文章编号：2096-1618(2023)04-0381-06

基于双目稀疏场景流的智能车运动目标检测

刘明文， 蒋 涛， 袁建英， 顾硕鑫， 徐智勇， 雷 婷
(成都信息工程大学自动化学院，四川 成都 610225)

摘要：为提高无人驾驶汽车视觉运动目标检测精度，提出一种基于深度学习与稀疏场景流结合的运动目标检测方法。首先使用深度学习网络 SOLO V2 分割交通场景，提取行人、车辆等潜在运动目标，缩小场景内运动目标搜索范围。其次，利用背景中特征匹配点估计相机自运动参数，在此基础上将潜在运动区域前后两帧特征点坐标映射到同一坐标系下，进而计算出仅由运动目标产生的稀疏场景流。最后，根据每个目标场景流估计误差的不同，计算每个目标场景流估计的不确定度，然后使用独立自适应阈值用于运动状态判断。使用 KITTI 数据集进行测试，实验结果表明：所提算法能明显提升运动目标检测精度，算法精度和召回率在两组数据集分别为92.3%、94.4%和87.4%、95.1%。

关 键 词：无人驾驶汽车；运动目标检测；场景流；自适应阈值
中图分类号：TP391.4 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2023.04.001

0 引言

环境感知是高级驾驶辅助系统(advanced driver assistance system，ADAS)和自动驾驶系统(autonomous driving solution，ADS)的核心问题之一，其中运动目标检测是感知系统的重要组成部分，因为它在机器人导航、同时定位与建图(simultaneous localization and mapping，SLAM)和交通监控等领域发挥着重要的作用。目前相机静止情况下的运动目标检测已被广泛研究，并提出了许多有效的解决方案[1]，包括背景减法[2-4]、帧差法[5-6]和光流法[7]。然而，相机运动情况下的运动目标检测仍然是一项具有挑战性的任务。目前对该问题解决的方案有基于单目相机、双目相机和激光雷达等传感器的方式。其中单目相机虽然硬件成本低并且能获取丰富的场景信息，但是缺乏深度信息；激光雷达虽然能获得场景高精度几何信息，但价格昂贵；而双目相机可以以低的硬件成本获取环境丰富的纹理、色彩和深度信息。因此研究基于双目相机的运动目标检测受到了越来越多学者关注。

根据运动目标检测原理的不同，基于双目相机的运动目标检测可分为 3 大类：基于极线约束的方法、光流法和结合深度学习与传统检测方法的检测方法。通过极线约束判断运动目标，首先通过特征点匹配计算前后两帧的基础矩阵，然后将超出极线约束阈值的点判断为运动点[8-9]。这种方法虽然简单，但基础矩阵在相机没有平移只有旋转或者所有匹配点都共面的情况下将失效，因此该方法适用场景受限。

基于光流的运动目标检测方法首先估计两帧之间相机的自运动，然后计算前后两帧特征点的光流，最后通过判断特征点是否超出运动判断准则的方式对目标的运动状态进行判断[10-12]。文献[10]使用一阶高斯近似传播的残差运动光流(residual image motion flow，RIMF)的不确定性代替固定阈值判断目标是否移动，文献[11]通过深度信息估计道路平面提升相机自运动参数估计的精度，并优化阈值设定方法以提升运动目标检测的鲁棒性。但是使用稠密光流有运算量大、耗时长的缺点；使用稀疏光流能够节省运算时间，但容易出现漏检的情况。

随着深度学习的飞速发展，近年来深度学习被应用于各个领域。文献[13-16]将语义信息加入系统，分割出图像潜在运动区域，一方面排除运动区域对相机自运动参数估计的干扰，另一方面完成对目标的分割。但目前大多数研究是使用光流或重投影误差的方式与深度学习方法结合判断物体运动状态。光流和重投影误差属于二维平面特征，受透视原理制约，对于平行或近似平行光轴运动的目标，即使在空间具有较大的运动位移，但其在像平面上位移也较小。对于智能驾驶交通环境，周围车辆和本车同向行驶是普遍存在的情况，此时目标运动方向和光轴方向夹角并不大，极易导致目标在像平面的运动特征不显著，进而导致基于光流或者反投影残差的运动目标判断方法失效。

针对以上问题，本文提出一种基于深度学习与稀疏场景流结合的运动目标检测方法，获取图像中运动目标及其运动矢量。相较于其他方法，该方法创新点如下：(1)通过深度学习网络完成目标分割，并仅使用

收稿日期：2022-09-09
基金项目：国家自然科学基金资助项目(62103064)；四川省自然科学基金资助项目(22NSFSC2317)；四川省科技计划资助资助(2021YFG0133、2021YFG0295、2021YFH0069、2021YFQ0057、2022YFS0565、2022YFN0020、2021YFG0308)
通信作者：袁建英. E-mail：yuanjy@cuit.edu.cn

静止区域特征的点进行相机自运动参数估计,避免运动区域对其计算的影响;(2)以目标表面特征点场景流作为目标运动状态判断参数,解决了二维平面运动特征受到几何透视影响的问题;(3)根据检测目标的场景流估计误差的不同,为每个目标设置独立自适应阈值用于判断运动状态。

1 系统框架

提出一种基于深度学习与稀疏场景流结合的运动目标检测算法,其系统框架如图 1 所示。首先,使用深度学习网络对场景进行分割,提取出潜在运动目标和静止的背景区域。其次,对图像提取特征点,静态区域特征点用于相机自运动参数估计,潜在运动区域特征点用于估计目标的稀疏场景流并合成运动目标的运动矢量。最后,计算每个目标场景流不确定度,根据每个目标场景流估计误差的不同为每个目标设置独立自适应阈值用于判断目标运动状态。

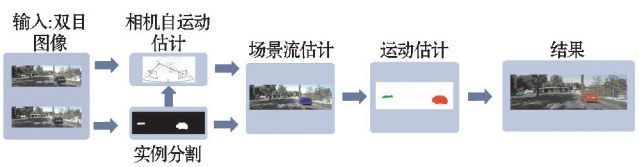


图 1 系统框图

1.1 实例分割

通常交通环境图像中的语义信息分为静态区域的背景(例如道路、建筑和树木)和潜在区域的前景(例如行人和车辆)。本文使用深度学习网络 SOLO V2[17] 对图像中存在的潜在运动目标进行实例分割,以相机左侧图像作为网络输入,网络输出为带标签的掩码区域,不同的标签代表着不同的目标,未被分类的区域被赋予背景标签。实例分割效果如图 2 所示,图 2(a)是原图,图 2(b)是实例分割后的图像,黑色区域代表静止的背景区域,灰度图像代表潜在运动目标所在区域,不同的目标使用不同的灰度值代表。

(a)原图

(b)实例分割结果

图 2 实例分割示例图

1.2 相机自运动参数估计

使用实例分割将潜在的运动目标分割为掩码区域,在相机自运动参数估计中将剔除这部分区域,仅使用场景中静止的区域估计相机的位姿变化。图 3 所示为通过环形匹配的方式在前后两个时刻的左右图上找到匹配的特征点。

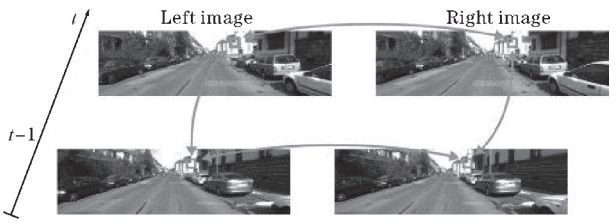


图 3 特征点的环形匹配

记 p^i 是第 i 个特征点在图像坐标系下的坐标 $[u,v,1]^{\mathrm{T}}$,m^i 是第 i 个特征点在相机坐标系下的三维坐标 $[x,y,z]^{\mathrm{T}}$。设 ${}^{t}p_t^i$ 和 ${}^{t-1}p_{t-1}^i$ 分别为当前帧和前一帧左图上的第 i 个匹配点,其对应关系为 $\{{}^{t-1}p_{t-1}^i \rightarrow {}^{t}p_t^i\}$,在各自的相机坐标系下的三维坐标对应关系为 $\{{}^{t-1}m_{t-1}^i \rightarrow {}^{t}m_t^i\}$。以 ${}^{t-1}m_{t-1}^i$ 为例进行说明,其中左上标 $t-1$ 代表以 $t-1$ 时刻的相机坐标系为参考系,右下标 $t-1$ 代表观测该点的时刻为 $t-1$ 时刻,右上标 i 代表第 i 个特征点。为求解相邻两帧间最优的相机旋转变化参数 R 和相机平移变化参数 t,可以构建代价函数:

$$\sum_{i=1}^{N} \| {}^{t}p_t^i - \pi^{(t)}({}^{t-1}m_{t-1}^i;R,t) \|^2 + \| {}^{t}p_t^i - \pi^{(r)}({}^{t-1}m_{t-1}^i;R,t) \|^2 \tag{1}$$

式(1)的前半部分是图 3 左图的代价函数,后半部分为图 3 右图的代价函数。其中 π 为投影函数,$\pi^{(l)}$ 是将三维空间的点投影到左相机上,$\pi^{(r)}$ 是将三维空间点投影到右相机。使用高斯牛顿优化方法对 R 和 t 进行优化求解。

1.3 场景流估计

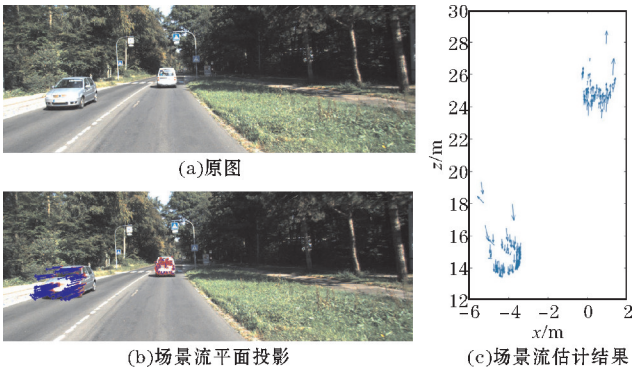
在得到相邻两帧间相机的旋转变化参数 R 和平移变化参数 t 后,可以计算出前一帧相机坐标系下特征点在当前帧相机坐标系下的映射 ${}^{t}m_{t-1}^i$。其运算关系:

$${}^{t}m_{t-1}^i = R^{t-1}m_{t-1}^i + t \tag{2}$$

则第 i 个特征点对应的场景流 v_i 的计算公式:

$$v_i = {}^{t}m_t^i - {}^{t}m_{t-1}^i \tag{3}$$

场景流估计如图 4 所示,图 4(a)为原图像;图 4(b)中箭头为特征点在图像平面上的运动位移;图 4(c)中的箭头代表特征点在三维空间中的场景流(x 轴和 z 轴方向)。



(a)原图　(b)场景流平面投影　(c)场景流估计结果

图 4　场景流估计

由于当前结果仅为离散采样的结果,不能代表该目标的整体运动,因此还需将离散的场景流合成为运动目标整体的运动矢量。将离散的场景流结果中属于该目标的特征点三维坐标的平均值作为合成场景流的起点,将各个离散场景流的方向和模长取平均值作为合成场景流的方向与均值。

1.4　运动目标检测

与光流不同,场景流(理想情况下仅由场景中目标运动引起)可以直接判断目标是否移动,但由于场景流估计误差的存在,运动状态的判决并不简单。通常的方法是选取一个固定阈值作为静态目标和动态目标的分界点,这种方法虽然简单但会出现阈值设计上的难点。阈值太高能使系统减少将静止目标误检为运动目标,但这也会使低速运动目标被错误地判断为静止目标;若阈值太低,由于场景流计算中存在误差,可能将静止目标误判为运动目标。

由场景流计算原理可知,场景流计算误差和以下因素有关:目标与相机距离、目标表面特征点匹配精度,目标表面纹理丰富程度等。也就是说,对一幅图像的不同目标场景流估计误差是不同的。基于上述原因,本文引入场景流估计不确定度。计算每个目标场景流不确定度,并为每个目标设定独立的自适应阈值。由于场景流是一个向量,因此场景流的误差评估需要同时考虑模长的差异和角度间的差异。

假设一幅图像上有多个目标,对于其中一个目标有 N 个特征点。则 P_i 就表示该目标第 i 个特征点对应场景流的不确定度,P 表示该目标的场景流的不确定度。其计算方法如下:

$$P_i=1-\frac{\exp\left(-\left(\frac{s_i^2}{2\sigma_S^2}+\frac{r_i^2}{2\sigma_r^2}\right)\right)}{2\pi\sigma_S\sigma_r} \tag{4}$$

$$P=\sum_{i=1}^{N}\frac{P_i}{N} \tag{5}$$

其中:

$$s_i=\theta_{v_i}-\theta_{\bar{v}} \tag{6}$$

$$r_i=r_{v_i}-r_{\bar{v}} \tag{7}$$

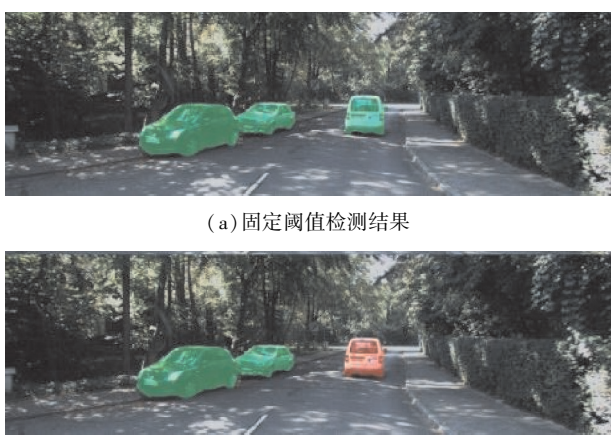
其中 σ_S 和 σ_r 分别代表单个目标中场景流的方向和模长的标准差,其中 θ 和 r 分别代表场景流的角度和方向,其中 s_i 和 r_i 代表场景流中第 i 个场景流与该目标的场景流均值 $\bar{v}$ 的方向角度差和模长差。

若 P 超过某个阈值,则认为本次估计场景流的误差较大,无法用于判断目标的运动状态。低于阈值的结果则进行运动状态阈值的计算。

$$\mathrm{th}=\overline{\mathrm{th}}+kP \tag{8}$$

其中 th 代表最终判断运动状态的阈值,$\overline{\mathrm{th}}$代表一个基准阈值常数,k 是比例常数,P 是不确定度。若目标场景流模长超过阈值 th,则认为是运动目标。

运动目标检测效果如图 5 所示,绿色代表检测结果为静止目标,红色为运动目标。其中,图 5(a)中低速运动车辆(图 5(a)最右边的车辆)低于固定阈值因此未能识别运动状态,图 5(b)中使用独立自适应阈值,成功检测到运动车辆。



(a)固定阈值检测结果

(b)独立自适应阈值检测结果

图 5　运动目标检测效果对比

2　实验结果与分析

在 KITTI 公开数据集中选取 304 个动态场景,并分为 KITTI1 和 KITTI2 两组对比数据集。该数据集既包含较为简单的仅含汽车的公路场景和高速路场景,同时也包含了具有行人、车辆和骑行自行车的人的复杂街区场景。另外,选用本地校园数据集对所提算法进行效果验证,选取的实验数据集均为连续两个时刻的双目相机图像。所有实验均在一台笔记本电脑上完成,该计算机的处理器是 AMD Ryzen 5 4600H CPU 3.0GHz,具有 16 GB RAM 和 GTX 1650,实验环境是 Ubuntu 16.04和 OpenCV 3.4.7。

2.1　定性分析

2.1.1　独立自适应阈值效果对比

为验证独立自适应阈值的效果,将提出的算法中阈值设定方法分别设为固定阈值,与独立自适应阈值进行对比测试。以 KITTI 公开数据中的数据作为测试样例,选取低速运动的车辆和行人进行效果的分析。其中,红色区域为运动目标,绿色区域为静止为物体。

由图 6 可以看出,使用独立自适应阈值可以明显提高低速运动的车辆的检测能力。

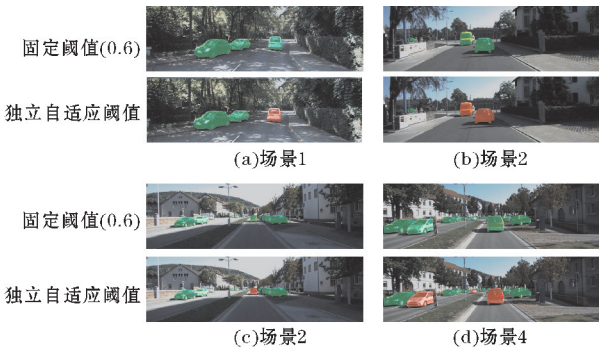


图 6 低速运动车辆效果对比

由图 7 可以看出,即使将固定阈值降低,依然会出现如图 7(a)、(b)、(d)中低速运动的行人无法被检测到和图 7(c)中能检测到骑行的人,但是漏检了运动速度更低的行人。而独立自适应阈值能够检测到图像中存在的运动汽车、骑行的人和运动速度更低的行人。

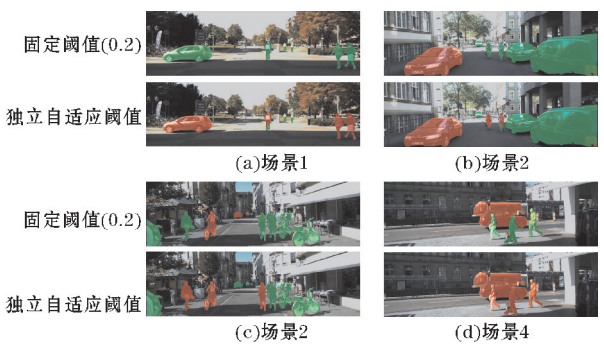


图 7 运动行人检测效果对比

为减少漏检运动物体而降低阈值,容易出现误检(红色框中的目标),如图 8 所示。本文所提方法不但能正确检测到运动目标,也能减少误检的发生,效果优于固定阈值。

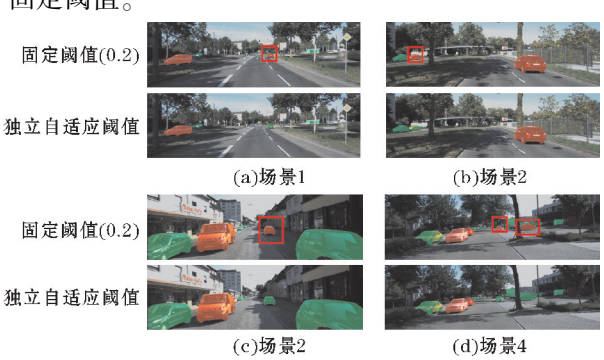


图 8 误检测效果对比

2.1.2 本文算法与其他算法对比

Zhou 等[10]和 Lin 等[11]作为在移动平台检测运动目标的经典算法,本文将与这两种算法进行对比。为了更好地展示,将本文算法输出结果与其他算法风格统一,仅显示红色的运动区域,并将结果分为一般场景和挑战场景进行展示。一般场景选取光照均匀,场景中运动目标仅为车辆的场景。挑战场景选取低速运动目标、远距离小目标、亮度不均、光照变化强烈的场景。

由图 9 可以看出,本文算法在这些场景中相较于 Zhou 和 Lin 的算法能做到更少的漏检如图 9(a)、(b)、(c)。也能做到更少的误检,如图 9(d)中静止的行人,具有更加优异的性能表现。

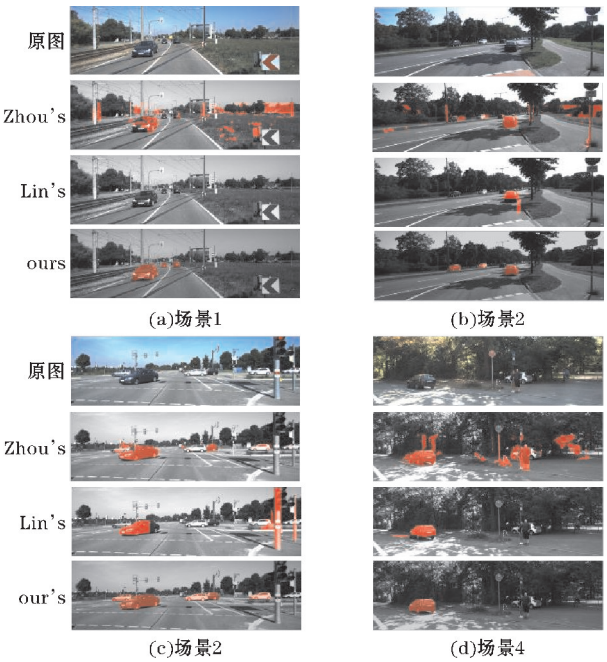


图 9 一般场景

由图 10(a)、(b)可知,本文算法在街道场景中对低速运动的运动目标具有较好的识别能力,其中漏检的行人(图 10(a)最右侧)是由于该行人上的采样点过少;由图 10(c)可知,本文算法相较于 Zhou 和 Lin 算法对于沿相机运动方向的目标具有检测能力,这也是场景流相对于光流的优势;由图 10(d)可知,本文算法在进入隧道后即使光线变化强烈也能识别到运动目标,并且能检测到隧道远处(40 m)的运动目标。

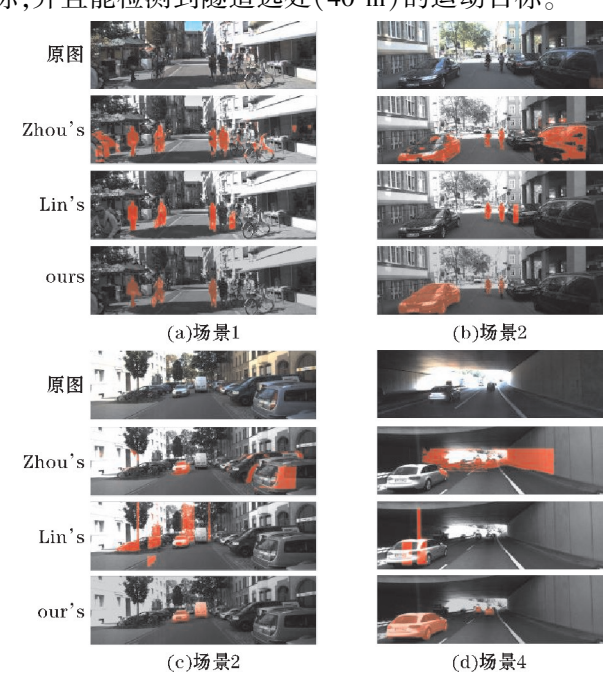
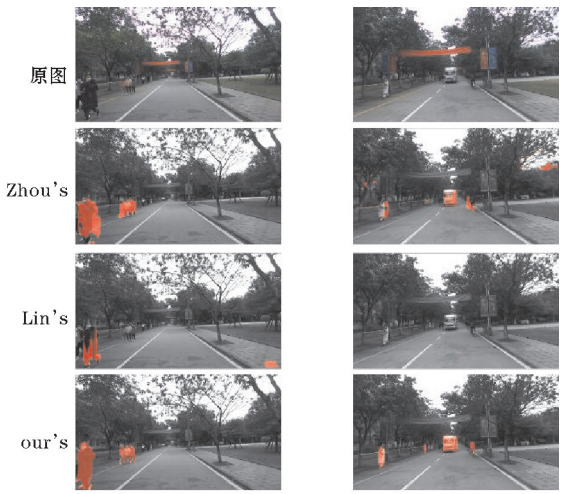


图 10 挑战场景

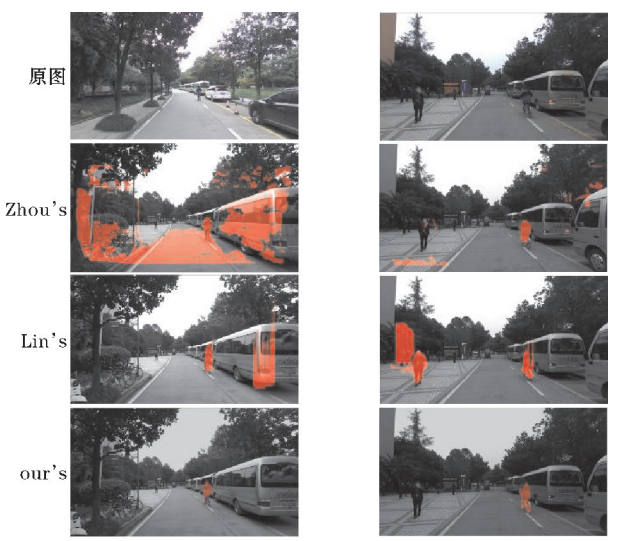
2.1.3　本地数据集效果展示

为充分展示本文所提方法的有效性,选取采集的校园数据集进行测试,并与其他方法对比。实验结果如图 11 和图 12 所示,实验结果表明本文算法能更好识别运动目标。



(a)场景1　(b)场景2

图 11　校园场景 1



(a)场景1　(b)场景2

图 12　校园场景 2

2.2　定量分析

运动目标检测是一个典型的二分类问题。设 TP 表示正确检测到的运动目标的数量;FP 表示原本为静止的目标,但是被算法误检为运动目标的数量;FN 表示漏检的运动目标数量;TN 表示原本为静止目标,检测结果仍然为静止场景。

为定量地分析算法的性能,评价指标采用精度 P,召回率 R 与 F。

$$P=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}},R=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},F=\frac{2R\cdot P}{R+P} \tag{9}$$

由式(9)可得知,精度 P 指的是真正为运动目标的数量与所有被检测为运动目标数量的比值;召回率 R 指的是运动目标被成功检测出来的数量与所有真正为运动目标的数量的比值。对精度和召回率两个公式进行分析,当阈值被设置得更高时,精度也会跟着提高;但是由于阈值比较高,便会丢失大量数据,将导致召回率下降。相反,如果将阈值设置得更低,精度会跟着降低;但是,由于测试结果中存在大量错误检测,将导致召回率增大。根据上述分析,准确率和召回率往往是矛盾的,因此,可以通过 P 和 R 的调和平均值即 F 来评估。

结果如表 1 和表 2 所示,实验结果表明,相较于固定阈值,本文所提独立自适应阈值方法能够在提升检测精度的同时,减少误检测的发生。同时,相较于 Zhou 和 Lin 的方法,本文所提方法召回率和精度都有大幅提高。这表明本文提出的方法能够提升运动目标的检测精度,降低误检测率,提升检测系统的鲁棒性。

表 1　KITTI1 数据集结果对比

KITTI1	P	R	F
Zhou′s	0.697	0.721	0.708
Lin′s	0.227	0.492	0.312
OURS+固定阈值(0.6)	0.703	0.952	0.809
OURS+固定阈值(0.2)	0.906	0.844	0.874
OURS+独立自适应阈值	0.923	0.944	0.933

表 2　KITTI2 数据集结果对比

KITTI2	P	R	F
Zhou′s	0.610	0.767	0.680
Lin′s	0.452	0.823	0.598
OURS+固定阈值(0.2)	0.812	0.805	0.808
OURS+独立自适应阈值	0.874	0.951	0.911

3　结束语

提出一种基于深度学习与稀疏场景流结合的运动目标检测方法。通过深度学习分割出潜在运动区域,这既能避免该区域对相机自运动参数估计的影响又能完成对目标的分割。基于场景流的估计原理,考虑到场景流易受噪声和匹配误差影响,到不同目标场景流估计误差的不同,提出独立自适应阈值方法进行运动目标判别。实验结果表明,所提独立自适应阈值相比固定阈值能够提升系统鲁棒性;同时与其他经典算法相比,在多种场景中检测效果也有明显提升。下一步计划优化特征点提取方法确保每个目标上都有足量的特征点可用于运动检测。

参考文献:

[1]　Radke R J, Andra S, Al-Kofahi O, et al. Image

change detection algorithms: a systematic survey [J]. IEEE transactions on image processing, 2005, 14(3): 294-307.
[2] 何楠楠, 杜军平. 智能视频监控中高效运动目标检测方法研究[J]. 北京工商大学学报(自然科学版), 2009, 27(4): 34-37.
[3] 巨志勇, 彭彦妮. 基于自动背景提取及 Lab 色彩空间的运动目标检测[J]. 软件导刊, 2018, 17(5): 183-186.
[4] 马波, 张田文. 基于 AOS 的运动目标检测算法[J]. 计算机辅助设计与图形学学报, 2003, 15(10): 1213-1217.
[5] 王春兰. 智能视频监控系统中运动目标检测方法综述[J]. 自动化与仪器仪表, 2017(3): 1-3.
[6] 王恩旺, 王恩达. 改进的帧差法在空间运动目标检测中的应用[J]. 天文研究与技术, 2016, 13(3): 333-339.
[7] 金玥佟, 杨耀权, 杜永昂. 电力监控场景下基于光流特征点的目标跟踪算法[J]. 电力科学与工程, 2020, 36(5): 40-47.
[8] Esparza D, Flores G. The STDyn-SLAM: A Stereo Vision and Semantic Segmentation Approach for VSLAM in Dynamic Outdoor Environments [J]. IEEE Access, 2022, 10: 18201-18209.
[9] Liu G, Zeng W, Feng B, et al. DMS-SLAM: A general visual SLAM system for dynamic scenes with multiple sensors[J]. Sensors, 2019, 19(17): 3714.
[10] Zhou D, Frémont V, Quost B, et al. Moving object detection and segmentation in urban environments from a moving platform [J]. Image and Vision Computing, 2017, 68: 76-87.
[11] Lin S F, Huang S H. Moving object detection from a moving stereo camera via depth information and visual odometry [C]. 2018 IEEE International Conference on Applied System Invention (ICASI). IEEE, 2018: 437-440.
[12] Chen L, Fan L, Xie G, et al. Moving-object detection from consecutive stereo pairs using slanted plane smoothing[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(11): 3093-3102.
[13] Cui L, Ma C. SOF-SLAM: A semantic visual SLAM for dynamic environments [J]. IEEE access, 2019, 7: 166528-166539.
[14] Ballester I, Fontán A, Civera J, et al. DOT: Dynamic object tracking for visual SLAM[C]. 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021: 11705-11711.
[15] Kaneko M, Iwami K, Ogawa T, et al. Mask-slam: Robust feature-based monocular slam by masking using semantic segmentation [C]. Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018: 258-266.
[16] Huang J, Yang S, Mu T J, et al. Clustervo: Clustering moving instances and estimating visual odometry for self and surroundings[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 2168-2177.
[17] Wang X, Zhang R, Kong T, et al. Solov2: Dynamic and fast instance segmentation [J]. Advances in Neural information processing systems, 2020, 33: 17721-17732.

Detection of Moving Objects in Smart Cars based on Binocular Sparse Scene Flow

LIU Mingwen, JIANG Tao, YUAN Jianying*, GU Shuoxin, XU Zhiyong, LEI Ting
(College of Automation, Chengdu University of Information Technology, Chengdu 610025, China)

Abstract: In order to improve the accuracy of visual moving object detection of unmanned vehicles, this paper proposes a moving object detection method based on deep learning combined with sparse scene flow. First, the deep learning network SOLO V2 is used to segment the traffic scene, extract potential moving targets such as pedestrians and vehicles, and narrow the search range of moving targets in the scene. Secondly, the camera self-motion parameters are calculated by using the feature matching points in the background. On this basis, the feature points of the potential motion area are mapped to a unified coordinate system. This feature point is at two moments before and after, and then the sparse scene flow generated only by its own motion is calculated. Finally, according to the difference of the estimation error of each target scene flow, the uncertainty of each target scene flow estimation is calculated and an independent adaptive threshold is set for each target for motion state judgment. The KITTI data set is used to test. The experimental results show that the proposed algorithm can significantly improve the accuracy of moving target detection. The accuracy and recall of the algorithm are 92.3%, 94.4% and 87.4%, 95.1% respectively in the two sets of data.

Keywords: driverless car; moving object detection; scene flow; adaptive threshold