙江省自然科学基金项目(LR24F020003,LZ22F020011)资助;国家自然科学基金项目(61972354)资助. 作者简介:**陈 涛**,男,1999年生,硕士研究生,研究方向为深度学习、多目标跟踪、计算机视觉;**肖 杰**,男,1984年生,博士,副教授,博士生导师,CCF会员,研究方向为AI系统可靠性与安全性、智能算法;**张雨飞**,男,1999年生,硕士研究生,研究方向为计算机视觉、深度学习;**周德龙**,男,1966年生,博士,副教授,研究方向为模式识别、计算机视觉.

帧中检测出感兴趣的目标,跟踪算法就会在前后帧计算各个目标之间的相似度,通过比较相似度来对目标的身份标识关联.

近年来,多目标跟踪领域涌现出了一些具有代表性的算法.例如在 2016 年,Bewley 等人提出的 SORT 算法[10]采用 Faster-RCNN 作为目标检测器得到检测框,结合卡尔曼滤波算法[11]得到预测框,随后计算检测框与预测框的交并比(Intersection over Union,IOU)作为每个目标之间的相似度,最后使用匈牙利算法[12]进行对每个目标进行关联.在 2017 年,Woike 等人提出了 DeepSORT 算法[13],该算法在 SORT 算法的基础上使用重识别网络 WRN[14]提取出每个目标进行外观特征后,通过级联匹配对每个目标进行最佳匹配,提高了跟踪精度.在 2020 年,Wang 等人提出了一种新的多目标跟踪范式 JDE[15].该范式将目标检测和外观特征学习任务整合到一个网络中.这种设计提高了多目标跟踪算法的速度,达到了接近实时的性能.同时,其跟踪精度与当时的先进跟踪器相当.在 2022 年,Zhang 提出了 Bytetrack[16],该算法将每个检测框根据置信度的大小分成高分框和低分框分开处理,利用低分框和轨迹之间的相似性,从低分框中找出感兴趣的目标,保持轨迹的连贯性.在 2023 年,Meng 提出时空自适应的多特征融合跟踪方法[17],通过空间正则化与动态学习率调整,强化滤波器学习,特征上结合 fHOG、CN 及 VGG 卷积特征,确保信息准确高效,在复杂背景、遮挡、快速运动等场景,提升了跟踪鲁棒性.Wang 提出了结合卷积 Transformer 的目标跟踪算法 CTTrack[18],结合卷积神经网络的局部特征提取能力和 Transformer 的长距离依赖属性,提高跟踪效果和速度,并通过设计新骨干网络、特征互增强与聚合网络,以及自适应动态调整搜索区域策略,进一步提升跟踪精度和稳定性.

尽管上述研究采用不同方式提高了多目标跟踪的跟踪精度,在目标跟踪过程中,提取到更精确的目标外观特征对于计算目标之间的相似度度量至关重要.然而,DeepSORT 算法采用简单的卷积特征提取目标外观特征信息,对目标的外观特征描述不够充分,无法满足实际应用场景的需求.融合 DeepSORT 算法中的重识别网络提取的描述目标间外观相似度的相似度矩阵与 SIFT[19]提取的目标间相似度矩阵,还采用与外观无关的连接模型与高斯平滑插值法[20]对目标轨迹进行优化,丰富目标的外观特征信息,提高算法在复杂场景下的跟踪精度.同时在公共数据集上验证了对该算法在准确度与精度上的提升.

1　本文算法框架

1.1　DeepSORT 模型

DeepSORT 是一种引入深度学习模型解决多目标跟踪任务的算法,其主要思想是结合目标检测和跟踪两个任务.首先使用目标检测算法在每一帧中检测出目标物体的位置.然后基于 SORT 算法,利用重识别模型提取目标的外观信息,将每个目标与先前帧中已跟踪的目标进行匹配.匹配过程中计算目标之间外观特征、运动特征的相似度,确定目标的身份和轨迹.DeepSORT 算法的主要步骤为:1)用目标检测器检测图像中的目标,获取到目标检测框;2)将这一帧的目标检测框与上一帧卡尔曼滤波预测的预测框进行级联匹配;3)将级联匹配中不匹配的轨迹和检测再进行一次 IOU 匹配;4)将不能确定和大于设定最大寿命的轨迹删除;5)更新目标的轨迹.

其中级联匹配这一步骤解决了当目标被遮挡时身份标识切换的问题.而级联匹配的主要流程如下:根据目标检测模型检测出图像中目标的检测框,通过卡尔曼滤波器对上一帧目标的运动轨迹进行预测,得到预测框.然后将检测框和预测框输入到外观模型,分别提取出外观特征 A 和外观特征 B.计算检测框 A 和预测框 B 之间的马氏距离与外观特征 A 和外观特征 B 之间的余弦距离,再根据这两个矩阵构建出代价矩阵.马氏距离计算公式如下所示:

$$d^{(1)}(i,j)=(d_j-y_i)^T S_i^{-1}(d_j-y_i) \tag{1}$$

其中,i 为第 i 个轨迹,j 为第 j 个检测,y_i、S_i 分别为某帧目标的预测位置和预测协方差矩阵,d_j 为下一帧的检测框信息.

余弦距离计算公式如下:

$$d^{(2)}(i,j)=\min\{1-r_j^T r_k^{(i)} \mid r_k^{(i)}\in R_i\} \tag{2}$$

其中,r_j 为第 j 个检测的外观特征向量,$r_k^{(i)}$ 为第 i 个检测的所有外观特征向量.代价矩阵计算公式如下:

$$c_{i,j}=\lambda d^{(1)}(i,j)+(1-\lambda)d^{(2)}(i,j) \tag{3}$$

其中,λ 为权重系数.代价矩阵同时考虑了目标的外观信息和运动信息,用于衡量检测框与预测框之间的匹配程度.当得到代价矩阵后,使用匈牙利算法按照匹配程度对检测框与预测框进行最优匹配,实现相同目标之间的关联,得到此帧的匹配结果.

1.2　SIFT 特征提取算法

SIFT(Scale Invariant Feature Transformation),即尺度不变特征变换,是由 David Lowe 于 2004 年完善的基于尺度空间的特征点检测算法.SIFT 算法的主要思想是将传统的图像匹配难题转化为更为简洁和直观的特征向量匹配问题.它通过在尺度空间上查找关键点,并赋予这些关键点详细的信息(如位置、尺度和方向),生成具有独特性的特征向量.这些特征向量具有尺度不变性、平移不变性和旋转不变性,对光照变化、仿射变换和噪声也具有一定的稳定性.同时,SIFT 算法具有很强的的鲁棒性和较快的运算速度.

1.3　多相似度矩阵融合

DeepSORT 算法在级联匹配中通过简单的重识别网络特征提取,无法充分描述目标的外观特征,无法适应实际复杂场景的需要,本文通过提取目标 SIFT 特征,与 DeepSORT 算法通过重识别网络提取的特征融合,丰富目标外观特征信息.对于某个需要跟踪的目标,DeepSORT 算法采用了行人重识别领域的宽残差网络 WRN 来描述目标的外观特征,但仅依靠 WRN 提取的外观信息是不足够的.因此本文提出了在得到相似度矩阵后再对其进行融合的算法,这种融合结合了 SIFT 对局部纹理的敏感性和 WRN 对深层特征的学习能力,还充分利用了两种特征之间的互补性,更综合考虑目标的外观信息.如图 1 所示,本文算法首先分别提取目标的 SIFT 特征和利用 DeepSORT 算法中的 WRN 网络所提取的外观特征.然后针对每一个目标,计算其与其余目标间的 SIFT 特征相似度以及 WRN 外观特征相似度,构建出两个独立的相似度矩阵.在此基础上,将基于 SIFT 特征的相似度矩阵与基于 WRN 特征的相似度矩阵进行融合,充分整合两种特征的互补优势,获得一

个更为全面的最终相似度矩阵.

本文算法流程如图 2 所示. 主要流程如下:

1)根据目标检测模型检测出目标的检测框,通过卡尔曼滤波器对上一帧目标的运动轨迹进行预测,得到预测框. 然后将检测框和预测框输入到外观模型 WRN 和 SIFT 特征提取器,分别提取出检测框的 SIFT 特征 A、预测框的 SIFT 特征 B、检测框的 WRN 特征 C、预测框的 WRN 特征 D.

2)计算检测框和预测框之间的马氏距离、检测框 SIFT 特征 A 和预测框特征 B 之间的余弦距离与检测框 WRN 特征 C 和预测框 WRN 特征 D 之间的余弦距离,再根据这 3 个距离构建出 WRN 特征相似度矩阵和 SIFT 特征相似度矩阵.

3)当得到两个相似度矩阵后,按权重进行融合得到最后的代价矩阵.

$$c_{i,j}=\lambda c^{(1)}(i,j)+(1-\lambda)c^{(2)}(i,j) \tag{4}$$

其中,λ 为权重系数,$c^{(1)}(i,j)$ 为 WRN 特征相似度矩阵,$c^{(2)}(i,j)$ 为 SIFT 特征相似度矩阵.

4)当得到代价矩阵后,使用匈牙利算法按照未成功匹配的帧数从 0 开始递增到 $A_{\max}$ 对轨迹进行循环匹配,未成功匹配的帧数小的优先进行匹配,其中匹配成功的轨迹和检测需要在门控矩阵的阈值内,循环次数大于 $A_{\max}$ 时,结束循环匹配,返回未匹配成功的轨迹和检测,得到匹配结果.

5)将匹配结果中未匹配的轨迹和未匹配的检测进行一次 IOU 匹配.

6)将不能确定的轨迹和大于设定最大寿命的轨迹删除.

7)将两次匹配成功的轨迹通过卡尔曼滤波,未匹配的检测新建为新轨迹,更新到下一帧的轨迹.

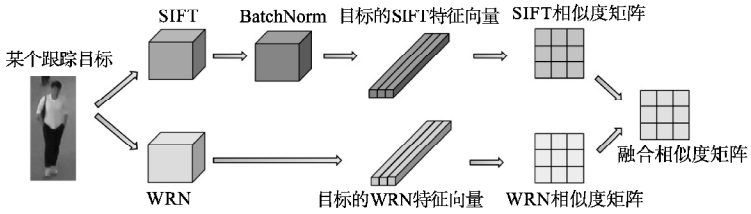

图 1 相似度矩阵融合过程

Fig. 1 Similarity matrix fusion process

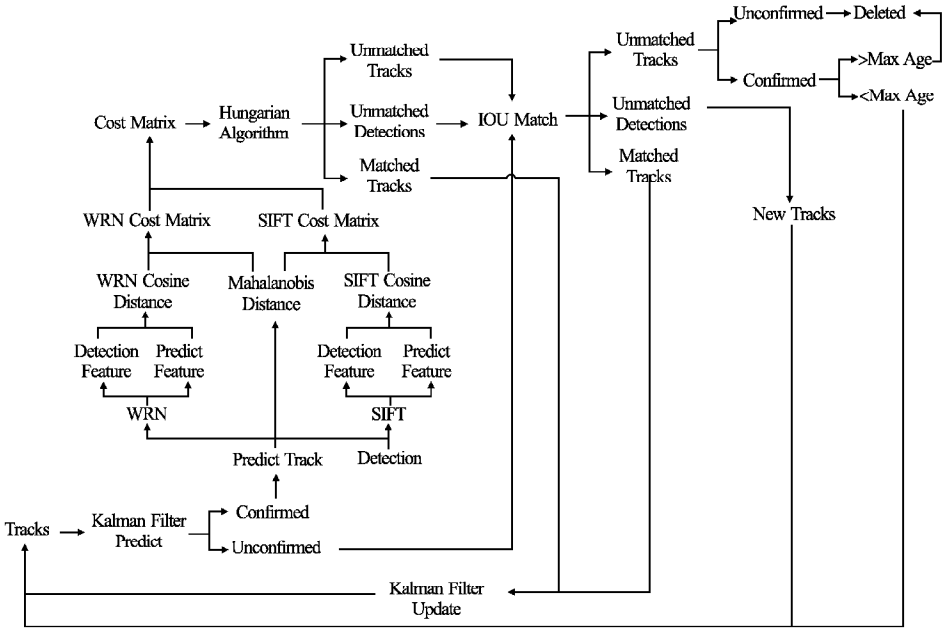

图 2 算法流程

Fig. 2 Algorithm process

1.4 优化轨迹处理

为了进一步优化跟踪结果,本文算法使用与外观无关的连接模型将短的轨迹关联成完整的轨迹,并通过高斯平滑插值法填补因缺失检测而造成的轨迹空白.

1.4.1 与外观无关的连接模型

传统的轨迹关联方法往往依赖于目标的外观信息,这在某些情况下可能导致关联错误,特别是在目标外观发生显著变化或遮挡时. 而与外观无关的连接模型(AFLink, Appearance-Free Link)不依赖于外观信息,它通过利用空间关系和时间连续性,来连接短的轨迹片段,形成完整的轨迹. 这使得 AFLink 在处理复杂场景时具有更高的鲁棒性和适应性,从而提高了整体算法的跟踪准确性和稳定性.

图 3 展示了 AFLink 模型的两个分支框架,为轨迹关联提供了一种高效且准确的方法. 模型以轨迹 T_i 和 T_j 作为输入,其中轨迹 $T_*=\{f_k,x_k,y_k\}_{k=1}^{N}$ 由最近 $N=30$ 帧 f_k 和对应位置 (x_k,y_k) 组成,对于那些长度小于 30 帧的轨迹,模型采用零填

充的方式,确保了输入的一致性,便于后续处理. 接着,模型在时间模块利用 7×1 的卷积核对输入轨迹沿时间维度进行卷积操作,提取出与时间相关的特征. 随后,模型在融合模块通过 1×3 的卷积操作,整合来自不同特征维度(即 f,x,y)的信息. 经过融合模块处理后,得到的两个特征图分别经过池化和压缩处理后转化为特征向量,然后拼接这两个特征向量,这些特征向量包含了丰富的时空信息. 最后,模型利用多层感知机(MLP)预测轨迹关联的置信度得分.

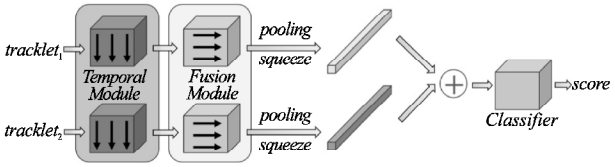

图 3　外观无关的连接模型框架

Fig. 3　Framework of the appearance free link model

1.4.2　高斯平滑插值法

插值被广泛用于填补由缺失检测引起的轨迹中的间隙. 线性插值作为一种常见的插值方法,由于其简单性和计算效率而广受欢迎. 线性插值的基本思想是在两个已知点之间绘制一条直线,并用这条直线上的点来估计缺失的数据. 这种方法假设目标在轨迹中的运动是匀速的,这在许多情况下并不成立. 实际上,目标的运动往往是复杂多变的,可能受到多种因素的影响,如速度变化、加速度、方向改变等. 因此,仅仅使用线性插值来填补轨迹间隙,往往无法准确地反映目标的真实运动状态. 相比之下,高斯平滑插值法(GSI,Gaussian-Smoothed Interpolation)考虑了目标的运动特性,以生成更接近真实运动的插值结果,更好地利用运动信息来提高插值的准确性.

高斯平滑插值方法的核心思想是利用高斯函数对轨迹数据进行平滑处理,并在平滑后的数据基础上进行插值. 首先计算轨迹数据的高斯加权平均值,这一过程会考虑到数据点之间的空间和时间关系,以及它们的权重. 权重通常根据数据点与待插值点之间的距离或其他相关因素来确定,距离越近的点通常具有更高的权重. 通过高斯加权,高斯平滑插值能够更好地捕捉轨迹的局部特征和变化趋势. 在得到平滑处理后的轨迹数据后,高斯平滑插值会使用这些数据来估计缺失位置的值. 插值过程中,不仅考虑已知数据点的值,还充分考虑了它们之间的空间和时间关系,以及轨迹的整体趋势. 这使得高斯平滑插值能够生成更接近于真实轨迹的插值结果.

高斯平滑插值将第 i 条轨迹建模如下:

$$p_t = f^{(i)}(t) + \epsilon \tag{5}$$

其中 t 为帧集合 F 里的帧,p_t 为第 t 帧的位置坐标,$p_t \in P$,ϵ 为高斯噪声,服从 $N(0,\sigma^2)$.

给定长度为 L 的轨迹 $S^{(i)} = \{t^{(i)}, p_t^{(i)}\}_{t=1}^{L}$,通过拟合函数 $f^{(i)}(t)$ 来解决非线性运动建模问题. 假设其服从高斯过程 $f^{(i)}(t) \in GP(0, k(\cdot,\cdot))$,其中 $k(x,x') = \exp\left(-\frac{\|x-x'\|^2}{2\lambda^2}\right)$ 为径向基核函数. 根据高斯过程的性质,在给定新的帧集 F^* 的情况下,对其平滑位置 P^* 进行了预测.

$$P^* = K(F^*,F)\,(K(F,F) + \sigma^2 I)^{-1} P \tag{6}$$

其中 $K(\cdot,\cdot)$ 是基于 $k(\cdot,\cdot)$ 的协方差函数.

此外,超参数 λ 控制轨迹的平滑度,其大小与轨迹长度 l 有关.

$$\lambda = \tau * \log\left(\frac{\tau^3}{l}\right) \tag{7}$$

其中 τ 的值设置为 10.

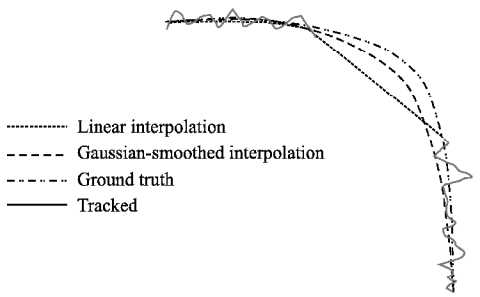

图 4　线性插值和高斯平滑插值的对比

Fig. 4　Comparison of linear interpolation and gaussian smooth interpolation

图 4 举例说明了高斯平滑插值与线性插值之间的区别. 在实际应用中,原始跟踪结果常常受到噪声和抖动的影响,导致轨迹数据不够平滑和准确. 线性插值虽然简单易行,但由于其忽略了运动信息,往往难以有效地处理这些问题. 相比之下,高斯平滑插值方法采用了自适应平滑因子,能够对整个轨迹进行平滑处理. 它充分考虑了轨迹数据中的空间和时间关系,通过高斯函数对轨迹点进行加权处理,高斯平滑插值的结果比线性插值更加平滑和准确,更接近真实运动状态. 高斯平滑插值不仅能够有效去除噪声和抖动,还能够保留轨迹的主要特征,提高了轨迹的准确性和可靠性.

2　实验与分析

本文实验部分所采用的电脑配置如下:GPU 采用 NVIDIA GTX 2080Ti,Ubuntu 系统 18.04,CPU 采用 Intel Core i9-10900kf,内存为 32GB,CUDA 版本为 11.2,pytorch 版本为 1.8,python 版本为 3.8.5.

2.1　数据集与评价指标

本文使用数据集选用 MOTchallenge 官方提供的多目标跟踪数据集 MOT16,场景为街道中行人轨迹跟踪[21]. 该数据集共有 14 个视频序列,分别为 7 段带有标注和 7 段不带标注的视频序列,本文选用 MOT16 数据集的 7 段带有标注视频序列进行实验,共有 5316 张图片. 本文所用的评价指标如表 1 所示.

表 1　多目标跟踪评价指标

Table 1　Multi-object tracking metrics

评价指标	说　明
MOTA	多目标跟踪精度. 衡量多目标跟踪算法性能的主要测试指标
IDF1	衡量算法识别和保持目标身份的准确性
FP	误检的目标总数
FN	漏检的目标总数
IDS	某条轨迹对应的目标身份发生变化的次数

2.2　实验结果与分析

本文以 YOLOv5 作为目标检测器,得到检测结果后,对 DeepSORT 算法和本文算法进行实验证明本文算法的有效性. 同时为了进一步证明本文算法的有效性,本文还对比了只使用 SIFT 作为目标外观提取模型的算法跟踪性能.

表 2 中呈现了不同权重 λ 下的本文算法在 MOT16 数据集上的实验结果. 表中 ↑ 表示该指标越大性能越好,↓ 代表该指标越小性能越好. 可以看到,当权重 λ 为 0.8 时,本文算法在 MOTA、IDF1 和 IDS 这 3 个关键评价跟踪性能的指标上取得了最优的表现. 在 FN 和 FP 这两个指标上,虽然权重 λ 为 0.8 时的算法略差于其他权重下的算法,但这两个指标对算法的性能几乎没有影响. 同时可以看到,权重 λ 为 1 下的算法取得了 FN 的最优表现,权重 λ 为 0.1 下的算法取得了 FP 的最优表现. 然而,这两种权重设置下的算法在其他关键指标上的表现并不理想.

表 2 不同权重下的算法效果对比

Table 2 Comparison of algorithm under different weights

λ 取值	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDS ↓
0	53.02	51.91	10261	40485	1123
0.1	53.07	51.56	**10228**	40482	1105
0.2	53.06	51.55	10237	40482	1105
0.3	53.04	51.56	10260	40475	1103
0.4	52.94	51.51	10378	40486	1103
0.5	52.86	51.64	10516	40433	1094
0.6	52.96	53.21	10919	40041	967
0.7	53.58	58.83	11730	38902	610
0.8	**54**	**61.21**	12356	37979	**452**
0.9	53.83	57.35	12904	37576	490
1	53.01	55.34	14218	**37050**	611

综上所述,权重 λ 为 0.8 时,本文算法在关键评价跟踪性能的指标上取得了最优表现,且在其他指标上的表现也相对稳定. 因此,在后续的实验中,本文将采用权重 λ 为 0.8 的设置,以进一步验证本文算法的性能.

表 3 中呈现了本文算法、仅用 SIFT 作为外观特征提取模型的算法、DeepSORT 算法、ByteTrack 和 StrongSORT 在 MOT16 数据集上的实验结果.

表 3 不同算法效果对比

Table 3 Comparison of different algorithm performances

算法	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDS ↓
DeepSORT	53.01	55.34	14218	**37050**	611
SIFT	53.02	51.91	10261	40485	1123
ByteTrack	53.38	59.85	**5332**	45784	**350**
StrongSORT	**54.94**	**61.24**	7362	41906	473
Ours	54	61.21	12356	37979	452

在 MOTA 这一关键指标上,本文提出的算法相比仅使用 SIFT 提取目标外观特征或仅用 WRN 提取目标外观特征的 DeepSORT 算法,平均提高了 1 个点. 在 IDF1 这一反映跟踪目标身份一致性的指标上,本文算法的表现同样出色,与仅使用 SIFT 和 WRN 的算法相比,本文算法的提升达到 5.77 个点和 9.3 个点. 在 IDS 指标上,本文算法与仅使用 SIFT 或 WRN 的算法相比,分别下降了 159 和 671. 然而,在 FP、FN 这两个指标上,本文算法略微差于每个单独的算法. 但这两个指标的略微下降对算法的性能几乎没有影响. 综合考虑所有指标,本文算法在主要评价指标上都优于仅用外观提取器的算法,显示了算法在整体性能上的优势.

同时,本文所提出的算法在多个指标上均取得仅次于最优算法的表现,这证明了其在多目标跟踪任务中的总体性能优良,不存在明显短板,更具有鲁棒性.

针对目标模糊这个问题,本文同样设计了实验来评估本文算法、仅用 SIFT 作为外观特征提取模型的算法与 DeepSORT 算法在应对不同模糊程度时的性能. 表 4 呈现了本文算法、SIFT 算法和 DeepSORT 算法在不同模糊程度时的实验结果.

表 4 算法在不同模糊程度下的表现

Table 4 Algorithm performance at different levels of blur

模糊程度	算法	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDS ↓
无高斯模糊	DeepSORT	53.01	55.34	14218	37050	611
	SIFT	53.02	51.91	10261	40485	1123
	Ours	54	61.21	12356	37979	452
卷积核大小为 3 * 3 的高斯模糊	DeepSORT	52.95	55.18	14253	37093	600
	SIFT	52.93	51.59	10387	40475	1107
	Ours	53.96	61.02	12348	38019	459
卷积核大小为 5 * 5 的高斯模糊	DeepSORT	52.94	54.87	14252	37085	610
	SIFT	52.87	51.51	10479	40450	1107
	Ours	54.05	61.71	12263	38023	439
卷积核大小为 7 * 7 的高斯模糊	DeepSORT	52.98	54.42	14220	37082	609
	SIFT	52.73	51.62	10633	40410	1140
	Ours	53.99	60.86	12244	38093	455

从表 4 中可以看出,在不同程度的高斯模糊场景下,本文算法在 MOTA 这一关键指标上保持了基本不变的趋势. 这一稳定性表明,算法在目标跟踪的精度方面具有较好的鲁棒性,能够在目标受到模糊影响时保持相对稳定的跟踪性能. 这一特点对于实际应用中的目标跟踪任务至关重要,尤其是在复杂多变的环境中. 其次,在 IDF1 指标上,本文算法虽然略有降低,IDF1 指标主要衡量的是目标身份识别的准确性,因此这一指标的略微下降可能意味着在模糊场景下,算法对目标身份的识别能力受到了一定程度的影响,但整体而言,这种影响并不足以对算法的性能造成影响. 此外,通过对比其他指标 FP、FN 和 IDS,可以看出在高斯模糊场景下,这些指标与无模糊场景下的表现基本持平. 这意味着算法在模糊场景下依然能够保持较低的误检、漏检和身份切换的次数,确保了跟踪的稳定性和连续性. 而 SIFT 算法和 DeepSORT 算法在高斯模糊的场景下,各项指标均出现了降低.

在跟踪任务中,目标尺寸的大小对算法性能的影响是一个不可忽视的因素. 特别是在实际场景中,由于摄像头视角、目标距离等因素,经常会出现目标尺寸较小的情况,这给算法的准确性和稳定性带来了挑战. 本文也设计了针对不同尺寸大小目标的实验,以验证本文算法在小尺寸目标情况下的性能表现. 表 5 呈现了本文算法、仅用 SIFT 作为外观特征提取模型的算法与 DeepSORT 算法测试不同尺寸大小的目标的实验结果.

从表 5 中可以看出,本文算法在衡量跟踪算法性能的重

要指标 MOTA 上,无论是 100% 尺寸还是 50% 尺寸的目标,MOTA 都保持在较高水平,并且随着目标尺寸的缩小,MOTA 甚至还有所提升. 这说明了算法在应对不同尺寸目标时,都能够有效地维持跟踪的准确度. 这一稳定性表明,算法在目标跟踪的精度方面具有较好的鲁棒性,能够在目标尺寸小时保持相对稳定的跟踪性能. 其次,本文算法在 IDF1 和 IDS 指标上均有提升,说明算法在处理不同尺寸目标时,能够保持目标身份一致性. 此外,本文算法在指标 FP 上略有提升,在指标 FN 上略有降低,但 FN 指标的略微下降,不足以对算法的性能造成影响. 这说明了本文算法在在目标尺寸小的场景下也能够实现精确的目标跟踪. 而 SIFT 算法和 DeepSORT 算法在目标尺寸缩小的的场景下,各项指标均出现了降低.

表 5　算法在不同尺寸大小下的表现

Table 5　Algorithm performance at different sizes

尺寸大小	算法	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDS ↓
100%	DeepSORT	53.01	55.34	14218	37050	611
	SIFT	53.02	51.91	10261	40485	1123
	Ours	54	61.21	12356	37979	452
50%	DeepSORT	52.93	55	14266	37081	618
	SIFT	52.71	51.39	10637	40410	1159
	Ours	54.04	61.67	12268	38033	440

在实际生活场景中,目标模糊和尺寸小的情况会经常同时出现,为了探究算法对模糊的小目标的跟踪性能,本文设计了模糊的小目标的复杂场景的实验. 表 6 呈现了本文算法、仅用 SIFT 作为外观特征提取模型的算法与 DeepSORT 算法对目标进行模糊处理和尺寸缩小的实验结果.

表 6　算法在目标进行模糊处理和尺寸缩小的表现

Table 6　Performance of the algorithm in blurring and downsizing objects

模糊程度 / 尺寸大小	算法	MOTA ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓
无高斯模糊/100%	DeepSORT	53.01	195	95	14218	37050	611
	SIFT	53.02	165	116	10261	40485	1123
	Ours	53.42	179	103	13329	37570	528
卷积核大小为 3 * 3 的高斯模糊/50%	DeepSORT	52.95	191	96	14202	37141	599
	SIFT	52.71	165	115	10699	40362	1153
	Ours	53.44	175	97	13344	37502	538
卷积核大小为 5 * 5 的高斯模糊/50%	DeepSORT	52.97	192	96	14226	37105	591
	SIFT	52.6	164	110	11036	40155	1143
	Ours	53.42	180	98	13394	37485	547
卷积核大小为 7 * 7 的高斯模糊/50%	DeepSORT	52.96	188	96	14243	37101	589
	SIFT	52.43	163	108	11496	39743	1273
	Ours	53.47	180	97	13412	37399	559

从表 6 中可以看出,本文算法在目标模糊和尺寸小场景下这些指标上能够继续保持基本不变的趋势,甚至略有提升. 这意味着本文算法不仅在应对目标模糊和尺寸缩小的挑战上表现出色,也进一步证明了本文算法在面对目标模糊和尺寸小等复杂场景下的目标跟踪能力.

为了探究优化轨迹方法对算法的跟踪性能影响,本文设计了两个优化方法的消融实验. 通过对比使用不同优化方法组合时算法的跟踪性能,深入地理解这些优化方法对算法性能的具体贡献.

表 7　优化轨迹方法的消融实验

Table 7　Ablation experiment of optimal trajectory method

算法	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDS ↓
DeepSORT	53.01	55.34	14218	37050	611
SIFT	53.02	51.91	10261	40485	1123
Ours	54	61.21	12356	37979	452
Ours + AFLink	54.02	61.09	12352	37980	434
Ours + GSI	54.25	61.38	12234	37852	423
Ours + AFLink + GSI	54.26	61.24	12239	37858	406

表 7 呈现了两个优化方法在本章算法上的实验结果. 从表中可以看出,使用 AFLink 优化方法后,算法在 MOTA、FP、FN 和 IDS 指标上略有提升,这表明 AFLink 优化方法能够有效提升算法的跟踪准确性、减少误检和身份切换. 虽然 IDF1 略微有降低,但这点降低不足以对算法的性能造成影响. 使用 GSI 优化方法后,算法在 MOTA、IDF1、FP、FN、IDS 所有指标上均表现出性能的提升. 当同时使用 AFLink 和 GSI 两种优化方法时,算法在 IDS 这个指标上有了更为明显的提升,这说明两种优化方法在减少身份切换方面能够相互补充,共同发挥作用. 而其他指标如 MOTA、IDF1、FP 和 FN 则与仅使用 GSI 时的表现相差不大,这表明两种优化方法在其他方面的效果可能存在一定的重叠. 然而,这并不影响它们共同提升算法跟踪性能的事实.

综上所述,AFLink 和 GSI 这两种优化方法都能在一定程度上提升算法的跟踪性能. 当同时使用这两种优化方法时,它们能够相互补充,共同提升算法的跟踪性能.

2.3　算法效果可视化对比

为了验证本文提出算法的有效性,对 DeepSORT 算法、SIFT 算法和本文算法在 MOT16 数据集标注的视频序列上进行了可视化,直观地展现文中算法的跟踪效果.

图 5 分别展示了 DeepSORT 算法、SIFT 算法和本文算法在 MOT16 数据集的 MOT16-10 序列上的第 15 帧、第 30 帧、第 41 帧的跟踪结果. 在 DeepSORT 算法的结果中,序列 MOT16-10 第 30 帧中 id 为 14 的目标被 id 为 5 和 id 为 10 的目标遮挡后,其 id 从 14 跳变为了第 41 帧时的 18. 在 SIFT 算法的结果中,序列 MOT16-10 第 30 帧中 id 为 27 的目标被 id 为 5 和 id 为 10 的目标遮挡后,其 id 从 27 跳变为了第 41 帧

时的68. 相比之下,在本文算法中,目标在第30帧时被其他目标遮挡,算法仍然能够准确地跟踪到id为14的目标,并在第41帧时保持其身份标识不变.

图5 在MOT16-10序列上的跟踪效果对比

Fig. 5 Tracking performance of MOT16-10 sequence

3 结 语

目标模糊、目标尺寸小、遮挡造成目标外观特征不稳定,容易造成跟踪任务出现ID切换问题. 本文引入SIFT和WRN的相似度矩阵融合,丰富目标的外观信息描述. 此外,本文还采用与外观无关的连接模型将短的轨迹关联成完整的轨迹,并通过高斯平滑插值法填补因缺失检测而造成的轨迹空白. 对比实验及分析表明,本文提出的跟踪算法在目标尺寸小、遮挡、模糊等复杂场景下展现了良好的性能,MOTA达到54. 26%、IDF1分数达到61. 24%、ID Switch仅为406,与DeepSORT算法相比,本文算法具有更好的鲁棒性和跟踪精度,能够更好地应用在目标跟踪领域. 未来将继续探索更多的策略,以进一步提升算法的跟踪性能.

References:

[1] Luo W, Xing J, Milan A, et al. Multiple object tracking: a literature review[J]. Artificial Intelligence, 2021, 293: 103448, doi: 10.1016/j. artint. 2020. 103448.

[2] Dendorfer P, Osep A, Milan A, et al. Motchallenge: a benchmark for single-camera multiple target tracking[J]. International Journal of Computer Vision, 2021, 129: 845-881, doi: 10. 1007/s11263-020-01393-0.

[3] Ciaparrone G, Sánchez F L, Tabik S, et al. Deep learning in video multi-object tracking: a survey[J]. Neurocomputing, 2020, 381: 61-88, doi: 10. 1016/j. neucom. 2019. 11. 023.

[4] Liu W, Anguelov D, Erhan D, et al. Ssd: single shot multibox detector[C]//Proceedings of European Conference on Computer Vision, 2016: 21-37.

[5] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779-788.

[6] Redmon J, Farhadi A. Yolov3: an incremental improvement[EB/OL]. https://arxiv. org/abs/1804. 02767, 2018.

[7] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014: 580-587.

[8] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE International Conference on Computer Vision, 2015: 1440-1448.

[9] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(6): 1137-1149.

[10] Bewley A, Ge Z, Ott L, et al. Simple online and realtime tracking[C]//Proceedings of the IEEE International Conference on Image Processing, 2016: 3464-3468.

[11] Kalman R E. A new approach to linear filtering and prediction problems[J]. Journal of Basic Engineering, 1960, 82(1): 35-45.

[12] Kuhn H W. The Hungarian method for the assignment problem[J]. Naval Research Logistics Quarterly, 1955, 2(1-2): 83-97.

[13] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric[C]//Proceedings of the IEEE International Conference on Image Processing, 2017: 3645-3649.

[14] Zagoruyko S, Komodakis N. Wide residual networks [EB/OL]. https://arxiv. org/abs/1605. 07146, 2016.

[15] Wang Z, Zheng L, Liu Y, et al. Towards real-time multi-object tracking[C]//Proceedings of European Conference on Computer Vision, 2020: 107-122.

[16] Zhang Y, Sun P, Jiang Y, et al. Bytetrack: multi-object tracking by associating every detection box [C]//Proceedings of European Conference on Computer Vision, 2022: 1-21.

[17] BAI K Q, ZHU Y L, YANG X Q, et al. Visual SLAM algorithm of integrating dynamic target tracking[J]. Information and Control, 2024, 53(5): 574-584.

[18] WANG C L, ZHANG J L, LI M H, et al. Object tracking algorithm combining convolution and transformer[J]. Computer Engineering, 2023, 49(4): 281-288 + 296.

[19] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60: 91-110, doi: 10. 1023/B: VISI. 0000029664. 99615. 94.

[20] Du Y, Zhao Z, Song Y, et al. Strongsort: make deepsort great again[J]. IEEE Transactions on Multimedia, 2023, 25: 8725-8737, doi: 10. 48550/arXiv. 2202. 13514.

[21] Milan A, Leal-Taixé L, Reid I, et al. MOT16: a benchmark for multi-object tracking [EB/OL]. https://arxiv. org/abs/1603. 00831, 2016.

附中文参考文献:

[17] 白克强,朱亚兰,杨秀清,等. 融合动态目标跟踪的视觉SLAM算法[J]. 信息与控制, 2024, 53(5): 574-584.

[18] 王春雷,张建林,李美惠,等. 结合卷积Transformer的目标跟踪算法[J]. 计算机工程, 2023, 49(4): 281-288 + 296.