

基于质心法的车联网目标跟踪方法与应用

叶阳 卢奇 程时伟

浙江工业大学计算机科学与技术学院 杭州 310023

摘要 车辆目标跟踪是实现车联网不可或缺的一环,旨在获取车辆的动态信息,以提高交通运行效率。其核心是对大量监控探头采集的视频图像进行分析处理,实现车辆的实时检测与跟踪。为了进一步提高目标检测效率,降低硬件成本,文中提出了基于二帧差分法的前景检测方法,以及基于质心法的车辆轮廓检测与跟踪方法。基于 OpenCV3.4.1 和 VS2017 进行验证实验和仿真测试,结果表明,该算法对车辆跟踪的精确率达到 89.1%,平均处理耗时 42.63 ms,具有较好的实时性和鲁棒性,可在车联网嵌入式设备上部署和应用。

关键词: OpenCV; 质心法; 目标跟踪

中图分类号 TP311

Centroid Method Based Target Tracking and Application for Internet of Vehicles

YE Yang, LU Qi and CHENG Shi-wei

School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

Abstract Vehicle target tracking is an indispensable part of the realization of the Internet of Vehicles, which aims to obtain vehicle dynamic information to improve the efficiency of traffic operation. Its core is to analyze and process the video images collected by a large number of monitoring probes to realize real-time detection and tracking of vehicles. In order to further improve the efficiency of target detection and reduce hardware costs, this paper proposes a foreground detection method based on the two-frame difference method, and a vehicle contour detection and tracking method based on the centroid method. Based on OpenCV3.4.1 and VS2017, the algorithm verification and simulation test are carried out. The results show that the accuracy of the algorithm for vehicle tracking reaches 92.3%, and the average processing time is 42.63 ms. It can be deployed and applied on embedded devices in the Internet of Vehicles.

Keywords OpenCV, Centroid method, Target tracking

1 引言

随着社会经济的飞速发展,交通运输在促进中国经济社会发展具有重要作用^[1]。近年来,车联网(Internet of Vehicles, IoV)在智能交通系统(ITS)领域发展迅速,它通过建立基于物联网的车辆移动网络来管理车流、减少事故,其中道路交通信息的研究必不可少。现有大量道路视频监控探头,系统收集视频后进行处理并加以分析,在此基础上就能实现车辆检测和实时跟踪,促进车联网的应用。

然而,如何开发一种实时且鲁棒的运动车辆目标跟踪方法,一直是研究人员面临的挑战之一。传统的目标检测算法有:光流法、Canny 法、均值漂移法、Kalman 滤波法、粒子滤波法等。光流信息被用于进行光流法检测,实际效果较好,但该方法也存在不少缺点,如抗噪有所不足,计算相对来说比较复杂,要求硬件的性能很高;在 Canny 算法的第一步,使用高斯模糊去噪会导致边缘信息的丢失。均值漂移法不能适应目标

尺度的变化,不能跟踪高速目标。卡尔曼滤波只能应用于服从高斯分布的目标,是一种线性的估计方法。粒子滤波的最大问题是粒子经过多次迭代后可能退化,从而影响目标跟踪的效果。

随着深度学习算法的迅速发展,其在复杂场景下表现出了优越性,如 RCNN^[2], YOLO^[3],但是深度学习算法对设备的硬件性能要求较高,如果将其应用在电子警察、智能监控摄像头等边缘设备中,需考虑其性能限制^[4-5]。

为此,本文在已有方法的基础上进行改进,对目前已有的常见跟踪算法进行仿真分析,然后识别车辆的边缘信息,实现车辆与背景的分离;再通过质心算法对车辆的轮廓进行框定和跟踪。

2 相关工作

Xu 等针对现存的很多跟踪算法在速度和准确度方面很难满足嵌入式跟踪开发的需要,提出了一种基于 Harris 角点

的快速跟踪算法。第一步是利用 Harris 角点,将目标角点的特征概括出来,紧接着利用金字塔光流法,匹配后续视频帧的角点,最后使用基于角点的质心跟踪算法,借此匹配目标的重心,确定目标的位置,重心跟踪算法可以较好地抵消由于旋转或扭曲带来的形变问题^[6]。Wang 等采用了自适应 Canny 边缘检测算法,借此搭建一个高效且可靠的实时目标跟踪平台。由于采用的是自适应算法,它可以确定平滑参数和高低阈值,得到了较好的边缘图像。Canny 算法可以计算目标的质心,具体做法是在处理图像后,得到生成的单像素边缘图,再根据边缘图求解。同时根据时间间隔预测目标质心的下一个位置,控制伺服机构实现目标跟踪,实验表明,基于 Canny 算法的目标跟踪系统能够满足实时跟踪的需要^[7]。Yao 等提出了一种基于质心算法的均值漂移跟踪算法。初始位置采用运动目标质心,质心位置采用均值漂移迭代法,根据巴氏杀菌系数判断当前目标与参考目标的匹配程度^[8]。

对于特定物体的跟踪,如人脸跟踪^[9]、手势跟踪^[10]和人体跟踪^[11]等,更加依赖于对特定物体训练有关的检测器。而在车辆跟踪中,由于车流的高密度特点,还需要考虑不同目标之间的识别、自遮挡、互遮挡等问题^[12],同时车辆具有高机动性,进行连续跟踪就要求处理的实时性较高。Tan 等通过引入搜索连通域方法,改善了质心跟踪的稳定性^[13]。但这些只适用于环境简单、车流稀少的条件,无法处理复杂环境下的车辆跟踪。Chen 等运用背景减除法和光流法相结合的技术进行车流检测,采用距离、面积双约束条件对车辆目标进行匹配跟踪^[14]。同样地,对于车流大、车道堵等复杂情况,还需提高检测的准确性。Zhou 等采用改进的自适应背景差分法对车辆进行检测,并结合迭代式阈值法和人工阈值法完成目标的分割,提高了检测精度^[15]。但由于初始值的设置与实际有所偏差,初始的跟踪效果较差。Zhu 等提出了一种基于 Mean-Shift 结合视觉词袋的车辆跟踪方法,采用轮廓和颜色信息进行初匹配,利用视觉词袋特征进行再次匹配,提高了跟踪的准确率^[16],但耗时较多,无法适用于实时性高的场景。Guo 等提出了一种基于中值滤波和多特征融合的粒子滤波算法,利用边缘和颜色特征之间的互补性,能够在遮挡、光照和旋转情况下对车辆的有效跟踪^[17];但是其计算复杂度较高,平均检测帧率只有 8 帧每秒,且只能跟踪单个目标。

综上所述,已有的研究大多存在无法适应复杂环境、耗时长、处理帧率低等问题。因此,本文将针对这些挑战做进一步探索,将多种方法相结合,从而提高车辆跟踪的准确率和效率。

3 图像预处理

本文使用基于 OpenCV (V3.4.1)的图像灰度化处理,其函数如下:

```
void cv::cvtColor(
    cv::InputArray src, //输入序列
```

```
cv::OutputArray dst, // 输出序列
    int code, //颜色映射码
    int dstCn=0 //输出的通道数
);
```

cvtColor 函数的参数如表 1 所列。原图与经灰度化处理后的图如图 1、图 2 所示。

表 1 cvtColor 函数的参数说明

Table 1 cvtColor function parameter description

InputArray src	颜色空间变换的原图像,可以是 Mat 类
OutputArray dst	颜色空间变换后存储图像,也可以是 Mat 类
int code	转换的代码或标识,即在此确定将什么制式的图片转换成什么制式的图片 COLOR_BGR2GRAY;由彩色图像处理成灰度图像 COLOR_GRAY2BGR;由灰度图像处理成彩色图像
int dstCn	目标图像通道数,如果取值为 0,则由 src 和 code 决定



图 1 原图

Fig. 1 Original image



图 2 灰度化处理图

Fig. 2 Gray processing diagram

中值滤波是非线性的信号处理方法,该方法能够解决一定条件下线性滤波器处理图像细节模糊的问题。图 3 是经过中值滤波处理后的效果图。



图 3 中值滤波处理图

Fig. 3 Median filter processing diagram

而高斯滤波赋予邻域内每个像素不同的权值,在进行图像平滑的同时,能够更多地保留原有图像中总体的分布特征。图 4 是经过高斯滤波处理过的效果图。



图4 高斯滤波处理图

Fig. 4 Gaussian filter processing diagram

进一步对图像进行形态学中的腐蚀处理,将图像的每个像素都用 3×3 的结构元素扫描,并将它们覆盖的结构元素和二进制图像用于“与”运算。如果两者都为1,则结果图像的像素为1,否则为0。腐蚀处理效果如图5所示。



图5 腐蚀处理效果示意图

Fig. 5 Corrosion treatment diagram

对图像进行膨胀处理,将图像的每个像素都用 3×3 的结构元素扫描,并且它们覆盖的结构元素和二进制图像用于“或”运算。如果两者都为0,则结果图像的元素为0,否则为1。膨胀处理效果如图6所示。

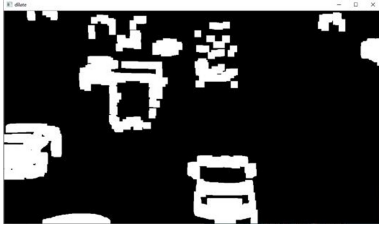


图6 膨胀处理效果示意图

Fig. 6 Expansion processing diagram

4 前景检测

前景是图像或视频场景中可见且更重要的部分。前景检测是识别视频中感兴趣的区域或移动对象(前景)和静态部分(背景)的过程。这是运动分析和视频监控中的关键步骤,也是最基本和最低级别的步骤。前景检测结果的质量与后续分析和应用程序处理的性能直接相关。

摄像机收集的视频序列具有连续性的特征。如果场景中没有移动目标,则连续帧的变化非常微弱;如果存在移动目标,则连续帧之间的变化将很大。前人基于上述思路提出了帧差法,具体地对两到三帧连续图像执行微分运算,减去与不同帧相对应的像素以确定灰度差异的绝对值。当绝对值超过某个阈值时,可以将其确定为移动目标。帧差法具有实现简单、运算效率高、对动态环境适应性强等优点,但对光线变化的敏感度较低。

二帧差分法的简要实现过程如下:

设 $f_k(x, y)$ 为第 k 帧图像, $f_{k-\tau}$ 为第 $k-\tau$ 帧图像,两帧图

像的差 $D_k(x, y)$ 为:

$$D_k(x, y) = |f_k(x, y) - f_{k-\tau}(x, y)|$$

接着,对两帧差值 $D_k(x, y)$ 进行二值化:

$$R_k(x, y) = \begin{cases} 1, & D_k(x, y) > T \\ 0, & \text{else} \end{cases}$$

其中, T 为阈值,需依据实际经验而定。如果其值太大可能会造成空洞,甚至检测不到目标;如果太小则会出现噪音。本文先后尝试多个固定的阈值(如50,100),效果均不理想,因此最终采用自适应阈值,其核心是通过计算某个邻域的高斯加权平均来确定阈值,能很好地适应某些光照不均的图像。

二帧差分法是一种实时性良好的检测方法,且具有一定的抗干扰能力,因此本文基于二帧差分法进行前景检测。图7是将连续两帧做差后的图像。

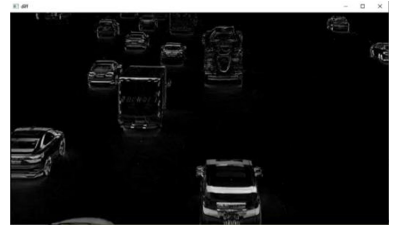


图7 差值图效果示例

Fig. 7 Example of difference graph effect

图8给出了采用了固定阈值的二值化处理后的效果。



图8 固定阈值化处理图

Fig. 8 Fixed threshold processing diagram

图9给出了采用了自适应阈值的二值化处理后的效果。



图9 自适应阈值化处理图

Fig. 9 Adaptive thresholding processing diagram

5 基于质心法的车辆跟踪

5.1 质心法

质心是质点系的中质量分布的平均位置,是质量中心的简称。假设 n 个质点组成一个质点系,它们的质量分别为 m_1, m_2, \dots, m_n 。不妨假设 r_1, r_2, \dots, r_n 分别表示矢径,即各个质点到某一固定点 O 的距离,用 r_σ 来表示,则有:

$$r_\sigma = \frac{\sum m_i r_i}{M}$$

其中,

$$M = \sum_{i=1}^n m_i$$

M 表示质点系的总质量。质点的坐标值会随着不同的坐标系选择而不同,但不变的是质心,因为质心不受质点系中每一个质点的相对位置或者坐标系的选择影响。质点系的质心只受每个质点的质量以及分布的相对位置影响。把用于计算质心的算法统称为质心法,它的作用是大幅提高处理图像的精度。

5.2 质心计算

除了以上的普通质心算法以外,还有几种其他常见的质心算法,如加强加权、阈值、距离等质心算法。由于普通质心算法的低复杂度能更好地适用于经济型嵌入式设备,本文采用了普通质心算法,具体包括以下几个步骤:

(1)轮廓检测。根据前景检测得到的二值图像进行轮廓检测,使用函数 `cvFindContours()`:

```
int cvFindContours(
    CvArr * image,
    CvSeq * * first_contour,
    CvMemStorage * storage,
    int mode=CV_RETR_LIST,
    int header_size=sizeof(CvContour),
    int method=CV_CHAIN_APPROX_SIMPLE,
    CvPoint offset=cvPoint(0,0)
);
```

其中, `image` 表示八比特、单通道的二值图像。非 0 像素处理为 1, 0 像素保持不变; `first_contour` 用于储存第一个外接轮廓的输出参数; `storage` 表示返回目标轮廓的容器; `mode` 表示检测方式; `header_size` 表示视频序列尺寸; `method` 表示求解边缘的近似方法; `offset` 表示偏移量。

(2)计算轮廓距。使用函数 `moments()`:

```
cv::moments(
    bool binaryImage=false
    InputArray array,
);
```

其中, `array` 表示输入数组,可以是光栅图像(单通道、八比特或浮点型二维数组),或者是一个类型为 `Point` 或 `Point2f` 的二维数组。`binaryImage` 默认值是 `false`, 如果为 `true`, 则所有非零的像素都会按值 1 对待,相当于进行了二值化处理,阈值为 1, 此参数仅对图像有效。

(3)计算每辆车的质心。定义 `Moments` 类:

```
class Moments{
public:
    Moments();
    Moments(double m00, double m10, double m01, double
    m20, double m11, double m02, double m30, double
    m21, double m12, double m03 );
    Moments(const CvMoments& moments );
    operator CvMoments() const;
    // spatial moments 空间矩
    double m00, m10, m01, m20, m11, m02, m30, m21,
    m12, m03;
```

// central moments 中心矩

double mu20, mu11, mu02, mu30, mu21, mu12, mu03;

// central normalized moments 归一化中心矩

double nu20, nu11, nu02, nu30, nu21, nu12, nu03;

}

spatial moments(空间矩)的计算式为:

$$m_{ji} = \sum_{x,y} (array(x,y) \cdot x^j \cdot y^i)$$

其中, `array` 即密度函数 $f(x, y)$, 当 i 和 j 都为 0 时, 即 m_{00} 。 m_{00} 表示质量, 或者面积。 `mass center` (质心) 的计算式是:

$$\bar{x} = \frac{m_{10}}{m_{00}}$$

$$\bar{y} = \frac{m_{01}}{m_{00}}$$

结合前文对于质心的定义, 可以计算质心:

```
mc[i]=Point2d(mu[i], m10/ mu[i], m00, mu[i], m01 /
mu[i], m00);
```

(4)查找正外接矩形。轮廓外接矩形函数的原型如下:

```
CV_EXPORTS_W Rect boundingRect(
    InputArray points
);
```

其中, `points` 表示输入的二维点集, 可接受 `Mat` 类型或是 `Vector` 向量。

然后使用下述 `rectangle()` 函数绘制出符合要求的矩形:

```
void rectangle(
    Mat& img,
    Rect rec,
    const Scalar& color,
    int thickness=1,
);
```

其中, `img` 表示输入图像为 `Mat` 类型; `rec` 表示矩形对象; `color` 表示矩形框线条的颜色, 以 RGB 形式表示, 数值为 0~255; `thickness` 表示线条的粗细, 取负值时为填充整个图形。

(5)质心匹配。经过上述处理, 我们得到车辆目标的质心坐标和正外接矩形, 为了在当前帧图像中找到目标, 搜索整个图像会导致耗时大量增加, 是极不合理的。因此, 本文采用局部搜索以满足实时性要求。

```
W0 = boundRect[i]. width;
```

```
H0 = boundRect[i]. height;
```

```
W1 = W0 + aW0;
```

```
H1 = H0 + bH0;
```

其中, W_0 为第 i 个外接矩形的宽度, H_0 为第 i 个外接矩形的高度, W_1 为搜索区域宽度, H_1 为搜索区域高度。

在当前帧中将前一帧质心 $mc[i]$ 作为搜索中心, 查找车辆轮廓, 计算搜索到的轮廓的质心 $mc[j]$ 。若唯一, 则认为该质心对应的轮廓为跟踪目标; 若不唯一, 则限制两帧中的质心欧氏距离小于阈值 T_d , 并取最大的轮廓为跟踪目标。

5.3 实验结果

图 10—图 12 是部分帧(第 105—107 帧)的效果图。使用搭载 Intel Core i7-6700HQ CPU 和 8GB 内存的电脑进行仿真测试。测试材料是一部时长为 10 s 的视频, 分辨率为 1 008 *

566, 帧率为 30 帧每秒, 共读取 300 帧图像进行车辆跟踪测试。由于篇幅有限, 部分实验数据(共 15 帧)如表 2 所列。

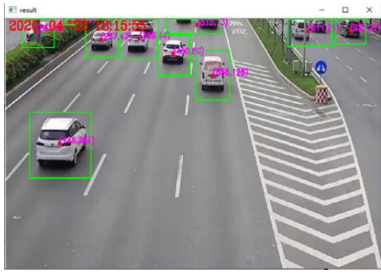


图 10 第 105 帧图像

Fig. 10 Frame 105 image

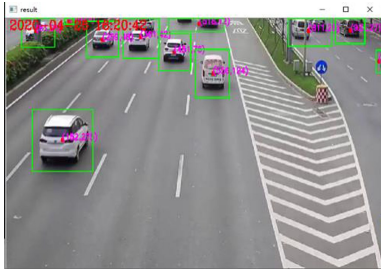


图 11 第 106 帧图像

Fig. 11 Frame 106 image

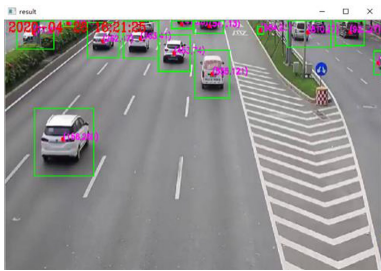


图 12 第 107 帧图像

Fig. 12 Frame 107 image

表 2 车辆跟踪耗时

Table 2 Centroid method tracking time-consuming

Frame position(帧数)	Duration(耗时)/s
71	0.042
72	0.041
73	0.041
74	0.037
75	0.041
76	0.039
77	0.040
78	0.041
79	0.041
80	0.036
81	0.040
82	0.041
83	0.039
84	0.043
85	0.044

统计分析全部 300 帧的耗时数据, 去除最大、最小值, 平均耗时为 42.63ms, 且在整个测试过程中耗时的变化幅度小于 10ms。视频中实际出现车辆共 147 辆, 成功跟踪 131 辆, 跟踪丢失 16 辆, 车辆跟踪准确率达 89.1%。具体结果如表 3 所列。

表 3 算法的准确率和耗时的比较

Table 3 Comparison of algorithms accuracy and time consuming

算法	准确率/%	耗时/ms
文献[18]	82.9	42
文献[19]	91.2	98
本文算法	89.1	42.6

从实验结果可以看出, 本文算法在耗时与文献[18]中的算法相当的情况下, 明显提升了跟踪准确率; 在准确率与文献[19]中的算法相近的情况下, 大幅缩短了跟踪耗时。因此, 该算法能满足实际交通场景, 适用于经济型嵌入式设备。

结束语 本文通过对视频图像进行分析处理以获取车辆动态信息, 先利用二帧差分法进行图像的前景检测, 识别出车辆目标的轮廓; 在此基础上, 基于质心法提出了实时性较强的目标跟踪算法, 在不影响跟踪效果的前提下, 简化算法复杂度, 降低对硬件性能的要求。实验结果表明, 该算法在保证较好的鲁棒性和准确性的同时, 能有效提高跟踪速度, 可用于车辆视频的快速跟踪, 适用于车联网应用中的经济型嵌入式设备。

下一步将针对车辆跟踪中的空洞检测、车辆之间的互相遮挡, 以及动态跟踪时的准确率降低等问题展开进一步研究, 以提高检测效果和效率。

参考文献

- [1] SONG H S, LI Y, YANG J, et al. Vehicle target tracking based on highway scenes[J]. Computer System Applications, 2019, 28(6): 82-88.
- [2] 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.
- [3] 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.
- [4] SHI W S, SUN H, CAO J, et al. Edge computing: a new computing model for the Internet era[J]. Journal of Computer Research and Development, 2017, 54(5): 907-924.
- [5] HUANG K Q, CHEN X T, KANG Y F, et al. Overview of intelligent video surveillance technology[J]. Journal of Computer Science, 2015, 38(6): 1093-1118.
- [6] XU L P, GENG B, LI X L, et al. Design of target tracking algorithm based on Harris corner combined with pyramid optical flow method[J]. Computer Measurement and Control, 2018, 236(5): 168-171, 175.
- [7] WANG Y W, YU H X, YAO B, et al. Target tracking based on Canny detection algorithm[J]. Electronic Design Engineering, 2012(3): 149-152.
- [8] YAO F W, XU C M. Mean shift tracking algorithm based on target centroid[J]. Computer Technology and Development, 2012(6): 110-112, 116.
- [9] WANG J X, LEI Z C. A Convolutional Neural Network Face Recognition Algorithm Based on Feature Fusion[J/OL]. Progress in Laser and Optoelectronics: 1-12. [2019-12-30]. <http://kns.cnki.net/kcms/detail/31.1690.TN.20191106.1156.026.html>.

orative Representation using Grassmann Manifolds for Image Set Classification[J]. *Pattern Recognition*, 2020, 100:107123.

[20] ZHANG Y, YUAN Y H, OU W H, et al. Weighted Discriminative Collaborative Competitive Representation for Robust Image Classification[J]. *Neural Networks*, 2020, 125:104-120.

[21] ZENG S, ZHANG B, LAN Y, et al. Robust Collaborative Representation-based Classification via Regularization of Truncated Total Least Squares [J]. *Neural Computing & Applications*, 2019, 31(10):5689-5697.

[22] GOU J, HOU B, YUAN Y, et al. A New Discriminative Collaborative Representation-based Classification Method via l₂ Regularizations [J]. *Neural Computing and Applications*, 2020, 32(8).

[23] MA X, HU S, LIU S, et al. Multi-focus Image Fusion based on Joint Sparse Representation and Optimum Theory [J]. *Signal Processing: Image Communication*, 2019, 78:125-134.

[24] HUANG L, MA Y, LIU X. A General Non-Parametric Active Learning Framework for Classification on Multiple Manifolds [J]. *Pattern Recognition Letters*, 2020, 130:250-258.

[25] HUANG P, YANG Z, CHEN C. Fuzzy Local Discriminant Embedding for Image Feature Extraction [J]. *Computers and Electrical Engineering*, 2015, 46:231-240.

[26] YANG M, SHANG R, JIAO L, et al. Feature Selection based Dual-Graph Sparse Non-Negative Matrix Factorization for Local Discriminative Clustering [J]. *Neurocomputing*, 2018, 290:87-99.

[27] YANG J, ZHANG D. Median Fisher Discriminator: a Robust Feature Extraction Method with Applications to Biometrics [J].

Frontiers of Computer Science in China, 2008, 2(3):295-305.

[28] ALEIX M, ROBERT B. The AR Face Database [R]. *Cvc Technical Report*, 1998, 24.

[29] <http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html>.

[30] SIM T, BAKER S, BSAT M. The CMU Pose, Illumination, and Expression database [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, 25(12):1615-1618.

[31] A P J P, B H W, B J H, et al. The FERET database and Evaluation Procedure for Face Recognition Algorithms [J]. *Image and Vision Computing*, 1998, 16(5):295-306.

[32] CHEN L, MAN H, NEFIAN A V. Face Recognition based on Multi-class Mapping of Fisher Scores [J]. *Pattern Recognition*, 2005, 38(6):799-811.



YANG Zhang-jing, born in 1979, associate professor. His main research interests include computer vision and pattern recognition, etc.



HUANG Pu, born in 1985, associate professor. His main research interests include machine learning and pattern recognition, etc.

(上接第 344 页)

[10] XU Q Y, QIN G H, SUN M H, et al. Feature Fusion based Hand Gesture Recognition Method for Automotive Interfaces [J/OL]. *Chinese Journal of Electronics*:1-12. [2021-01-22]. <http://kns.cnki.net/kcms/detail/10.1284.TN.20200727.1424.004.html>.

[11] MA M, LI Y B, WU X Q, et al. Multi-feature fusion human pose tracking in video [J]. *Journal of Image and Graphics*, 2020, 25(7):1459-1472.

[12] ZHOU Z. Design and implementation of multi-target tracking system based on embedded platform [D]. Beijing: Beijing University of Technology, 2018.

[13] TAN X Q, DONG C J. A vehicle tracking algorithm based on video image sequence [J]. *Modern Industrial Economics and Informatization*, 2016, 6(1):80-81, 84.

[14] CHEN C, ZHU Y, XIAO Y L, et al. An effective vehicle tracking algorithm and abnormal vehicle detection [J]. *Journal of East China University of Science and Technology (Natural Science Edition)*, 2015, 41(2):205-209.

[15] ZHOU M J, WANG J W. Research on the Detection and Tracking Methods of Moving Vehicles in Video Sequences [J]. *Techn*

nology and Innovation, 2017(4):76-78.

[16] ZHU H N, XU M M, SHEN Y. Research on Multi-Video Vehicle Tracking Based on Mean Shift [J]. *Computer Science*, 2018, 45(S1):220-226.

[17] GUO X X, CUI A J, WAN H L, et al. Research on particle filter vehicle tracking algorithm based on median filter and multi-feature fusion [J]. *Journal of Shandong Normal University (Natural Science Edition)*, 2017, 32(3):69-75.

[18] LI T. Research on Application of video surveillance detection algorithm in traffic intersection [D]. Hubei: Wuhan University of technology, 2009.

[19] ZHAO J M, ZHANG L P. Research on moving vehicle detection and tracking technology in traffic video [J]. *Vehicle and Power Technology*, 2012(4):46-49.



YE Yang, born in 1980, postgraduate, lab master, is a member of China Computer Federation. His main research interests include virtual reality, digital image processing and human-computer interaction.