

基于路侧摄像头的多目标跟踪算法优化设计

王 平, 姚宇阳, 王新红

(同济大学 电子与信息工程学院, 上海 201804)

摘要: 针对当前多目标跟踪算法应对路侧交通场景的缺陷, 提出一种基于路侧摄像头的多目标跟踪算法。选择 one-shot 跟踪算法路线, 基于 FairMOT 设计神经网络, 使单个网络同时生成目标检测结果与外观特征结果, 增强实时性效果; 采用新的数据关联方式, 减少遮挡对跟踪器的影响; 引入新的运动相似度量方式——缓冲交并比, 弥补线性运动预测模型产生的误差; 提出基于速度判别的丢失轨迹移除算法和基于历史位置匹配算法, 实现长时间遮挡轨迹的身份恢复。在 UA-DETRAC 公开多目标跟踪数据集上进行实验, 验证该算法有效性。为证明该算法在真实路侧环境的适用性, 在国家智能网联汽车(上海)试点示范区开放道路采集真实路侧场景数据。最后, 将该算法和 SORT、DeepSORT、ByteTrack、FairMOT 算法在真实路侧场景数据上进行对比实验。实验结果表明, 本算法在 identification F-Score、ID switch、fragmentation、mostly tracked、mostly lost、multiple object tracking accuracy 等评估指标上优于其他算法。

关键词: 多目标跟踪; 目标检测; 路侧感知

中图分类号: TP311.1

文献标志码: A

Optimization Design of Multi-object Tracking Algorithm Based on Roadside Cameras

WANG Ping, YAO Yuyang, WANG Xinhong

(School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China)

Abstract: In response to the limitations of current multi-object tracking algorithms in handling roadside traffic scenarios, a multi-object tracking algorithm based on roadside cameras was proposed in this paper. First, the one-shot tracking algorithm chosen and a neural network based on FairMOT was built to simultaneously generate both object detection results and appearance feature results, thereby enhancing the real-time performance of

the algorithm. Then, a novel data association method was adopted to lessen the effect of occlusion on the tracker. After that, a new motion similarity measurement called buffered intersection over union was introduced to compensate for the errors caused by linear motion prediction models. Subsequently, a velocity-based discriminative algorithm for removing lost trajectories and a history-based position matching algorithm to retrieve the identities of occluded trajectories over lengthy periods of time were developed. Experiments were conducted on the UA-DETRAC public multi-object tracking dataset to verify the effectiveness of the algorithm. Additionally, to demonstrate the applicability of our algorithm in real-world roadside environments, real roadside scene data were collected on open field in the National Intelligent Connected Vehicle (Shanghai) Pilot Demonstration Zone. Finally, comparative experiments between the algorithm proposed and SORT, DeepSORT, ByteTrack and FairMOT algorithms were conducted using real-world roadside scene data. The experimental findings indicate that the proposed algorithm performs better than other algorithms in terms of identification F-score, ID switch, fragmentation, mostly tracked, mostly lost, and multiple object tracking accuracy.

Keywords: multi-object tracking; object detection; roadside perception

随着物联网、人工智能和 5G 等技术的快速发展, 旨在缓解道路拥堵、减少交通事故的智能交通系统(intelligent transportation systems, ITS)应运而生^[1]。早期的 ITS 研究主要集中在车载端, 通过在汽车上安装传感器, 采集周边环境信息, 进行感知。然而, 由于车身高度限制, 车载传感器感知范围有限, 导致复杂道路场景下的感知性能下降。对此, 研

收稿日期: 2023-11-11

基金项目: 上海市科委重点项目(22dz1203400)

第一作者: 王 平, 副教授, 博士生导师, 工学博士, 主要研究方向为智能网联汽车, 车路协同。

E-mail: pwang@tongji.edu.cn

通信作者: 姚宇阳, 硕士生, 主要研究方向为摄像头感知。E-mail: 2133003@tongji.edu.cn



论文
拓展
介绍

究人员推出车路协同的智能交通系统。车路协同系统将传感器部署在路侧端,进行环境感知,将获得的路况信息发送给附近车辆,实现车端与路侧端信息交互,保证交通安全并提高通行效率^[2]。

路侧感知传感器包括激光雷达、毫米波雷达、摄像机及它们的组合^[3]。其中,摄像机提供丰富的环境信息且成本低廉,成为感知系统的重要组成部分。多目标追踪(multiple object tracking, MOT)是视觉感知的重要环节,需要在没有目标外观、数量等先验信息下,追踪多个属于同一类或不同类的目标。

多目标追踪主要分为生成模型方法和判别模型方法两大类。生成模型方法对当前帧中的目标对象区域进行数学建模,在下一帧中寻找与目标最相似的区域作为跟踪结果,如mean shift^[4]、粒子滤波^[5]和卡尔曼滤波^[6]等方法。判别模型方法使用机器学习算法来学习目标特征,使用分类器在下一帧中寻找与当前帧最匹配的区域作为跟踪结果。当前主流的多目标追踪算法为判别模型方法,即基于检测结果进行追踪。基于检测结果的追踪算法包含4个部分:目标检测、外观特征提取、运动预测、关联计算。

目前基于检测结果的多目标追踪算法分为两类:two-shot系列算法和one-shot系列算法。

two-shot系列算法将检测网络和外观特征提取网络完全分离,用两个神经网络分别完成检测和外观特征提取。DeepSORT^[7]是two-shot系列算法中的经典算法。DeepSORT在SORT(simple online and realtime tracking)^[8]的基础上加上外观特征提取与匹配,性能较SORT有大幅度提升,但追踪速度有一定程度的降低。在DeepSORT发布以后,涌现大量基于DeepSORT的追踪算法,以BoT-SORT^[9]和StrongSORT^[10]为代表。BoT-SORT相较于DeepSORT,改进卡尔曼滤波,提高运动预测准确性。加入相机运动补偿,提出融合特征度量方案。StrongSORT相较于DeepSORT,加入相机运动补偿,使用噪声尺度自适应卡尔曼滤波^[11],提出无外观链路模型实现轨迹关联,使用高斯平滑插值^[12]补偿缺失的检测。two-shot系列算法适配各类检测器,可针对检测任务和外观特征提取任务分别调优。但检测任务和外观特征提取任务串联进行,速度慢。

one-shot系列算法使用一个神经网络,同时进行目标检测和外观特征提取。JDE(joint detection and embedding)^[13]作为one-shot系列算法的代表作,将外观特征提取网络合并到目标检测中,使检测器同时输出检测结果与外观特征,这种改进提高了算法运

行速度。FairMOT^[14]是one-shot领域的又一经典算法,以CenterNet^[15]作为检测器,使用两个同质分支网络,同等对待检测任务和外观特征提取任务。

基于上述研究现状,本文开展路侧多目标追踪研究,设计一种基于路侧摄像头的多目标追踪算法,以基于检测结果的多目标追踪算法为基础,选择合适的检测网络和外观特征提取网络。在公开数据集上评测,验证算法的先进性。在真实场景数据上测试,验证算法在真实环境的有效性。

1 基于路侧摄像头多目标追踪算法面临的挑战与改进策略

基于路侧摄像头的多目标追踪面临以下挑战:

首先,多目标追踪算法运行速度慢。当前主流的多目标追踪算法是two-shot系列算法。目标检测网络首先处理每帧图片,找到目标对象,生成检测框。特征提取网络对检测框中目标进行外观特征提取。由于目标检测与外观特征提取串行进行,算法运行速度慢。以DeepSORT及其扩展算法为例,根据本地运行结果和公开论文结果,DeepSORT在AMD 锐龙 6800H CPU和NVIDIA GeForce RTX 3060 Laptop GPU的硬件条件下,处理速度为9.07 FPS(frames per second)。StrongSORT在Tesla V100 GPU的硬件条件下,处理速度为7.10 FPS。BoT-SORT在英特尔酷睿 i9-11900F CPU和NVIDIA GeForce RTX 3060 Laptop GPU的硬件条件下,处理速度为4.50 FPS。随着多任务学习的成熟,可将目标检测和外观特征提取融合到一个网络,并行运行目标检测与外观特征提取。本算法借鉴FairMOT,设计合适的目标检测网络与外观特征提取网络,提高算法运行速度。

其次,路侧多目标追踪算法抗遮挡能力差。在路侧场景中,遮挡往往来自车辆间遮挡,交通信号灯、标志物对车辆、行人的遮挡,车辆对行人的遮挡,如图1所示。传统算法只考虑高置信度分数的检测框,筛选出高分检测结果与轨迹匹配。被遮挡目标的置信度分数低,传统算法直接丢弃,造成轨迹中断与丢失。对此,本文引入新的数据关联方式——BYTE^[16]。BYTE考虑每一个检测结果,被遮挡目标也会与轨迹匹配,减少遮挡的影响。

然而BYTE无法应对严重遮挡情况,即前车完全遮挡后车,检测器无法检测到被遮挡目标,没有检测结果与轨迹匹配,造成轨迹中断、丢失。当轨迹被

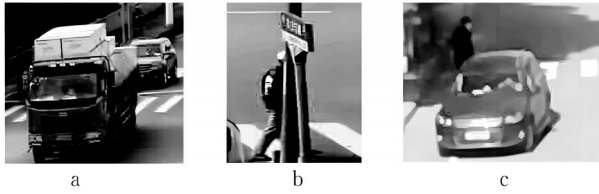


图1 路侧遮挡现象

Fig.1 Roadside occlusion phenomenon

长时间严重遮挡后,当前算法无法实现轨迹身份恢复。路侧交通场景中,长时间严重遮挡通常来自红灯期间车辆停靠。现实生活中,红灯时间为20~40 s,摄像头采样速率为30 FPS。红灯期间,遮挡时长为600~1 200帧。如图2所示,2车被1车完全遮挡,4车被3车完全遮挡。2车、4车轨迹在红灯期间处于丢失状态,遮挡结束后,轨迹身份无法恢复。长时间被遮挡轨迹身份丢失与笼统删除的丢失轨迹移除策略有关。轨迹丢失时间超过30帧,追踪器移除丢失轨迹。因原始轨迹被移除,目标再次出现后,被追踪器赋予新的身份。对此,本文提出两点改进措施:①提出基于速度判别的丢失轨迹移除算法,只有真正离开画面的轨迹才会被移除,拒绝笼统地移除丢失轨迹;②针对路侧红灯停靠场景,提出基于历史位置的匹配算法。



图2 路侧场景中的长时间遮挡

Fig.2 Long-term occlusion in roadside scenarios

最后,路侧多目标跟踪算法应对突发状况差。大多跟踪算法使用卡尔曼滤波进行运动预测。卡尔曼滤波为匀速运动模型,只能应对规则运动。路侧场景经常出现突发情况,例如目标突然拐弯、提速,如图3所示, t 时刻,白车正常向前行驶, $t+1$ 时刻,白车突然向左转弯。不规则突发运动往往导致卡尔曼滤波预测失效,运动相似度匹配失败,轨迹中断。对此,本文采用新的运动相似度度量方式——缓冲



图3 路侧突发状况

Fig.3 Roadside emergencies

交并比(buffered intersection over union, BIoU)^[17],扩大检测目标与轨迹的匹配空间,弥补卡尔曼滤波的预测偏差。

2 基于路侧摄像头多目标追踪算法框架结构

本算法使用基于FairMOT设计的卷积神经网络同时进行目标检测和外观特征提取,使用卡尔曼滤波,预测上一时刻轨迹在新时刻的运动状态,最后使用四阶段匹配算法进行关联计算,完成多目标追踪。框架结构如图4所示。

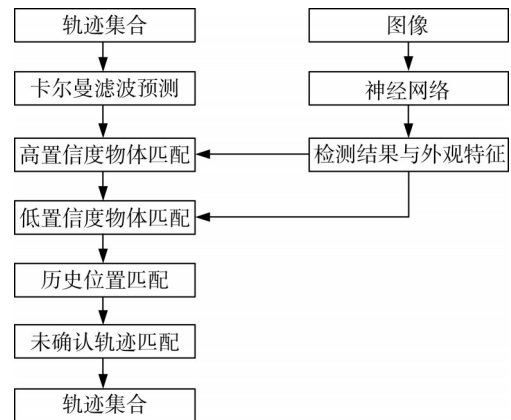


图4 算法框架结构

Fig.4 Framework structure of algorithm

首先, t 时刻图像经过神经网络得到检测结果与外观特征结果。 $t-1$ 时刻轨迹集合划分为:追踪轨迹、未确认轨迹、丢失轨迹。追踪轨迹与丢失轨迹经过卡尔曼滤波预测,得到在 t 时刻轨迹预测位置,与高分检测物进行高置信度物体匹配。匹配失败轨迹与低分检测物进行低置信度物体匹配。前两次匹配基于BYTE数据关联方式,两次匹配使用BIoU运动相似度度量。低置信度物体匹配失败的轨迹与前两次匹配失败的检测物进行历史位置匹配。最后,对于历史位置匹配失败的检测物,选出其中的高置信度物体,与未确认轨迹进行未确认轨迹匹配。经过上述4次匹配,得到 t 时刻轨迹集合。

3 基于路侧摄像头多目标追踪算法优化

3.1 基于FairMOT的目标检测网络与外观特征提取网络

本算法使用的目标检测卷积神经网络和外观特征提取卷积神经网络均基于FairMOT。使用两个

独立同质分支网络分别进行目标检测和外观特征提取,平等对待目标检测任务与外观特征提取任务。解决以往 one-shot 系列算法过度关注目标检测,外观特征提取过度依赖目标检测,提取的外观特征无法有效表征目标的缺陷。

本算法使用的卷积神经网络流程如图5所示,图片首先输入到预处理模块,把图片变成标准大小(1 088×608)。经过预处理的图片输入到编码器-解码器网络中,提取特征,生成高分辨率特征图。高分辨率特征图经过目标检测分支网络和提取外观特征分支网络,产生检测框和外观特征。

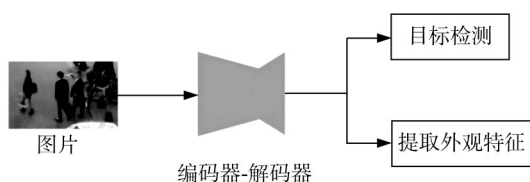


图5 卷积神经网络流程

Fig.5 Process of convolutional neural network

图6为编码器-解码器网络,编码器-解码器网络选用DLA-34(deep layer aggregation-34)^[18]结构,为目标检测分支网络和提取外观特征分支网络提供高分辨率特征图。编码器-解码器网络的卷积核步长为4,包括下采样、上采样、保持分辨率、求和4个模块。下采样减少特征图尺寸,生成缩略图。上采样进行反卷积,恢复特征图原始尺寸。输入图片的大小设置为 $W_{image} \times H_{image}$ 。其中, W_{image} 为输入图片的宽度, H_{image} 为输入图片的高度。编码器-解码器网络产生的特征图大小为 $C \times W \times H$ 。 W 为特征图的宽度, H 为特征图的高度, C 为目标的类别总数。 W 、 H 与 W_{image} 、 H_{image} 的关系为: $W = W_{image}/4$, $H = H_{image}/4$ 。

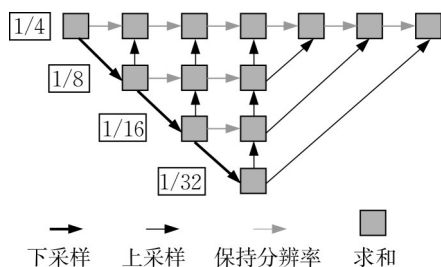


图6 编码器-解码器网络

Fig.6 Encoder-decoder network

图7为目标检测分支网络。目标检测分支包含3个平行的预测头,3个预测头分别预测热力图(heatmap)、目标中心点偏移量(center offset)和目标

尺寸(box size)。每个预测头使用256通道 3×3 大小的卷积核进行卷积运算。最后经过 1×1 卷积,生成最终结果。热力图预测头估计目标物体中心位置,输出维度为 $C \times W \times H$ 。中心偏移量预测头用于精确定位目标。编码器-解码器网络对图片下采样,带来4个像素的量化误差。偏移量预测头是为了减少误差,输出结果维度为 $W \times H \times 2$ 。尺寸预测头用来预测目标对象的尺寸,生成目标对象的宽和高。尺寸预测头的输出结果维度为 $W \times H \times 2$ 。最后将3个预测头产生的结果组合,形成目标对象的检测框。

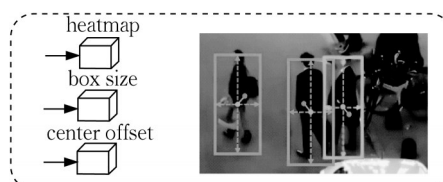


图7 目标检测分支网络

Fig.7 Branch network of object detection

图8为提取外观特征分支网络。提取外观特征分支网络用于生成目标的外观特征,相同目标的外观相似度大于不同目标的外观相似度。提取外观特征分支网络接收高分辨率特征图,在特征图上使用128通道 1×1 卷积核,进行卷积运算,生成目标的外观特征,外观特征维度为 128×1 。



图8 提取外观特征分支网络

Fig.8 Branch network of appearance feature extraction

3.2 基于路侧摄像头多目标追踪算法遮挡处理优化

本算法使用BYTE数据关联方式,减少遮挡对追踪器的干扰;提出新的丢失轨迹移除策略与匹配策略,恢复受长时遮挡轨迹的原始身份。

3.2.1 BYTE数据关联方式

传统追踪算法只考虑高分检测框,把低分检测框视为背景干扰,直接舍弃。但低分检测框不仅包含干扰,还包含因遮挡致使置信度分数降低的检测目标,一味舍弃会造成检测缺失。

检测器无法精准识别被遮挡目标,造成被遮挡目标置信度分数低。被遮挡目标被丢弃,造成轨迹

中断、丢失。

对此,本文引入BYTE数据关联方式。

首先设置阈值,以阈值为界,将检测物体分成高分集合与低分集合。

第一阶段,BYTE将高分检测物和轨迹进行匹配,匹配使用的相似度量度量为运动特征度量与外观特征度量的结合。

第二阶段,BYTE将低分检测物和第一次匹配失败的轨迹进行匹配。考虑到低分物体往往受到遮挡,外观特征不可靠,第二次匹配采用的相似度量度是运动特征度量。

被遮挡目标与轨迹的运动特征相似度大于背景与轨迹的运动特征相似度,轨迹优先与被遮挡目标

匹配,没有成功匹配的背景干扰被丢弃。这使得BYTE准确将背景过滤,保留被遮挡目标。

图9展示引入BYTE后算法的性能。图9a为检测器的输出结果,在 t 帧和 $t+1$ 帧时,白车没有遮挡摩托车,摩托车的检测分数为0.75和0.69。在 $t+2$ 帧和 $t+3$ 帧,摩托车被白色车辆遮挡,检测分数为0.31和0.39。图9b为引入BYTE前的追踪算法处理结果。 $t+2$ 和 $t+3$ 帧,摩托车的检测分数低,检测结果被丢弃,摩托车轨迹中断。图9c为引入BYTE后的追踪算法处理结果。引入BYTE后,追踪器会考虑每一个检测结果,将低分检测物与轨迹匹配,被遮挡目标被持续追踪,轨迹连续。

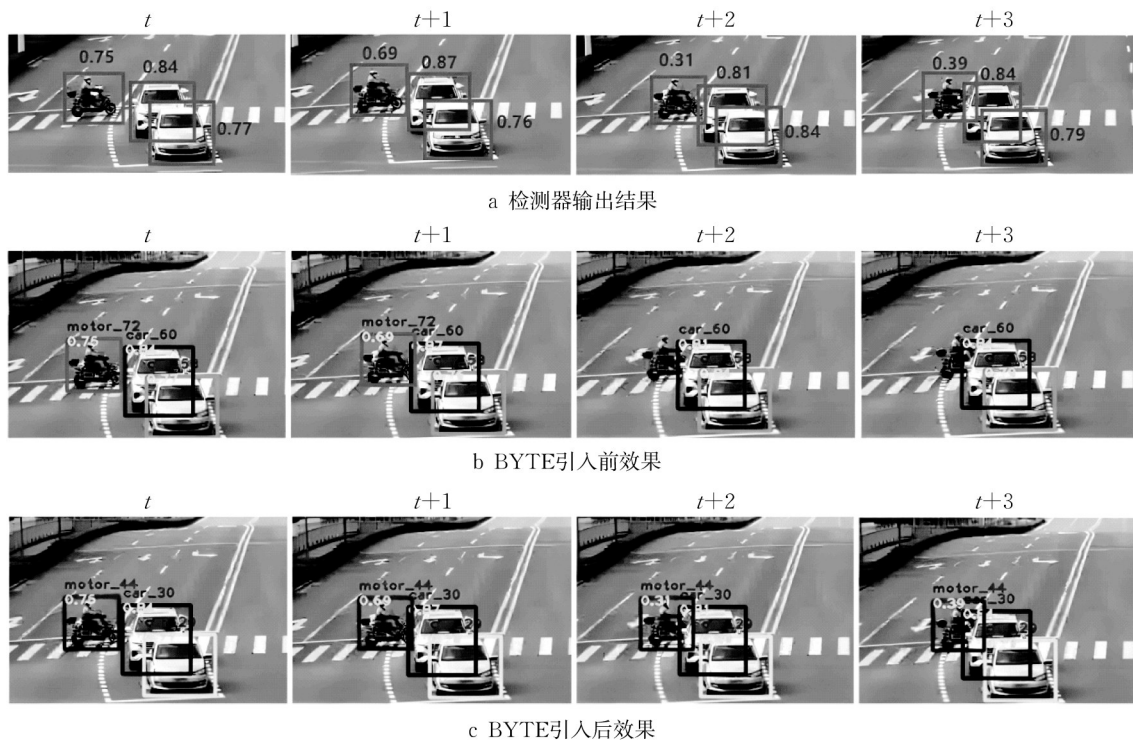


图9 BYTE效果展示

Fig.9 Demonstration of BYTE effect

3.2.2 长时间遮挡轨迹的身份恢复

以往追踪器不会考虑轨迹丢失的原因。轨迹丢失可能由于目标离开画面,也可能由于目标被严重遮挡。对于丢失的轨迹,追踪器只保留30帧,在这30帧中,丢失的轨迹与检测结果继续进行匹配。30帧后,若轨迹仍处于丢失状态,追踪器移除丢失轨迹。长时间遮挡导致轨迹丢失时间超过30帧,轨迹被移除。被遮挡目标重新显现后,因没有轨迹与该目标匹配,该目标被追踪器初始化为新轨迹,导致长时间遮挡前后轨迹身份切换。

本文采用基于速度判别的丢失轨迹移除算法删除离开画面的丢失轨迹,保留被长时间遮挡的丢失轨迹,为基于历史位置的匹配算法运行提供条件。同时使用基于历史位置的匹配算法将检测目标与丢失轨迹关联,完成身份恢复。

(1) 基于速度判别的丢失轨迹移除算法。

物体离开画面要满足两个条件:①边界条件,物体处在画面边界区域;②速度条件,物体的速度背离画面中心点。以图10为例,车辆离开画面的表现为车辆处在画面底部,且向下运动。基于速度判别的

丢失轨迹移除算法基于此。

设图片的宽为 W , 高为 H , (x, y, w, h, x_m, y_m) 为轨迹未丢失时状态量, (x, y) 为边界框左上角坐标, (w, h) 为边界框的宽和高, (x_m, y_m) 为边界框中心点坐标。设 $(x_1, y_1, w_1, h_1, x_{m1}, y_{m1})$ 为轨迹丢失时状态量, 由 (x, y, w, h, x_m, y_m) 经过卡尔曼滤波预测得到。如图 10 所示, t 时刻, 车辆被追踪, (x, y, w, h, x_m, y_m) 为车辆 t 时刻轨迹状态。 $t+1$ 时刻, 车辆驶出画面, 轨迹丢失, $(x_1, y_1, w_1, h_1, x_{m1}, y_{m1})$ 为车辆 $t+1$ 时刻轨迹状态。

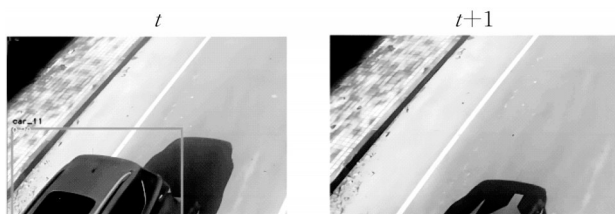


图 10 状态向量说明

Fig.10 Explanation of state vector

对于丢失的轨迹, 判断是否满足边界条件。 $(y_1 - b h_1) < 0$, 目标在上边界; $(x_1 - b w_1) < 0$, 目标在左边界; $(y_1 + h_1 + b h_1) > H$, 目标在下边界; $(x_1 + w_1 + b w_1) > W$, 目标在右边界。 b 为比例系数, 设为 0.5。

同时进行速度判断, 速度判断采用相邻帧中心点坐标位置判断。 $x_m > x_{m1}$, 目标速度向左; $x_m < x_{m1}$, 目标速度向右; $y_m > y_{m1}$, 目标速度向上; $y_m < y_{m1}$, 目标速度向下。

边界条件与速度条件严格对应, 满足条件的丢失轨迹被移除。被长时间遮挡的丢失轨迹不满足以上条件被保留。

基于速度判别的丢失轨迹移除算法流程如图 11 所示。其中, s 为丢失轨迹集合 S_{lost} 中的一个元素, 即一个丢失轨迹。 $s.x_1, s.y_1$ 为 $t+1$ 时刻轨迹 s 对应目标框的横坐标、纵坐标, $s.w_1, s.h_1$ 为 $t+1$ 时刻轨迹 s 的宽、高。 $s.x_{m1}, s.y_{m1}$ 为 $t+1$ 时刻轨迹 s 的中心横坐标、纵坐标。 $s.x_m, s.y_m$ 为 t 时刻轨迹 s 的中心横坐标、纵坐标。

(2) 基于历史位置的匹配算法。

该算法基于路侧交通场景的特点, 用于第 3 次匹配, 匹配的双方为高置信度物体匹配失败的检测框、低置信度物体匹配失败的检测框与低置信度物体匹配失败的轨迹。两者之间的相似度度量为运动

输入: 丢失轨迹集合 S_{lost}

输出: 保留轨迹集合 S_{remain}

```

1: for  $s$  in  $S_{\text{lost}}$ 
2: if  $s.y_1 - b \times s.h_1 < 0$  and  $s.y_m > t.y_{m1}$  do
3: delete  $s$ 
4: if  $s.x_1 - b \times s.w_1 < 0$  and  $s.x_m > s.x_{m1}$  do
5: delete  $s$ 
6: if  $s.x_1 + s.w_1 + b \times s.w_1 > W$  and  $s.x_m < s.x_{m1}$  do
7: delete  $s$ 
8: if  $s.y_1 + s.h_1 + b \times s.h_1 > H$  and  $s.y_m < s.y_{m1}$  do
9: delete  $s$ 
10: 将保留的丢失轨迹赋值给  $S_{\text{remain}}$ :  $S_{\text{remain}} \leftarrow S_{\text{lost}}$ 
11: return  $S_{\text{remain}}$ 

```

图 11 速度判别的丢失轨迹移除算法流程

Fig.11 Algorithm flow for removing lost trajectories in speed discrimination

相似度, 即目标检测框与轨迹历史边界框的 IoU (intersection over union)。

历史边界框信息在轨迹经过卡尔曼滤波预测之前获取, 是上一时刻经过目标检测结果修正的轨迹位置。对于丢失轨迹, 由于位置信息没有经过修正, 存在偏差。即轨迹丢失以后, 历史位置不再更新, 保持不变。

图 12 展示引入基于速度判别的丢失轨迹移除算法和基于历史位置的匹配算法后, 追踪器的性能。图 12a 为引入前效果, 第二辆白车遮挡前的身份为 14。在红灯期间受到长时间严重遮挡, 遮挡时间达到 500 帧, 轨迹一直处于丢失状态, 在 840 帧, 白车被重新检测到, 身份发生切换, 变为 39。图 12b 为引入后效果。长时间遮挡之前, 第二辆白车身份为 5, 长时间遮挡后, 白车重新找到, 身份依旧为 5, 即轨迹原始身份恢复。

3.3 BIoU 运动相似度度量方式

BIoU 扩大轨迹边界框与目标检测框的匹配计算空间, 弥补卡尔曼滤波带来的预测偏差。现实场景中会出现目标对象突然转弯、提速, 导致卡尔曼滤波构建的匀速运动模型失效。经过卡尔曼滤波预测后, 轨迹预测位置与实际测量位置差距大, IoU 数值低, 关联失败, 轨迹中断。

BIoU 在不改变边界框的中心点位置、宽高比、形状的前提下, 扩大边界框面积, 如图 13 所示, 实线框为原始边界框, 虚线框为扩展边界框。它们拥有共同中心点。原始的边界框宽和高为 w, h 。扩展后的边界框宽和高为 w_b 和 h_b 。两者的对应关系为

$$b = \frac{w_b - w}{2w} = \frac{h_b - h}{2h} \quad (1)$$

式中: b 为扩充系数。

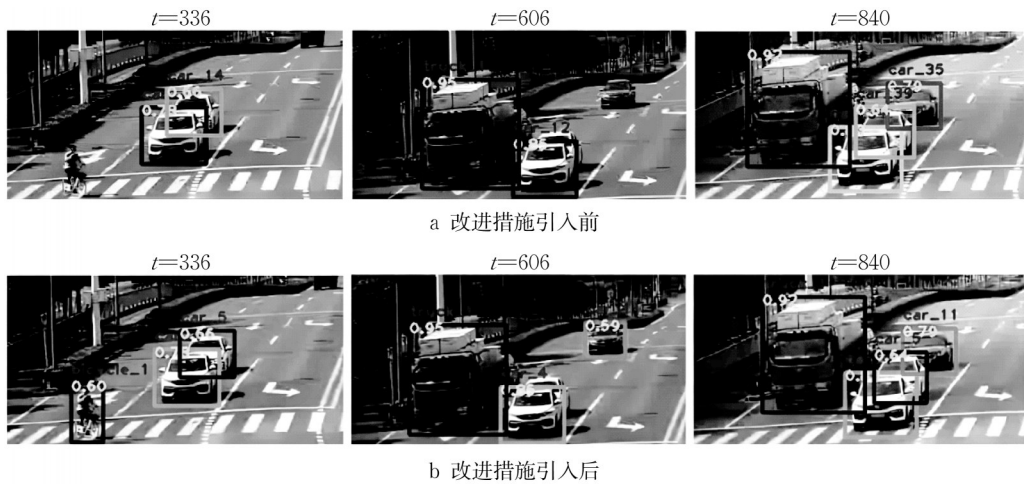


图12 丢失轨迹移除算法和历史位置匹配效果展示

Fig.12 Demonstration of lost trajectory removal algorithm and historical position matching effect

$o=(x, y, w, h)$ 表示原始的边界框, (x, y) 为左上角坐标, w, h 为边界框的宽、高。扩展后的边界框为 $o_b=(x-bw, y-bh, w+2bw, h+2bh)$ 。

图14展示IoU与BIoU的计算区域。实线框为原始的IoU计算区域,虚线框为BIoU计算区域。

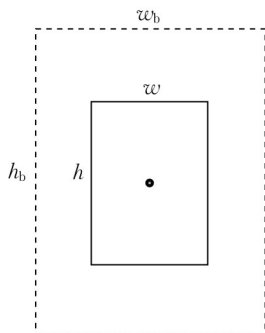


图13 BIoU展示

Fig.13 BIoU display

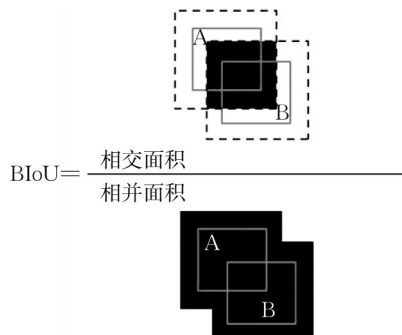


图14 IoU与BIoU计算区域

Fig.14 Computation domain of IoU and BIoU

BIoU也会造成匹配空间过度放大,产生错误关联。为缓解匹配空间过度放大,本算法在高置信度

物体匹配时选用小扩充系数,在低置信度物体匹配时选用大扩充系数。

图15展示引入BIoU之后,算法的性能。图15a为未引入BIoU的实验结果,由于摩托车在行驶过程中突然加速,运动预测模型失效,轨迹身份连续切换,23到24到25。图15b为引入BIoU后的实验结果。引入BIoU后,同样的摩托车,身份保持不变,保持为9,轨迹连续。

4 实验验证

4.1 实验平台

本实验的模型训练部署在百度飞桨AI Studio高级GPU服务器,核心数量4,线程数量8;显卡为Tesla V100,显存大小为32 GB。神经网络训练使用PaddlePaddle实现。

本实验的算法验证使用AMD Ryzen 7 6800H with Radeon Graphics,核心数量8,线程数量16;显卡为NVIDIA GeForce RTX 3060 Laptop GPU, 6 GB显存;算法验证使用PaddlePaddle实现。

4.2 实验数据

4.2.1 UA-DETRAC多目标追踪数据集

UA-DETRAC^[19]是一个路侧视角下的多目标追踪数据集,该数据集由北京和天津的24个不同位置的佳能EOS 550D相机采集。采集时长达到10 h,视频分辨率为960×540。UA-DETRAC包含训练集与测试集两部分,其中训练集84 000张图片,5 900个真值轨迹。测试集56 000张图片,2 337个真值轨迹。标注4种类别的目标对象:轿车、面包车、公交车和其他。

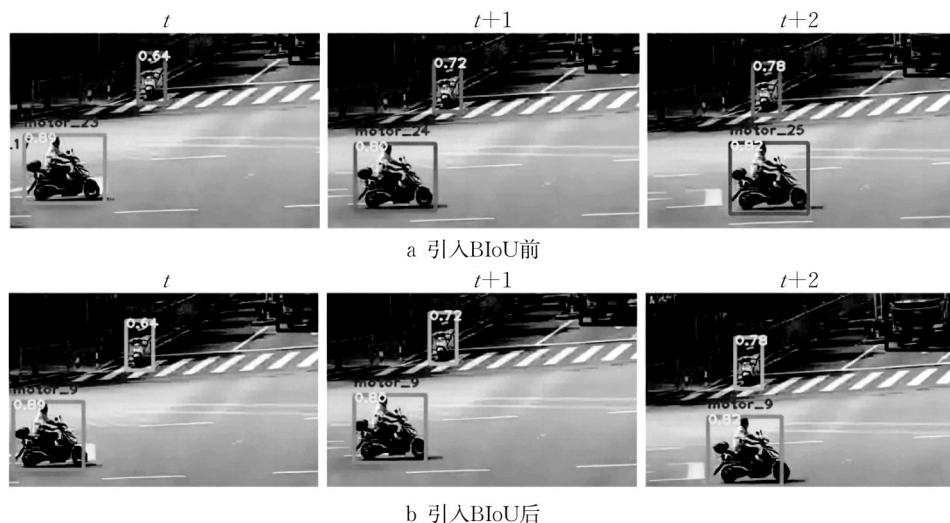


图 15 BIoU 效果展示

Fig.15 Demonstration of BIoU performance

4.2.2 真实场景路侧数据

本文在国家智能网联汽车(上海)试点示范区开放道路采集真实场景数据,包含 4 个不同路口场景,2 个时间段,视频总长度达到 13.81 min。将采集数据拆分为训练数据与测试数据,训练数据 6.92 min,测试数据 6.89 min。

对采集的数据,按照国际公开数据集标准^[20]人工标注。数据集标注对象为路侧交通场景下的行人、车辆,主要为运动物体。标注的类别如表 1 所示,总共标注 12 类物体。ignored region 为忽略区域;pedestrian 为站立或行走的人,其他状态的人为 people 类;bicycle 为自行车;car 为汽车;van 为面包车;truck 为卡车;tricycle 为三轮车,包含三轮车上的;awning-tricycle 为带遮阳棚的三轮车;bus 为公交车;motor 为摩托车或者电动车,包含车上的人;others 为其他类别。

表 1 路侧数据标注类别

Tab.1 Annotation categories of roadside data

类别	ID	类别	ID
ignored region	0	truck	6
pedestrian	1	tricycle	7
people	2	awning-tricycle	8
bicycle	3	bus	9
car	4	motor	10
van	5	others	11

4.3 实验评估指标

实验使用的算法评估指标包括身份切换次数、多目标跟踪准确度、身份识别指标、中断、大部分跟踪、部分跟踪、大部分丢失。

身份切换(ID switch,IDS)次数为追踪器分配身

份发生错误变换的次数,每当轨迹身份发生一次错误变化,身份切换次数加 1。多目标跟踪准确度(multiple object tracking accuracy, MOTA)^[21]衡量追踪器的检测性能。身份识别指标主要用 IDF1 (identification F-Score)^[22]来衡量,IDF1 通过双射方式准确测量推理跟踪轨迹与真实轨迹之间的匹配情况。中断(fragmentation, FM)为轨迹中断次数,每一个真值轨迹中断并恢复,即真值轨迹丢失后又找到,称为一次中断。大部分跟踪(mostly tracked, MT)为真值轨迹被跟踪器跟踪时间达到生命周期的 80% 及以上。部分跟踪(partially tracked, PT)为真值轨迹被跟踪器跟踪时间达到生命周期的 20%~80%。大部分丢失(mostly most, ML)为真值轨迹被跟踪器跟踪时间小于生命周期的 20%。

追踪算法产生的大部分跟踪轨迹数目越多,大部分丢失轨迹数目越少,算法性能越好。

4.4 对比实验结果分析

本实验选取的对比算法有: SORT (使用 YOLOv3^[23]目标检测网络替换原始检测网络 Faster-RCNN^[24])、DeepSORT、ByteTrack、FairMOT。

本文首先在 UA-DETRAC 数据集上进行实验,验证本算法的先进性。然后在真实场景路侧数据进行对比实验,验证本算法的场景适用性。

4.4.1 UA-DETRAC 实验结果

为了与公开论文^[25]评估实验一致,各个算法的神经网络在 UA-DETRAC 训练集上的训练轮次为 30,学习率为 1.25×10^{-4} 。实验结果如表 2 所示。算法在 UA-DETRAC 数据集下, MOTA 为 54.6%, IDF1 为 73.0%, IDS 为 620, FM 为 4 318, MT 为 1

351,ML为191。IDF1、FM、MOTA指标排名第一,MT、ML指标排名第二,综合性能最佳。

4.4.2 真实场景实验结果

为保证实验的公平性,各个算法的神经网络在相同真实场景训练集上训练,训练轮次为100,训练策略一致。实验结果如表3所示。本算法在路侧场

景下,MOTA为75.3%,IDF1为86.5%,IDS为24,FM为320,MT为113,ML为5。各项指标均高于其他算法,本追踪算法的检测性能和轨迹维持能力均达到最佳。本算法推理速度达到13.16 FPS,实时性较two-shot系列算法有所提升。

表2 UA-DETRAC对比实验结果
Tab.2 UA-DETRAC comparison experimental results

算法	IDF1/%	IDS	FM	MT	PT	ML	MOTA/%
SORT	72.8	440	6 300	1 328	815	194	52.7
DeepSORT	70.8	1 073	4 479	1 315	765	257	51.3
ByteTrack	72.5	430	6 186	1 376	796	165	50.9
FairMOT	68.1	819	8 074	1 307	760	270	45.4
本文	73.0	620	4 318	1 351	795	191	54.6

表3 真实场景对比实验结果
Tab.3 Real-world scenario comparison experimental results

算法	IDF1/%	IDS	FM	MT	PT	ML	MOTA/%
SORT	84.1	40	432	98	24	10	71.7
DeepSORT	82.1	75	512	95	26	11	70.8
ByteTrack	83.5	26	325	104	22	6	71.2
FairMOT	83.4	139	613	100	26	6	74.1
本文	86.5	24	320	113	14	5	75.3

4.5 可视化结果

图16为本算法的可视化结果。从图16中可知,本算法可以对每个交通目标正确识别并追踪。



图16 可视化结果
Fig.16 Visualization results

5 结语

针对路侧多目标追踪算法普遍存在的问题,提出一种基于路侧摄像头的多目标追踪算法。在UA-DETRAC数据集与真实场景数据集上进行对比实验,结果表明:

(1)基于FairMOT的神经网络同时输出检测结

果和外观特征。算法在真实场景处理速度达到13.16 FPS,实时性效果增强。

(2)BYTE数据关联方式增强追踪器抗遮挡性能。在目标被遮挡条件下,实现追踪,轨迹保持连续。基于速度判别的丢失轨迹移除算法和基于历史位置的匹配算法实现长时间被遮挡轨迹身份恢复。

(3)BIoU扩大轨迹边界框与目标检测框之间的匹配空间,弥补卡尔曼滤波预测误差,有效减少身份切换。

(4)在公开数据集与真实路侧交通场景下,本文算法均优于其他先进的多目标追踪算法。

在未来的研究工作中,将会在上述基于路侧摄像头的多目标跟踪算法的基础上,进一步研究路侧场景跨摄像头多目标追踪,并和摄像机标定技术结合,使追踪器输出轨迹的真实世界坐标。

作者贡献声明:

王平:系统设计思路。
姚宇阳:算法设计与实现,仿真分析与论文的撰写。
王新红:论文润色。

参考文献:

[1] 刘睿健.“协同”有道,“无人”驾成!——车路协同自动驾驶系统发展漫谈[J]. 中国交通信息化, 2020(10):18.
LIU Ruijian. “Collaboration” is the way, “driverless” is the

- goal! – a discussion on the development of collaborative autonomous driving systems [J]. China Transportation Informatization, 2020(10): 18.
- [2] 冉斌, 谭华春, 张健, 等. 智能网联交通技术发展现状及趋势[J]. 汽车安全与节能学报, 2018, 9(2): 119.
RAN Bin, TAN Huachun, ZHANG Jian, *et al.* Development status and trends of intelligent connected transportation technology [J]. Journal of Automotive Safety and Energy Conservation, 2018, 9(2): 119.
- [3] 张新钰, 邹镇洪, 李志伟, 等. 面向自动驾驶目标检测的深度多模态融合技术[J]. 智能系统学报, 2020, 15(4): 758.
ZHANG Xinyu, ZOU Zhenhong, LI Zhiwei, *et al.* Deep multimodal fusion technology for autonomous driving object detection[J]. Journal of Intelligent Systems, 2020, 15(4): 758.
- [4] COMANICIU D, RAMESH V, MEER P. Real-time tracking of non-rigid objects using mean shift[C]//IEEE Conference on Computer Vision and Pattern Recognition. Hilton Head: IEEE, 2000: 142-149.
- [5] OKUMA K, TALEGHANI A, DEFREITAS N, *et al.* A boosted particle filter: multitarget detection and tracking[C]//European Conference on Computer Vision. Berlin, Springer: 2004: 28-39.
- [6] KALMAN R E. A new approach to linear filtering and prediction problems [J]. Trans ASME-J Basic Eng, 1960, 82(1): 35.
- [7] WOJKE N, BEWLEY A, PAULUS D. Simple online and realtime tracking with a deep association metric [C]//IEEE International Conference on Image Processing. Beijing: IEEE, 2017: 3645-3649.
- [8] BEWLEY A, GE Z, OTT L, *et al.* Simple online and realtime tracking [C]//IEEE International Conference on Image Processing. Phoenix: IEEE, 2016: 3464-3468.
- [9] AHARON N, ORFAIG R, BOBROVSKY. BoT-SORT: robust associations multi-pedestrian tracking[J/OL]. [2022-07-07]. <https://arxiv.org/abs/2206.14651v2>.
- [10] YUN D, CHENG Z, YANG S, *et al.* StrongSORT: make deepsort great again [J]. IEEE Transactions on Multimedia, 2023, 25(1941): 8725.
- [11] DU Y, WAN J, ZHAO Y, *et al.* GiauTracker: a comprehensive framework for mc-mot with global information and optimizing strategies in visdrone 2021 [C]//IEEE International Conference on Computer Vision. Montreal: IEEE, 2021: 2809-2819.
- [12] WILLIAMS C, RASMUSSEN C. Gaussian processes for regression [C]//8th International Conference on Neural Information Processing Systems. Denver: MIT, 1995: 514 – 520.
- [13] WANG Z, ZHENG L, LIU Y, *et al.* Towards real-time multi-object tracking [C]//16th European Conference on Computer Vision. Glasgow: Springer, 2020: 107-122.
- [14] HANG 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(11): 3069.
- [15] KAIWEN D, SONG B, XIE L, *et al.* CenterNet: keypoint triplets for object detection[C]//IEEE International Conference on Computer Vision. Seoul: IEEE, 2019: 6569-6578.
- [16] ZHANG Y, SUN P, JIANG Y. Bytetrack: multi-object tracking by associating every detection box[C]//17th European Conference on Computer Vision. Tel Aviv: Springer, 2022: 1-21.
- [17] FAN Y, SHIGEYUKI O, SHOICHI M, *et al.* C-BIoU: hard to track objects with irregular motions and similar appearances? Make it easier by buffering the matching space [C]//IEEE/CVF Winter Conference on Applications of Computer Vision. Waikoloa: IEEE, 2023: 4799-4808.
- [18] YU F, WANG D, SHELHAMER E, *et al.* Deep layer aggregation [C]//IEEE International Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 3865-3875.
- [19] WEN Longyin, DU Dawei, CAI Zhaowei, *et al.* UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking [DB/OL]. [2020-01-24]. <https://arxiv.org/abs/1511.04136>.
- [20] MILAN A, LEAL-TAIX L, REID I, *et al.* Mot16: a benchmark for multi-object tracking [DB/OL]. [2016-05-03]. <https://arxiv.org/abs/1603.00831>.
- [21] BERNARDIN K, STIEFELHAGEN R. Evaluating multiple object tracking performance: the clear mot metrics [J]. EURASIP Journal on Image and Video Processing, 2008(2008): 1.
- [22] LUITEN J, OSEP A, DENDORFER P, *et al.* Hota: a higher order metric for evaluating multi-object tracking [J]. International Journal of Computer Vision, 2021, 129(11): 548.
- [23] REDMON J, FARHADI A. Yolov3: an incremental improvement. [J/OL]. [2018-04-08]. <https://arxiv.org/abs/1804.02767>, 2018.
- [24] 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.
- [25] YU Chaoshun, YANG Jianxi, JIANG Shixin, *et al.* ECCNet: efficient chained centre network for real-time multi-category vehicle tracking and vehicle speed estimation [J/OL]. [2022-06-06]. <https://trid.trb.org/view/2043508>.