第 46 卷 第 2 期 计 算 机 工 程 2020 年 2 月
Vol. 46 No. 2 Computer Engineering February 2020

·图形图像处理· 文章编号: 1000-3428(2020)02-0230-05 文献标志码: A 中图分类号: TP391

基于 YOLOv3 的嵌入式实时视频目标检测算法

尹彦卿,龚华军,王新华

(南京航空航天大学 自动化学院,南京 210000)

摘 要: 深度神经网络在目标检测领域具有优异的检测性能,但其结构复杂、计算量大,难以在嵌入式设备上进行高性能的实时目标检测。针对该问题,提出一种基于 YOLOv3 的目标检测算法。采用半精度推理策略提高 YOLO 算法的推理速度,并通过视频运动自适应推理策略充分利用前后帧视频之间目标的关联性,降低深度学习算法的运行频率,进一步提高目标检测速度。在 ILSVRC 数据集上的实验结果表明,该算法可以在 NVIDIA TX2 嵌入式平台上实现 28 frame/s 的视频目标检测,且检测精度与原始的 YOLOv3 算法相当。

关键词: YOLOv3 算法; 深度学习; 目标检测; NVIDIA TX2 嵌入式平台; 半精度; 粒子滤波

开放科学(资源服务)标志码(OSID):

中文引用格式: 尹彦卿,龚华军,王新华. 基于 YOLOv3 的嵌入式实时视频目标检测算法[J]. 计算机工程,2020,46(2):230-234.

英文引用格式: YIN Yanqing, GONG Huajun, WANG Xinhua. Embedded real-time video object detection algorithm based on YOLOv3[J]. Computer Engineering, 2020, 46(2): 230-234.

Embedded Real-time Video Object Detection Algorithm Based on YOLOv3

YIN Yanqing, GONG Huajun, WANG Xinhua

(College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210000, China)

【Abstract】 Despite the outstanding performance of deep neural network in object detection, it is hard to implement high-performance real-time object detection on embedded devices due to the complex structure and large amounts of required computation. To address the problem, this paper proposes a YOLOv3-based object detection algorithm. The algorithm uses half precision inference strategy to accelerate the inference of YOLO algorithm. Another inference strategy adaptive to video motions is also adopted to use object correlation between adjacent frames to decrease the running frequency of the deep learning algorithm, and further improve the speed of object detection. Experimental results on the ILSVRC dataset show that the proposed algorithm can implement video object detection on NVIDIA TX2 embedded platforms at a speed of 28 frame/s, and its detection accuracy is close to that of the original YOLOv3.

【Key words】 YOLOv3 algorithm; deep learning; object detection; NVIDIA TX2 embedded platform; half precision; particle filter

DOI: 10.19678/j.issn.1000-3428.0053584

0 概述

目标检测是指从场景中区分出不同对象,定位每个对象的边界框并判别其类型的技术。如何实现精确的目标检测是计算机视觉领域一个基础性难题,引起了研究人员的广泛关注。

随着深度学习技术的发展,许多基于卷积神经网络(Convolutional Neural Network, CNN)的目标检测方法都已具备优越的检测性能。例如,特征金字塔网络(Feature Pyramid Network, FPN)算法在 coco 数据集上的检测精度达到 59.1%,但该算法由于网络参数繁多、计算复杂,即使在 Titan GPU 上,其检测速度也仅能达到 6 frame/s,难以实现实时目标检测[1]。

2018 年,YOLOv3 目标检测算法被提出,它将目标检测视作一个回归问题,通过一个神经网络同时输出目标的位置和类别,在几乎不损失识别精度的前提下,其检测速度明显提升。在 Titan-X GPU 上,

基金项目: 中国博士后科学基金(2016M591845)。

作者简介: 尹彦卿(1994—),男,硕士研究生,主研方向为计算机视觉、深度学习、嵌入式开发;龚华军,教授、博士生导师;王新华,副教授。

收稿日期: 2019-01-07 **修回日期**: 2019-02-10 **E-mail**: 977380289@qq.com

YOLOv3-416 网络的运行速度达 35 frame/s,可用于实现对视频信息的实时检测[2-4]。

尽管 YOLOv3 在强大的 GPU 上具有实时检测性能,但高端 GPU 价格昂贵、体积巨大,通常仅能在服务器上安装,极大地限制了 YOLO 算法的应用场景。在智能手机、无人机、无人车等常见的嵌入式设备上,计算资源和内存都非常有限,因此,嵌入式设备上的实时目标检测仍是一大挑战。YOLOv3 的研究者还提出了一种 tiny-YOLOv3 网络,其网络结构简单,可在嵌入式设备上进行实时检测,但其 mAP 值仅为 33.1%,无法兼顾实时性和检测准确度[5]。

本文提出一种基于 YOLOv3 的嵌入式平台实时视频目标检测算法。该算法运用半精度推理策略,在几乎不损失检测精度的情况下提高检测速度,同时,利用视频目标运动自适应推理策略,降低深度网络推理的运行频率,进一步提高视频目标检测的帧率。结合上述 2 种策略,在 NVIDIA Tegra 系列嵌入式 GPU 平台上实现实时、准确的目标检测[6-7]。

1　YOLOv3 网络的半精度推理

1.1　半精度推理的可行性分析

为了保证计算精度,目前主流算法框架的深度网络权值均使用单精度浮点数据类型进行保存。单精度浮点类型在计算机中较为常见,但其在存储时需要 4 Byte 的空间,且计算复杂度较高。在 IEEE 754 标准中定义了一种半精度浮点类型[8],其在 CUDA 编程环境中又称为 Half 类型。半精度浮点类型需要 2 Byte 存储空间,其优势在于 GPU 中的计算硬件可在一个周期内同时完成 2 个半精度浮点类型的运算,大幅提高计算速度。2 种数据类型的比较如表 1 所示。

表 1　单精度与半精度类型的比较

Table 1　Comparison of single precision and half precision types

数据类型	存储空间/Byte	精度	数值范围
单精度	4	1.40×10^{-45}	-3.4×10^{38} ~ $+3.4\times10^{38}$
半精度	2	5.96×10^{-8}	−65 504 ~ +65 504

由表 1 可知,2 种数据类型的性能差异主要体现在数值范围与精度上。在数值范围方面,根据本文在 coco 数据集上的测试结果,YOLOv3 网络权值和激活值普遍分布在 ±20 之内,故 2 种数据类型的数值范围均能满足要求。在精度方面,由于网络训练时反向传播算法需要计算权值的梯度,权值的一点微小变化都会对结果产生较大影响,因此在训练过程中,权值的精度至关重要,只能使用单精度类型储存。在本文算法的网络推理过程中,推理速度是最重要的影响因素,因此,半精度类型权值的精度损失对网络推理结果的影响程度,将直接决定能否使用半精度推理策略进行 YOLO 网络加速。

将待检测的目标类别数记作 C,则 YOLO 网络对每个检测目标的输出结果为一个五元组(c,x,y,w,h) 和一个长度为 C 的向量(P_1,P_2,…,P_C),其中,P_i 表示目标属于第 i 类的分类概率,c 表示目标存在的概率,(x,y,w,h) 表示目标的位置和大小,其含义是实际目标框相对于一个固定预设框的位置偏移和长宽比例,而不是目标框的绝对坐标。实际输出目标框相对于预设框的位置关系如图 1 所示。

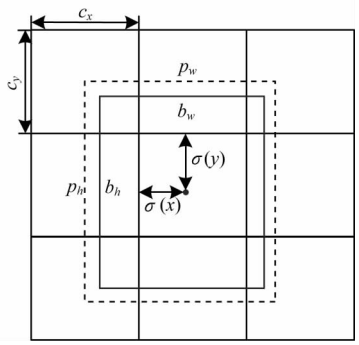


图 1　实际目标框相对于预设框的位置和大小

Fig. 1　Position and size of the actual target frame relative to the preset frame

令每个网格距离图片左上角的像素距离分别为 c_x、c_y,每个预设框的宽高为 p_w、p_h,σ 函数表示将范围在 0 ~ 1 的输入线性映射到网格的实际长宽范围,则物体实际在图像中的位置和大小如下:

$$\sigma(x) = 32x$$
$$b_x = \sigma(x) + c_x$$
$$b_y = \sigma(y) + c_y$$
$$b_w = p_w e^w$$
$$b_h = p_h e^h$$

本文在 coco 数据集上分别进行单精度和半精度推理实验。根据实验结果的统计数据,对于 YOLO 网络最终输出的激活值,使用半精度类型运算与单精度类型运算的误差在 ±0.001 以内。下面将逐一分析该误差对网络输出变量的影响:

1) 目标存在概率 c。本文算法以 0.5 为阈值判断是否存在目标,将低于 0.5 的值判定为背景。c 的取值范围为 0 ~ 1,因而在绝大多数情况下,该误差不足以影响 c 相对于阈值的关系,其对判定结果基本没有影响。

2) 目标分类概率 P_i。算法通过 argmax 函数选取其中最大值的下标 i 作为目标分类的结果。P_i 的分布范围为 0 ~ 1,而正确的目标分布概率普遍在 0.9 以上。因此,该误差不能改变最大值的分布,因而可认为该误差对于目标分类结果几乎没有影响。

3) 目标位置(x,y)。经过 σ 函数线性映射后,目标位置(x,y) 的误差在 0.032 像素以内,而该误差经过指数函数放大后,其对目标大小(w,h) 产生的误差也在 1 像素以内。因此,可认为该误差对于目标定位位置影响较小。

综上所述,使用半精度类型取代单精度类型进行网络权值和激活值的表示,对于网络输出结果基

本没有影响，但可以大幅加快推理速度和降低存储要求，便于在嵌入式平台上进行推理。

1.2 TX2 平台上半精度推理的实现

需要注意的是，并非所有嵌入式设备均支持半精度浮点类型运算。自 Tegra X1 以后，NVIDIA 的嵌入式 GPU 平台也支持原生的半精度计算指令，因此，本文选用 Tegra 系列的最新处理器 TX2 进行 YOLOv3 网络半精度推理的实现。TX2 处理器的核心板大小仅为 5 cm ×8 cm，但其能够为深度学习应用提供超过 1 Tflops 的浮点运算性能，同时功耗低于 7.5 W，故适合在视频监控摄像头、无人机、机器人等嵌入式平台上部署应用[9]。

在进行 CUDA 编程时，需要使用内置的 Half 2 类型对半精度运算进行加速。Half 2 类型为 2 个半精度类型在显存中的组合，其结构如图 2 所示。其中，S 为符号位，E 为阶码，M 为尾数。Half 2 类型在计算时将 2 个半精度类型同时输入 32 位 GPU，可在一个周期内同时完成 2 个半精度浮点类型的运算。

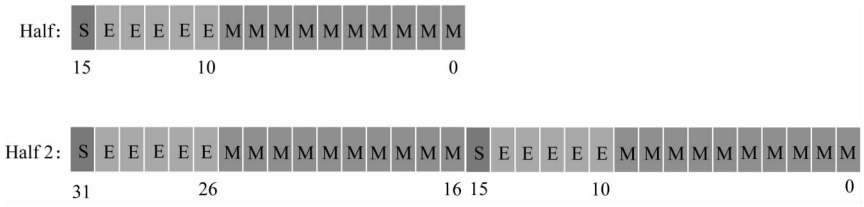


图 2 Half 2 类型在显存中的存储结构

Fig. 2 Storage structure of the Half 2 type in video memory

由于需要 2 个半精度类型同时运算才能起到加速效果，因此在 YOLO 网络推理时要尽量保证所有的操作均为内存中相邻两组操作数的计算，避免计算资源的浪费[10]。在嵌入式 GPU 平台上使用半精度类型进行推理的主要过程如下[11]：

1) 读入待检测的图像数据和网络权值数据，送入显存，并在 GPU 中转换成半精度类型。

2) 使用 CUDNN 提供的卷积运算接口 API 进行卷积计算，在设置参数时，将数据类型选为半精度类型[12]。

3) 进行偏置、正则化等计算。首先将参数在显存中拷贝一份，然后使用 Half 2 类型进行计算，以达到最大的加速效果。以偏置计算为例，其 Half 2 类型的快速计算过程如图 3 所示。

4) 在 GPU 中将计算结果转换成单精度类型，并传回内存中由 CPU 进行最终处理，然后输出检测结果。

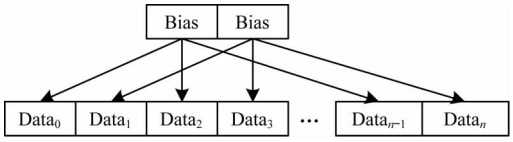


图 3 Half 2 类型的快速偏置计算

Fig. 3 Fast offset calculation of Half 2 type

2 视频运动自适应推理

考虑到本文算法主要应用于视频目标检测，目标信息在视频中的相邻帧之间具有连续性，因而并不需要对每一帧图像进行完整的深度网络推理。为进一步提高视频目标检测的速度，本文引入运动自适应推理的策略来判断视频帧是否需要重新进行完整的深度推理[13]。

对于新输入的视频帧，使用三帧差分法进行运动估计[14]。当前帧、上一帧、上上帧的图像分别记为 I_t、I_{t-1}、I_{t-2}，则三帧差分的结果如下：

$$M_{\text{movement}} = (I_t - I_{t-1}) \cap (I_{t-1} - I_{t-2})$$

对三帧差分法得到的运动图进行阈值化、形态学处理和滤波后，即可得到图像中的运动区域。对于相对上一帧运动较大的视频帧，需要运行一次完整的深度推理，重新检测图像中的目标位置和分类；对于相对上一帧运动不大的视频帧，则使用粒子滤波来进行视频目标跟踪，并更新之前检测出的目标位置。考虑到粒子滤波器在进行长时间的目标跟踪时容易出现漂移、跟踪失败的情况，设置强制深度推理间隔为 m，即在进行 m 帧的粒子滤波后，对第 $m+1$ 帧强制使用深度推理重新检测目标位置。每次完成深度推理后，将深度网络的检测结果作为粒子滤波器的初始值，重新初始化粒子滤波器[15-17]。整个过程如图 4 所示。

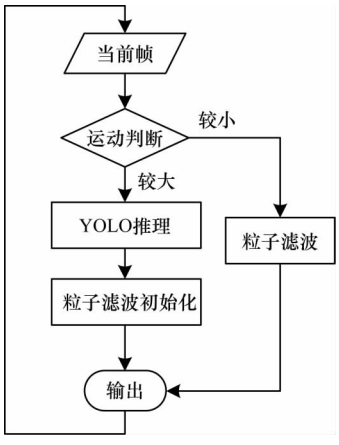


图 4 视频运动自适应推理流程

Fig. 4 Adaptive inference proceedure of motions in video

在 TX2 平台上进行测试后发现，GPU 加速后的三帧差分法耗时 5 ms，对单个目标进行粒子滤波跟踪耗时 3 ms，比进行一帧深度推理的耗时要少。当视频中的目标个数较少，例如，将帧差法与粒子滤波的

耗时忽略不计时,整个算法运行帧率的上限为深度推理运行帧率的 m 倍。根据本文的测试结果,m 取 1 ~ 3 时较为合适。当 m 取值较大时,算法的帧率上限有所提升,但其输出对于粒子滤波结果的依赖性较强,准确度有所下降。若视频中的目标个数较多,对目标逐个进行粒子滤波跟踪的耗时增大,在这种情况下,可考虑使用 CamShift 等相对更快的视频目标跟踪算法降低耗时,进一步改善算法的实时性[18]。

3　实验结果与评估

本文算法的所有评估工作均在 NVIDIA TX2 嵌入式平台上进行,处理器运行于高性能模式,软件环境为 CUDA 9.0、CUDNN 7.1。本文利用 coco 数据集对不同输入尺度下单精度网络和半精度网络的单帧图片推理性能进行测试。由于 coco 数据集中全部为静态图像,因此仅启动半精度推理加速策略,视频目标跟踪策略没有明显的效果[19],结果如表 2 所示。

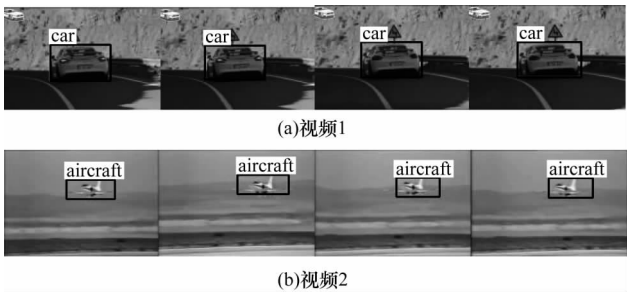
表 2　不同输入尺度下单精度与半精度推理性能的对比

Table 2　Comparison of single-precision and half-precision inference performance at different input scales

网络类型	输入尺度/像素	mAP/%	耗时/ms	帧率/(frame·s^{-1})
tiny-YOLOv3	416×416	33.1	35	28.6
单精度 YOLOv3	320×320	51.5	125	8.0
	416×416	55.3	350	2.9
	608×608	57.9	550	1.8
半精度 YOLOv3	320×320	51.2	45	22.2
	416×416	55.1	105	9.5
	608×608	57.6	190	5.3

由表 2 可以看出,在不同的输入尺度下,本文的半精度推理策略相对于其他 2 种策略有 3 倍左右的加速效果,同时其 mAP 值损失不到 0.3%,检测精度明显优于 tiny-YOLO 和单数度 YOLOv3 实时目标检测算法,使 YOLO 算法在嵌入式设备上的推理性能大幅提升。

本文选用 ILSVRC 视频目标检测(VID)数据集对算法的性能进行测试[20]。该数据集包含数千个完全标注的视频片段,每个片段的长度为 56 帧 ~ 458 帧。在 2 段视频中各截取 4 帧相邻图片,如图 5 所示。



(a)视频1　(b)视频2

图 5　ILSVRC VID 测试序列中的检测结果

Fig. 5　Detection results in the ILSVRC VID test sequence

在图 5 中,第 1 帧与第 4 帧图片的检测结果由深度网络推理得到,中间 2 帧图片的检测结果为粒子滤波跟踪得到。可以看出,如果视频画面中的目标运动不太明显,通过粒子滤波能够可靠地跟踪深度网络检测到的目标位置,降低深度推理的运行频率,大幅节省计算量。在完整的 ILSVRC VID 测试数据集上,本文算法与原始 YOLO 算法的实验结果对比如表 3 所示。

表 3　ILSVRC VID 数据集上的视频目标检测结果对比

Table 3　Result comparison of video object detection on the ILSVRC VID dataset

算法	深度推理间隔	深度推理频率/Hz	帧率/(frame·s^{-1})	mAP/%
YOLOv3	0	100	2.9	63.2
本文算法	0	100	9.5	63.0
	1	52	18.5	62.9
	2	34	28.2	62.8

从表 3 可以看出,本文的嵌入式实时视频目标检测算法在深度推理间隔为 2 时,可通过视频目标运动自适应推理策略将深度推理运行次数减少 66%,结合半精度推理策略,可在 TX2 嵌入式平台上将运行帧率提高至 28.2 frame/s,同时 mAP 值损失很低,检测精度仍与原始 YOLOv3 网络相当。增大深度推理间隔可得到更高的帧率,但也会导致目标检测精度降低。

考虑到 ILSVRC VID 数据集中的测试序列长度普遍较短,且在同一个测试序列中出现的目标个数较少,大部分为单个目标面积较大的视频序列,其目标检测难度较低,因此,本文选取一段高速公路监控摄像头拍摄的视频进行复杂场景下的多目标检测实验。从测试视频中截取 2 帧相邻的画面,检测结果如图 6 所示。由图 6 可以看出,在复杂场景下,当视频中出现多个目标且每个目标的面积均较小时,本文算法依然具有良好的检测效果,可以应用于实际工程中。

(a)视频帧1　(b)视频帧2

图 6　测试序列中的视频目标检测结果

Fig. 6　Video object detection results in the test sequence

4　结束语

本文提出一种基于 YOLOv3 的嵌入式平台实时视频目标检测算法。通过半精度推理策略和视频目标运动自适应推理策略改进原始 YOLO 算法在嵌入式设备上的性能,提高算法的检测速度。实验结果表

明,该算法在 NVIDIA TX2 嵌入式平台上进行单次推理的速度比原始 YOLO 算法快 3 倍以上,其在 ILSVRC VID 数据集上可实现 28.2 frame/s 的实时检测,且检测精度与 YOLOv3 算法几乎相同。下一步将继续优化 YOLO 算法在嵌入式平台上的部署策略,以实现深度学习目标检测算法在更多设备上的应用。

参考文献

[1] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]//Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2017: 936-944.

[2] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2016: 779-788.

[3] REDMON J, FARHADI A. YOLO9000: better, faster, stronger[C]//Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2017: 6517-6525.

[4] REDMON J, FARHADI A. YOLOv3: an incremental improvement[C]//Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2018: 1-6.

[5] MA Jing, CHEN Li, GAO Zhiyong. Hardware implementation and optimization of tiny-YOLO network[J]. Digital TV and Wireless Multimedia Communication, 2017, 815: 224-234.

[6] GAO Zong, LI Shaobo, CHEN Jinan, et al. Pedestrian detection method based on YOLO network[J]. Computer Engineering, 2018, 44(5): 215-219, 226. (in Chinese)
高宗,李少波,陈济楠,等. 基于 YOLO 网络的行人检测方法[J]. 计算机工程,2018,44(5):215-219,226.

[7] GONG Jing, CAO Li, QI Lin, et al. Moving vehicle target detection based on YOLOv2 algorithm[J]. Electronic Science and Technology, 2018, 31(6): 5-8, 12. (in Chinese)
龚静,曹立,亓琳,等. 基于 YOLOv2 算法的运动车辆目标检测方法研究[J]. 电子科技,2018,31(6):5-8,12.

[8] IEEE. IEEE standard for floating-point arithmetic: 754-2008[S]. Washington D. C., USA: IEEE Press, 2008.

[9] QI Jian. NVIDIA Jetson TX2 platform: accelerating the development of miniaturized artificial intelligence terminals[J]. Intelligent Manufacturing, 2017(5): 20-21. (in Chinese)
齐健. NVIDIA Jetson TX2 平台:加速发展小型化人工智能终端[J]. 智能制造,2017(5):20-21.

[10] AMERT T, OTTERNESS N, YANG M, et al. GPU scheduling on the NVIDIA TX2: hidden details revealed[C]//Proceedings of 2017 IEEE Real-Time Systems Symposium. Washington D. C., USA: IEEE Press, 2017: 104-115.

[11] LUSZCZEK P, KURZAK J, YAMAZAKI I, et al. Towards numerical benchmark for half-precision floating point arithmetic[C]//Proceedings of High Performance Extreme Computing Conference. Washington D. C., USA: IEEE Press, 2017: 1-5.

[12] LIU Yong. Analysis of general scientific computing using GPU acceleration-CUDA technology[J]. Science and Technology Information, 2008(24): 396, 413. (in Chinese)
刘勇. 使用 GPU 加速通用科学计算-CUDA 技术解析[J]. 科技信息,2008(24):396,413.

[13] SHAFIEE M J, CHYWL B, LI F, et al. Fast YOLO: a fast you only look once system for real-time embedded object detection in video[EB/OL]. [2018-12-24]. https://arxiv.org/pdf/1709.05943.pdf.

[14] XIONG Ying. Extraction of moving objects based on background and inter-frame difference method[J]. Computer Era, 2014(3): 38-41. (in Chinese)
熊英. 基于背景和帧间差分法的运动目标提取[J]. 计算机时代,2014(3):38-41.

[15] LI Yuanzheng, LU Zhaoyang, GAO Quanxue, et al. Particle filter and mean shift tracking method based on multi-feature fusion[J]. Journal of Electronics and Information Technology, 2010, 32(2): 411-415. (in Chinese)
李远征,卢朝阳,高全学,等. 基于多特征融合的均值迁移粒子滤波跟踪算法[J]. 电子与信息学报,2010,32(2):411-415.

[16] DAI Dingzhang. Research on particle filter algorithm and its application in target tracking[D]. Harbin: Harbin Institute of Technology, 2006. (in Chinese)
戴丁樟. 粒子滤波算法研究及其在目标跟踪中的应用[D]. 哈尔滨:哈尔滨工业大学,2006.

[17] OKUMA K, TALEGHANI A, FREITAS N D, et al. A boosted particle filter: multitarget detection and tracking[C]//Proceedings of European Conference on Computer Vision. Berlin, Germany: Springer, 2004: 28-39.

[18] WANG Xin, TANG Zhenmin. An improved Camshift-based particle filter algorithm for real-time target tracking[J]. Journal of Image and Graphics, 2010, 15(10): 1507-1514. (in Chinese)
王鑫,唐振民. 一种改进的基于 Camshift 的粒子滤波实时目标跟踪算法[J]. 中国图象图形学报,2010,15(10):1507-1514.

[19] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft COCO: common objects in context[C]//Proceedings of 2014 European Conference on Computer Vision. Berlin, Germany: Springer, 2014: 740-755.

[20] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2014, 115(3): 211-252.

编辑 樊丽娜