

基于视频的交叉口目标轨迹自动采集

杨 轸, 张秋晨

(同济大学 道路与交通工程教育部重点实验室, 上海 201804)

摘要: 提出了一种基于视频的交叉口内目标运动轨迹自动采集方法. 首先利用 ViBe(Visual Background Extractor)算法提取目标前景, 然后提出了基于光流的目标跟踪 OPC(object-point-contour)算法, 最后基于透视变换得到目标的真实轨迹和运动参数. 方法可采集交叉口内所有交通对象的类别、轨迹、速度与加速度信息, 并对目标停滞与遮挡现象有较好的跟踪稳定性. 经检验, 该方法对机动车、非机动车和行人的轨迹提取准确率分别为 88.89%、86.00% 和 83.33%, 速度提取准确率为 91.71%, 为交叉口管理与安全研究提供一种视频处理和数据采集手段.

关键词: 运动轨迹; 目标跟踪; 视频处理; 光流

中图分类号: U495

文献标志码: A

A Video-based Framework for Automated Trajectory Collection at Intersections

YANG Zhen, ZHANG Qiuchen

(Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai 201804, China)

Abstract: A video-based framework was presented to collect trajectories of objects at intersections. Firstly, a modified ViBe algorithm was used to extract the foreground of moving objects. Then, an OPC (Object-Point-Contour) matching approach was developed for paring, tracking and generating image trajectories. Finally, real world trajectories were obtained through perspective transformation so that velocity and acceleration could be estimated. The framework was stable to occlusion and stagnation. The accuracy of 88.89%, 86.00%, and 83.33% were obtained for vehicles, bicycles and pedestrians counts and 91.71% for velocity estimation. This automatic system provides a solution for data collection and video analysis for intersection safety management and research.

Key words: trajectory; video tracking; video analysis; optical flow

城市交叉口是城市道路的重要结点, 交叉口内各种机动车流、非机动车流与行人流相互交织, 与普通路段相比更容易诱发交通事故. 据国家统计局统计, 2015 年我国共发生交通事故约 18.8 万起, 造成约 5.8 万人伤亡, 带来直接经济损失达上亿元^[1], 其中发生在交叉口的事故占据了 30%, 交叉口安全问题仍亟待解决^[2].

研究交叉口安全问题需要观测大量数据, 目前常见的数据采集方法有人工法、线圈法、GPS 法和视频法, 其中视频法由于信息量大、可溯源、鲁棒性强等特点, 应用最为广泛. 视频法根据其自动化程度可分为人工处理法和自动处理法.

人工处理法指通过人为观察或点选轨迹获取交通对象的数量、速度或行为等信息. 慈玉生等^[3]通过人工记录获取交叉口过街人数的分布情况, 并基于该分布提出了机动车综合延误计算模型. William 等^[4]以行人过街的起终点为特征点, 统计了香港不同过街设施处行人的过街速度与过街流量. Su 等^[5]利用视频数据对右转车辆与行人的交通冲突进行观测, 并提出了基于冲突概率的右转专用相位设置方法. 张晓玮^[6]选用 3 名经过专业培训的观察员观测交叉口冲突, 并通过人工标点的方法绘制冲突轨迹, 选取 TTC (time to collision)、PET (post encroachment time) 和 DST (deceleration to safety) 为评价指标对冲突严重程度进行划分. 唐克双等^[7]使用坐标处理软件 George2.1 通过点击视频中目标位置由软件自动完成轨迹生成与真实坐标转换, 获得进口道车辆从绿闪前 2s 至红灯亮起时的时空轨迹, 实现了交通参数的半人工采集, 并基于该轨迹构建了危险驾驶行为预测模型. 综上, 人工法处理交通视频的可靠性较高, 数据采集难度较低, 但需花费较多人力和时间, 且采集的样本量较少, 无法进行大规模的数据分析.

收稿日期: 2018-01-26

基金项目: 国家重点研发计划(2017YFC0803902)

第一作者: 杨 轸(1974—), 男, 副研究员, 博士生导师, 工学博士, 主要研究方向为道路安全. E-mail: yangzhen5719@tongji.edu.cn

随着近年来计算机视觉技术的发展,越来越多的学者尝试采用自动检测的方法处理交通视频.自动处理法指利用图像处理算法自动捕捉视频中的交通对象,并保存其时空轨迹以供后续研究^[8].自动处理法具有采集信息量大、效率高的优点,且随着硬件设备与处理算法的不断发展,处理的准确率与稳定性也在不断提升. Sayed 等^[9]利用视频自动提取交叉口车辆轨迹,并提出了基于 HMM(Hidden Markov Model)的交通冲突预测方法. 王俊骅等^[10]结合背景差法和区域跟踪法提取机动车辆在交叉口的行驶轨迹并对交通冲突进行分析. Guo 等^[11]等利用光流法获取交叉口内各交通对象的轨迹并根据交通冲突评价设置外侧左转车道对交叉口安全的影响.

上述研究虽然都实现了对交通视频的自动化处理,但仍存在着提取目标类型单一或轨迹连续性不强的局限性,且不能很好地处理目标遮挡与停滞现象. 如文献^[9]仅提取了交叉口车辆的行驶轨迹,忽视了行人和非机动车对交叉口安全的影响,文献^[10]提出的视频处理方法对于行进中车辆的跟踪效果较好,但对停滞车辆的跟踪效果一般.

针对上述局限性,提出一种交叉口视频处理方法,可采集交叉口内所有交通对象的时空轨迹,并对目标停滞与遮挡现象有一定的鲁棒性.

1 前景目标提取

算法整体流程为:

(1) 前景提取. 基于 ViBe (visual background extractor)算法获取视频中运动目标前景,并进行形态学处理.

(2) 目标跟踪. 提取目标特征点,基于 KLT (Kanade Lucas Tomasi)光流提出新型目标跟踪 OPC(object-point-contour)算法,在视频中提取目标图像轨迹.

(3) 轨迹后处理. 对图像轨迹进行分离修正,根据目标活动区域与特征信息进行交通对象分类,并将图像坐标转换为现实坐标.

(4) 结果输出. 根据目标时空轨迹点计算速度、加速度等运动参数.

交叉口内人车混行,背景复杂,直接在全图中检测交通对象耗时严重,准确率也较低,常用方法是先通过背景建模提取前景区域,然后再对前景图像进行目标检测与跟踪. 目前常见的前景提取方法有帧差法^[12]和混合高斯法^[10]. 帧差法对相邻 2 帧或 3 帧

进行差分,保留差异较大的像素作为前景,计算量小且对运动变化敏感,但目标内部留下的空洞部分形成鬼影,为后续目标跟踪带来诸多困难. 混合高斯模型(GMM)为每个像素点构建单个或多个高斯分布模型,通过计算概率确定像素点是否属于前景. GMM的优点是可适应背景轻微变化、对光照与景物扰动有较好稳定性,缺点是计算效率低、对高清视频处理速度较差且对停滞目标的提取效果一般,当目标颜色与背景颜色接近时也会存在较大误差^[13].

ViBe^[14]是一种基于样本一致性的背景建模方法,采用随机更新和邻域更新,保证了样本值平滑的生命周期,使检测结果更为准确. ViBe 作为一种新型的背景建模方法,在交通视频处理领域应用尚不广泛. 与 GMM 相比,ViBe 对背景扰动的稳定性较弱,但可较好地提取停滞目标和接近背景颜色目标的前景. 由于视频处理系统面向交叉口全对象,输入视频多为俯视拍摄,树木、电线等扰动较大的景物图像一般不会出现在交叉口内部,因此可通过设置感兴趣区从而较方便地排除背景扰动影响. 另一方面,交叉口内行人与非机动车行为模式复杂,存在较多的停滞现象,且人群着装各异,易与地面背景产生混淆,这些因素使得 ViBe 十分契合提取前景的需求.

ViBe 算法根据第 1 帧图像进行模型初始化,故第 1 帧中的前景目标会被误认为背景,导致目标离开该区域后会出现“虚假前景”现象(表 1),为后续目标检测带来干扰. 为消除“虚假前景”现象,采用 GMM 对 ViBe 进行改进. 首先使用 GMM 训练视频前 500 帧图像,得到初始背景;然后以背景作为 ViBe 的第 1 帧输入,重新训练背景并提取前景图像;最后对前景图像中的感兴趣区进行滤波和形态学处理,以消除噪声并增强前景目标连通性.

表 1 对比了使用 GMM、ViBe 和本文使用的改进 ViBe 算法提取前景的效果. 可以看到,GMM 提取的前景轮廓与目标相比有较为明显的缩水,不利于后续特征点提取;ViBe 由于存在“虚假前景”,误将路面检测为运动目标;相比之下,本文方法可较好地提取交叉口内全交通对象的前景,为后续的目标跟踪提供准确的前景输入.

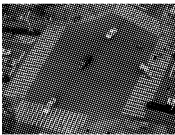
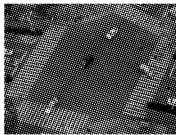
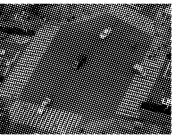
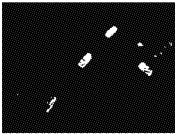
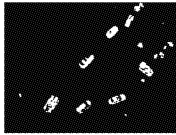
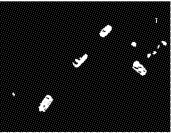
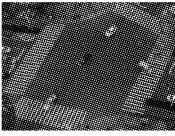
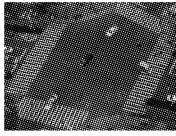
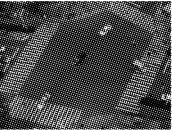
2 基于 KLT 光流的目标跟踪

获得交叉口前景图像后,对前景中的目标进行识别和跟踪. 利用连通性可将前景图像划分成诸多前景轮廓 C , 1 个轮廓可能包含 1 个或多个目标 O .

目标跟踪要做的就是将每一帧的轮廓与目标通过某种规则匹配起来,并对同一目标的轮廓变化进行动态更新. KLT 法^[15]是一种经典的光流估算差分方法,包括特征角点检测和光流匹配两部分,特征点是指图像灰度值发生剧烈变化的点或者在图像边缘上曲率较大的点,稳定的特征点需具有良好的旋转不变性和缩放不变性. 光流匹配的原理是假定特征点的局部邻域内光流恒定不变,利用最小二乘对特征点邻域内所有像素求解基本光流方程. KLT 法通过综合多个附近像素的信息,可以有效解决光流方程固有的模糊性问题,并且图像噪点和光照变化具有较强稳定性. 基于 KLT 光流,以特征角点 P 为媒介,可实现轮廓 C 与目标 O 的动态匹配. 匹配过程包括目标初始化、特征点更新、轮廓-目标匹配、目标属性更新 4 个步骤.

表 1 GMM、ViBe、本文方法提取前景效果对比

Tab.1 Foreground extraction using GMM、ViBe and method in this work

图像	GMM 方法	ViBe 方法	本文方法
原始			
前景			
输出			

2.1 目标初始化

设第 n 帧图像中的特征点为集合 P_n .

$$P_n = \{p_{n,i} \mid 1, \dots, I\} \quad (1)$$

式中: P_n 表示第 n 帧的特征点集合; $p_{n,i}$ 表示第 n 帧的第 i 个特征点; I 表示单帧特征点总数,出于特征点性能的考虑,设置 $\max(I) = 500$,并根据交叉口大小适当调整.

对前景轮廓依次编号,并分配给轮廓内的所有特征点,同一轮廓内的特征点拥有相同的轮廓编号,即

$$C_n = \{c_{n,j} \mid j = 1, \dots, J\} \quad (2)$$

$$\forall p_{n,i} \in c_{n,j}, C(p_{n,i}) = j \quad (3)$$

式中: C_n 表示第 n 帧的轮廓集合; $c_{n,j}$ 表示第 n 帧的第 j 个轮廓; $C(p_{n,i})$ 表示特征点 $p_{n,i}$ 所属的轮廓编号; J 表示单帧轮廓总数.

当 $n=1$ 时,系统需进行目标初始化,目标对象包含 5 个属性,分别为编号、时间序列、图像中心序列、图像面积和特征点集合,即

$$O_k = (k, F_k, C_k, A_k, \{P_n(O_k) \mid n \in f_k\}) \quad (4)$$

式中: O_k 表示第 k 个目标的属性集合, k 初始化后不可改变; F_k 、 C_k 、 A_k 分别表示目标 O_k 出现的帧数、每帧中心位置和每帧面积,为动态数组,随后续跟踪增加长度,且三者长度一致; $P_n(O_k)$ 表示目标 O_k 在第 n 帧的特征点集合. 由于第 1 帧默认不存在遮挡与分离现象,因此目标与轮廓编号呈一一对应关系,得到目标在第 1 帧的特征点集合如式(5):

$$P_1(O_k) = \{p_{1,i} \mid C(p_{1,i}) = k\} \quad (5)$$

之后便可根据目标对应的前景轮廓计算其中心位置与面积,并与其特征点一并保存,完成目标初始化.

2.2 特征点更新

从第 2 帧开始,每帧图像都以前一帧的特征点作为输入,并利用 KLT 光流进行跟踪,以得到更新后的特征点. 为保证光流特征点的稳定性,通过反向检测去除前后匹配不一致的特征点,并添加由该帧图像新检测出的特征点 $p_{n+1,i}$,如式(6)、(7)、(8):

$$p_{n+1,i} = K(p_{n,i}) \quad (6)$$

$$p'_{n,i} = K^{-1}(p_{n+1,i}) \quad (7)$$

$$P_{n+1} = \{p_{n+1,i} \mid D(p_{n,i}, p'_{n,i}) < t_k\} \quad (8)$$

式中: $K(p)$ 表示对点 p 进行正向光流跟踪; $K^{-1}(p)$ 表示对点 p 进行逆向光流跟踪; p' 为点 p 反向检测后的镜像点; $D(p, p')$ 表示两点之间的距离; t_k 为 KLT 反向检测的稳定阈值, t_k 越小,两帧间的匹配精度越高,但在视频分辨率较低时易出现目标丢失情况.

2.3 OPC 匹配

轮廓与目标匹配是目标跟踪的关键所在,其匹配准确率直接影响跟踪结果. 提出匹配算法 OPC,以特征点 P 为媒介对每帧生成的前景轮廓与之前跟踪的目标进行匹配. 对于第 n ($n > 1$) 帧图像,获得前 1 帧图像中各目标包含的特征点,得到特征点与目标的对应关系. 然后更新特征点,依据对应关系将新特征点分配至各个目标. 最后根据轮廓内特征点对应的目标,判断轮廓与目标的匹配情况. 典型 OPC 匹配仅以 1 个特征点作为媒介,匹配原理如图 1 所示.

在实际进行检测时,目标往往具有多个特征点,这些特征点可能属于零个、1 个或多个前景轮廓,同样地,多个目标的特征点也可能汇聚在同一轮廓内. 根据 $C-O$ 的映射关系可将匹配过程分为如下 5 种

情况.

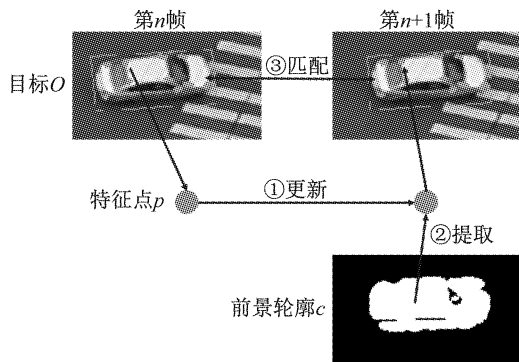


图 1 一次 OPC 匹配

Fig.1 One OPC pairing

2.3.1 目标与轮廓一一对应

如表 2 所示,目标所有特征点更新后均处在同一轮廓内,且轮廓内无其他特征点,是跟踪单个交通对象时最理想情况,前后帧的特征点与目标按式(9)

表 2 目标与轮廓一一对应情形

Tab.2 One object matches one contour

帧数	目标	轮廓
n		
$n+1$		

表 4 单目标对应多轮廓情形

Tab.4 One object matches multiple contours

帧数	情形 1		情形 2	
	目标	轮廓	目标	轮廓
n				
$n+1$				

2.3.4 多目标对应单轮廓

如表 5 所示,多个目标的特征点全部集中在一个前景轮廓内,这是目标跟踪过程中最常见的一种情况.其形成原因有 2 种:① 特征点转移,即一个目标的特征点由于某种原因转移至另一个目标上,如

进行匹配:

$$k(p_{n+1,i}) = k(p_{n,i}) \quad (9)$$

式中: $k(p)$ 表示特征点 p 所属的目标编号.

2.3.2 目标无对应轮廓

如表 3 所示,目标所有特征点均未出现在前景中,说明目标已离开交叉口,中断对目标的跟踪.

表 3 目标无对应轮廓情形

Tab.3 Object matches no contour

帧数	目标	轮廓
n		
$n+1$		

2.3.3 单目标对应多轮廓

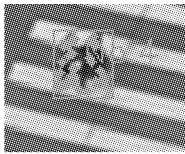
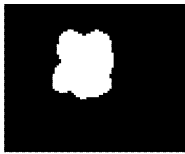
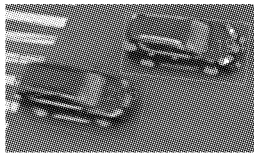


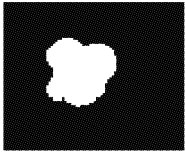
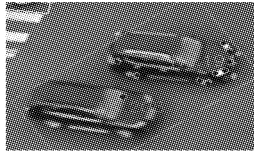

如表 4 所示,目标的特征点存在于多个前景轮廓内.出现该情况有 2 种可能:① 同一目标的前景提取不完全,跟踪大面积目标时可能出现,比如深色车辆车顶颜色与路面过于接近时,ViBe 将同一辆车分成多了个前景(表 4 情形 1);② 目标结群进入交叉口,且在交叉口内部分离(情形 2).若要判断分离现象属于哪种情况需得知分离后的轨迹趋势,轨迹一致则分离轨迹属同一目标,反之则属于不同目标.因此,单目标对应多轮廓情况在跟踪过程中暂不考虑,待整体轨迹生成后再做处理(见 3.1 节).

表 5 情形 1. 由于道路环境复杂,图像中相似特征较多,一些弱特征点难以保证持续稳定的匹配,属于 KLT 无法避免的系统误差;② 目标轨迹交汇产生遮挡,如表 5 情形 2. 此时可利用动态面积分配法确定前景轮廓内包含的目标数量:首先根据 OPC 匹配算

法确定待选目标,然后按目标包含的特征点数量多少将待选目标降序排列,最后循环对比目标面积和

剩余轮廓面积直至二者差超过阈值,此时所有遍历过的目标即为轮廓实际包含目标.

表 5 多目标对应单轮廓情形
Tab.5 Multiple objects match one contour

帧数	情形 1		情形 2	
	目标	轮廓	目标	轮廓
n				
$n+1$				

2.3.5 多目标对应多轮廓

该种情况较为罕见,多为特征点转移和前景提取不完全同时发生导致,因此可将其分解为多个多目标对应单轮廓的情况,利用动态面积分配法进行匹配.

2.4 目标属性更新

待上述匹配过程完成后,需对匹配后的目标属性进行更新.按轮廓包含目标数是否唯一可分为 2 种情况.

2.4.1 多目标对应多轮廓

此时轮廓内所有新旧特征点均重新分配给该目标,根据轮廓中心和面积更新目标中心和面积.

2.4.2 轮廓包含多个目标

若特征点所属目标仍在轮廓内,则仍分配给该目标;若不在轮廓内或为新检测的特征点,则利用动态规划思想按最小距离进行分配,即分配给与其距离最近的特征点所属目标,若不存在,则寻找次近特征点的所属目标,如此递归直至所有特征点均获得目标.此情况下目标面积不做更新,目标中心位置根据前 2 帧运动状态推演得到.

至此完成对一帧图像内所有目标的跟踪,以目标属性作为下一帧的输入,重复 2.2 至 2.4 节的步骤直到视频结束,即可得到交叉口全对象的初始轨迹,目标跟踪界面如图 2 所示.

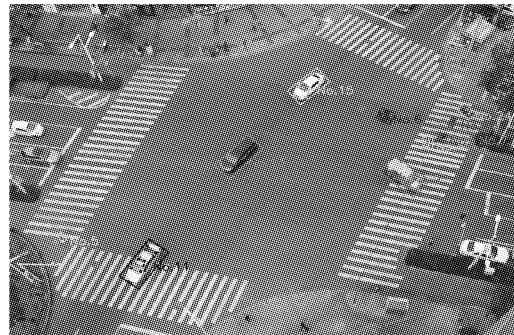


图 2 目标跟踪界面

Fig.2 Objects tracking

为:提取分离后的轨迹,并对齐起始点坐标.设目标 O_k 的轨迹 $T_k = \{t_{k,l} | l \in 1, \dots, N(k)\}$,其中 $N(k)$ 表示轨迹 T_k 的长度; T_k 中第 l 个坐标点 $t_{k,l} = (x_{k,l}, y_{k,l})$, x, y 表示点 t 的横、纵坐标.任选 2 条轨迹 T_{k1} 和 T_{k2} 使用 LCSS(longest common subsequence)算法^[16]计算二者相似长度 $L(T_{k1}, T_{k2})$.

$$L(T_{k1}, T_{k2}) = \begin{cases} 0, & N(k1) = N(k2) = 0 \\ 1 + L(H(T_{k1}), H(T_{k2})), & D(t_{k1, N(k1)}, t_{k2, N(k2)}) < \epsilon \\ \max(L(H(T_{k1}), T_{k1}), L(T_{k2}, H(T_{k2}))), & \text{其他} \end{cases} \quad (10)$$

式中: $H(T_k) = \{t_{k,l} | k \in 1, \dots, N(T_i) - 1\}$,表示轨迹 T_k 去除末尾点后的子轨迹; ϵ 为相似阈值,默认为真实坐标系下 1.6 m(一般小汽车宽度)对应的图像距离,需根据图像分辨率和拍摄高度进行计算.

由于各目标的轨迹长度不同,为统一标准,计算轨迹相似比 $S(T_{k1}, T_{k2})$ 以评价轨迹间的相似性, $S(T_{k1}, T_{k2}) = L(T_{k1}, T_{k2}) / \min(N(k1), N(k2))$.若两轨迹相似比小,则认为发生了多目标轨迹分离,将原轨迹分为 2 条独立轨迹保存;反之,则认为发生同

3 轨迹后处理

3.1 轨迹分离

当目标结群进入交叉口时会产生轨迹分离现象,需根据分离轨迹的趋势进行后处理.处理方法

目标前景轮廓分离,任取其中一条轨迹保存.

3.2 轨迹修正

初始轨迹跟踪的是目标的图像中心,存在一定误差,相比而言特征点的稳定性更强,因此可用特征点对初始轨迹进行修正,即保证第 1 帧坐标不变的情况下,利用特征点位移代替中点位移,根据是否存在轨迹分离现象修正过程可分为 2 种情况:

(1)若存在轨迹分离现象,则无法得知其特征点属于分离后的哪条轨迹,因此使用每帧与分离后轨迹最接近的特征点进行修正.

(2)对其他轨迹而言,特征点跟踪时间越长就说明其越稳定,因此以第 1 帧为起始,选取最长特征点序列进行修正,再以修正结束时刻为起点,循环此过程直至所有坐标修正完毕.

3.3 坐标转换

目前得到的所有轨迹均处于图像坐标系下,为计算目标的真实交通参数,还需要将图像坐标转换为地面坐标.不考虑边缘畸变的情况下,该问题可简化视平面投影问题^[6],并通过透视变换进行求解,变换公式为

$$\begin{cases} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \end{cases} \quad (11)$$

$$\begin{cases} u' = \frac{u}{w} = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + a_{33}} \\ v' = \frac{v}{w} = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + a_{33}} \end{cases} \quad (12)$$

式中: (x, y) 为图像坐标; (u', v') 为真实坐标; u, v, w 为透视变换计算的中间量; \mathbf{A} 为转换矩阵,其元素 $a_{pq}(p, q=1, 2, 3)$ 可通过代入 4 组标定点求解^[6].之后对所有图像坐标点应用式(12)可得到目标真实轨迹(图 3).使用局部加权回归法(lowess)对轨迹进行平滑处理,根据平滑后的时空轨迹即可计算目标速度,图 4 显示了图 3 目标的速度和加速度变化情况,

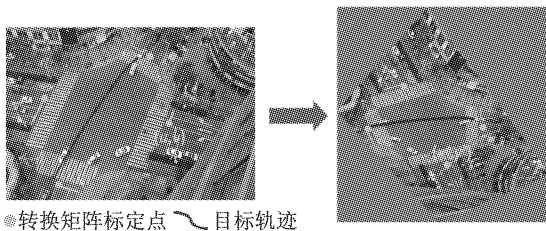


图 3 透视变换示意

Fig.3 Perspective transformation

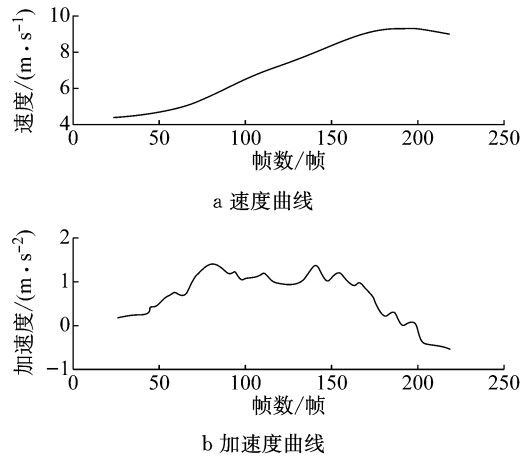


图 4 目标速度与加速度变化

Fig.4 Velocity and acceleration of tracking object

可较为明显地看出目标启动—提速—稳速的行驶过程.

3.4 目标分类

根据目标的轨迹、速度和图像面积等属性判断目标属于机动车、非机动车或行人.首先考虑目标的一般交通行为,即机动车仅从机动车道驶出驶入,非机动车仅从非机动车道驶出驶入,行人仅在人行道活动,此时可根据轨迹起终点和分布情况判断目标类型,如图 5 所示.

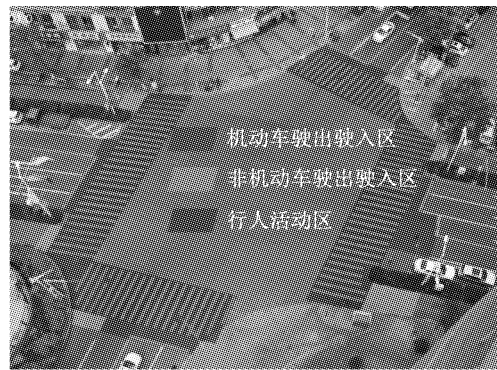


图 5 一般交通行为下目标分类

Fig.5 Object classification under normal behaviors

但现实中并非所有交通对象都按规则行驶,因此还需建立规则对非一般行为产生的轨迹进行分类.对行人而言,由跟踪特征点生成的轨迹往往存在较大波动^[17],当特征点位于手、脚等位置上时波动更加明显,故可通过计算平滑前后轨迹的差异性辅以速度特征判别行人目标.

$$d(O_k) = \sum_l (L_o(x_{k,l}) - x_{k,l})^2 + \sum_l (L_o(y_{k,l}) - y_{k,l})^2 \quad (13)$$

式中: $d(O_k)$ 表示轨迹平滑前后的差异性特征值; $L_o(x), L_o(y)$ 分别表示平滑后的 x, y 坐标.

当 $d(O_k)$ 小于某阈值且速度小于 $3 \text{ m} \cdot \text{s}^{-1}$ 时,目标判定为行人.对非机动车和机动车而言,二者轨迹均较为稳定,且在交叉口内的速度差异性也不明显,故考虑通过图像面积对二者进行分辨,即面积小于一定阈值时目标判定为非机动车,否则为机动车.上述 $d(O_k)$ 阈值与面积阈值需根据视频拍摄角度和高度进行标定.

整体轨迹后处理的流程如图 6 所示,3 条轨迹分别代表机动车、非机动车和行人.

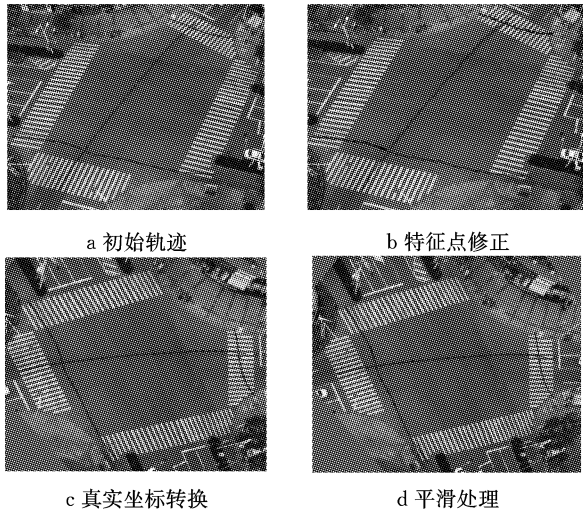


图 6 轨迹后处理

Fig.6 Trajectory post-process

4 案例分析

4.1 前期准备

输入视频的采集地点为上海市嘉定区新源路与泽普路交叉口(图 7). 视频分辨率为 $1\ 920 \times 1\ 080$, 时长 $5 \text{ min } 52 \text{ s}$, 帧率 $30 \text{ 帧} \cdot \text{s}^{-1}$, 共 $10\ 560$ 帧. 系统基于 OpenCV3.3 由 C++ 和 Python 实现, 并在搭有双核 Intel Xeon E5 处理器和 64G 内存的工作站上运行.



图 7 视频采集地点

Fig.7 Video shooting location

4.2 处理结果

对视频处理后共得到 483 条轨迹, 其中机动车 175 条, 非机动车 173 条, 行人 135 条. 目标轨迹分布如图 8 所示, 速度分布如图 9 所示.

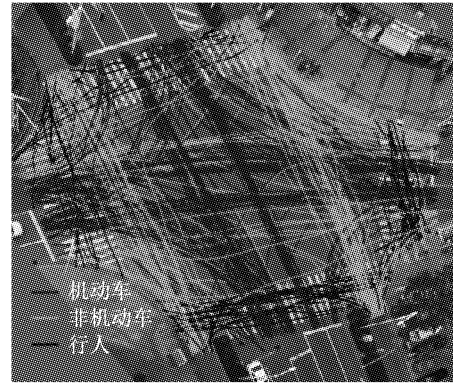


图 8 目标轨迹分布

Fig.8 Trajectories at the intersection

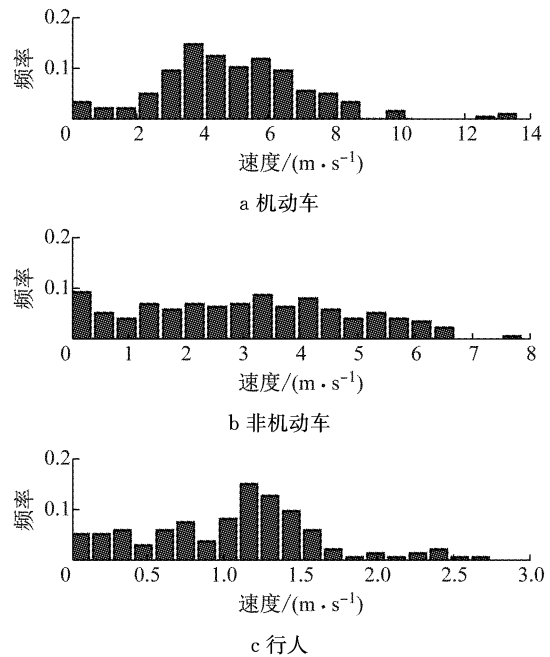


图 9 交叉口速度分布

Fig.9 Velocity distribution at the intersection

4.3 准确性检验

4.3.1 轨迹准确性

对比人工观测轨迹结果与系统自动提取轨迹结果如表 6 所示. 表 6 左半部分记录了系统捕获情况, 其中第 1 行表示: 视频中实际出现机动车 180 辆, 系统成功捕捉到其中 162 辆车的轨迹, 误把 10 辆车标记为非机动车, 6 辆车标记为行人, 漏检 2 辆车(标记为背景). 右半部分为准确率计算, 其中捕获率指系统提取轨迹数(不分类别)与实际轨迹数的比值; 分

类准确率指系统提取到的轨迹中正确分类的比例;整体准确率指系统提取到的正确轨迹数(斜体)与实际轨迹数的比值.从捕获率来看,机动车和非机动车捕获率极高,均超过 98%,说明少有目标丢失的情况,而行人由于目标面积小,被误检为背景的概率大,捕获率较前两者略有降低.从分类准确率来看,机动车的分类准确率较高,超过了 90%,行人和非机动车由于其交通行为的复杂性,分类准确率有所降低.整体准确率是前 2 个评价指标的综合体现,机动车由于体积大,行为规律性强,整体识别准确率最高,达 88.89%,相应的,行人准确率最低为 83.33%,非机动车居中为 86.00%,系统的平准准确率为 86.07%.

表 6 轨迹准确性检验

Tab.6 Trajectory extraction accuracy

情形	系统捕获的个数				捕获率/%	分类准确率/%	整体准确率/%
	机动车	非机动车	行人	背景			
机动车	162	10	6	2	98.77	90.12	88.89
非机动车	10	150	8	3	98.00	88.00	86.00
行人	3	12	120	5	95.83	87.50	83.33
背景	0	1	1				
总计	175	173	135	483	97.53	88.54	86.07

4.3.2 速度准确性

为检验目标个体轨迹和速度的准确性,视频中安排了装有 GPS 和测速仪的机动车进行对照实验.实验车辆分别在交叉口进行左转、直行和右转操作, GPS 轨迹与系统提取轨迹如图 10 所示.可以看出视频提取的轨迹与实测 GPS 轨迹有较好的重合度,且左右转弯时轨迹重合度比直行时要高,这是因为转弯车速较低,从视频中可以获得更为密集的数据,系统的处理精准度也随之提升.图 11 对比了实验车辆



图 10 GPS 轨迹与视频处理轨迹对比

Fig.10 Trajectory samples extracted from GPS and video

的真实速度和系统采集速度,与轨迹相似,系统对转弯的车速捕捉相较直行更为准确.表 7 展示了系统的速度准确率,平均车速误差指实际平均车速与系统平均车速的差异比值.整体准确率 e 为每个采样点测速误差与实际车速之比的均值(式(14)),3 条轨迹的速度整体准确率分别为 94.07%、85.01%和 96.04%,平均值为 91.71%.

$$e = \sum_u \frac{|v_{r,u} - v_{s,u}|}{v_{r,u}} \quad (14)$$

式中: e 表示整体准确率;下标 u 表示采样点序号; v_r 、 v_s 分别表示实际车速与系统车速.

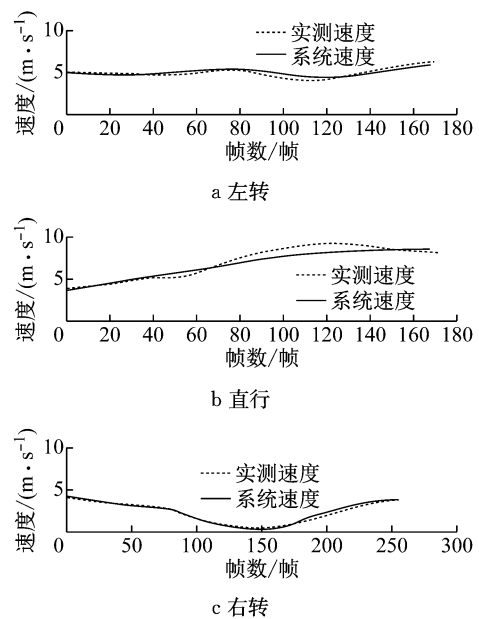


图 11 实测车速与系统提取车速对比

Fig.11 Velocity comparison between system estimation and ground truth

表 7 速度准确率

Tab.7 Velocity estimation accuracy

行为	实际平均车速/ ($m \cdot s^{-1}$)	系统平均车速/ ($m \cdot s^{-1}$)	平均车速 误差/%	整体准 准确率/%
左转	5.00	4.95	1.1	94.07
直行	6.68	7.02	5.1	85.01
右转	2.41	2.27	5.6	96.04
平均值			3.9	91.71

4.4 结果对比

表 8 列出了国内外交通视频处理研究的典型方法及其适用对象和准确性.由于研究对象和研究目的不同,各研究对准确性的检验指标有所差异,但仍具有一定参考价值.从表中可以看出,与其他研究成果相比,本文提出的视频处理方法适用于交叉口内全体交通对象,在保持较高的轨迹计数与速度检测准确率同时,加强了对停滞和遮挡现象的鲁棒性.

表 8 交通视频研究成果

Tab.8 Traffic video analysis research

视频处理方法	目标交通对象	准确性/%	停滞鲁棒性	遮挡鲁棒性
GMM ^[10]	路段内车辆	未检验	较差	无
帧差法 ^[18]	路段内车辆	流量检测:81.61	无	无
光流法 ^[18]	路段内车辆	流量检测:86.00	较差	无
Adaboost+帧差法 ^[18]	路段内车辆	流量检测:94.42	较好	较差
特征点+运动匹配 ^[19]	交叉口车辆	冲突检测:86.00	较好	较差
特征点+区域匹配 ^[8]	停止线前车辆、行人	车辆等待时间:90.00 行人过街数量:90.00	较好	较好
ViBe+OPC(本文)	交叉口内车辆、行人、非机动车	轨迹计数:86.07 速度检测:91.71	较好	较好

5 结论

提出了一种适用于交叉口全对象的轨迹和运动参数自动采集方法.方法包含前景提取、目标匹配跟踪和轨迹后处理 3 个主要步骤.在前景提取阶段,以 ViBe 为基础,提出了与 GMM 相结合的前景提取方法,在保留 ViBe 强大的背景差分能力的同时,解决了 ViBe 首帧残留造成的“虚假前景”现象.在目标匹配跟踪阶段,提出了基于 KLT 光流的轮廓-目标匹配算法,算法根据 OPC 匹配的情形不同分为 5 种情况讨论,解决了由目标间短暂遮挡造成的轨迹中断问题.在轨迹后处理阶段,通过轨迹分离和特征点修正进一步优化所提取的轨迹,并基于透视变换原理将图像坐标转换为真实坐标,从而实现对目标速度和加速度的提取.最后根据交通对象的行为和属性(速度、图像面积)对目标进行分类.经检验,系统提取的机动车、非机动车和行人轨迹准确率分别为 88.89%、86.00% 和 83.33%,速度准确率为 91.71%.

视频轨迹采集系统可为交叉口管理与安全研究提供一种高效便捷的数据采集手段,但系统的输入视频需从较高位置俯视拍摄,存在一定的应用局限性,需根据后续研究进一步改善.

参考文献:

[1] 国家统计局. 交通事故年度数据[EB/OL]. [2017-12-15]. <http://data.stats.gov.cn/easyquery.htm/>. National Bureau of Statistics of the People's Republic of China. Annual data on traffic accidents[EB/OL]. [2017-12-15]. <http://data.stats.gov.cn/easyquery.htm/>.

[2] The Organization for Economic Cooperation and Development. Road injury accidents[EB/OL]. [2016-07-28]. <http://stats.oecd.org/>.

[3] 慈玉生,郑淑妮,赵韩涛. 基于过街人数分布的路段机动车延

误经验模型[J/OL]. [2017-09-06]. <http://kns.cnki.net/kcms/detail/50.1190.U.20170906.1401.010.html>.

CI Yusheng, ZHENG Shuni, ZHAO Hantao. Empirical vehicle delay model for road segment based on pedestrians crossing number distribution[J/OL]. [2017-09-06]. <http://kns.cnki.net/kcms/detail/50.1190.U.20170906.1401.010.html>.

[4] WILLIAM H K L, CHUNG Y. Pedestrian speed/flow relationships for walking facilities in Hongkong[J]. Journal of Transportation Engineering, 2000, 126(4):343.

[5] SU Y, YAO D, ZHANG Y, et al. Vehicle/pedestrian conflict analysis and exclusive right-turn phase setting study[J]. 2008, 26(7):733.

[6] 张晓玮. 信号控制交叉口行人与右转车辆的冲突研究[D]. 成都:西南交通大学, 2016.

ZHANG Xiaowei. Study on conflict between pedestrian and right-turning vehicles at signalized intersection[D]. Chengdu: Southeast Jiaotong University, 2016.

[7] 唐克双, 杨博文, 许凯, 等. 基于车辆轨迹数据的交叉口危险驾驶行为预测[J]. 同济大学学报(自然科学版), 2017, 45(10):1454.

TANG Keshuang, YANG Bowen, XU Kai, et al. Prediction of driver's risky behaviors based on vehicle trajectory data at signalized intersections [J]. Journal of Tongji University (Natural Science), 2017, 45(10): 1454.

[8] SHIRAZI M S, MORRIS B. A typical video-based framework for counting, behavior and safety analysis at intersections [C]// Intelligent Vehicles Symposium. [S.l.]: IEEE, 2015: 1264-1269.

[9] SAUNIER N, SAYED T. Clustering vehicle trajectories with hidden markov models application to automated traffic safety analysis [C]// International Joint Conference on Neural Networks. [S.l.]: IEEE, 2006:4132-4138.

[10] 王俊骅, 张方方, 张兰芳. 基于 OpenCV 和 Halcon 的交通冲突视频自动检测及数据处理[J]. 同济大学学报(自然科学版), 2010, 38(2):238.

WANG Junhua, ZHANG Fangfang, ZHANG Lanfang. Halcon and OpenCV based traffic automatic conflicting detecting method and data transaction[J]. Journal of Tongji University (Natural Science), 2010, 38(2):238

[11] GUO Y, SAYED T, ZAKI M, et al. Safety evaluation of unconventional outside left-turn lane using automated traffic conflict techniques[J]. Canadian Journal of Civil Engineering, 2016, 38(2):238.