

基于 HOG 的快速行人计数算法^①

李峰松¹, 张运楚^{1,2}, 杨红娟^{1,2}, 王业篷¹

¹(山东建筑大学 信息与电气工程学院, 济南 250101)

²(山东省智能建筑技术重点实验室, 济南 250101)

摘要: 为解决智能监控中空间受限情景下的行人计数问题, 设计了基于 HOG 的行人识别与计数算法. 通过限制 HOG 目标检测算法中图像的缩放范围, 并设置感兴趣的检测区域, 相比直接应用 OpenCV 函数实现的目标检测, 其时间效率大幅提升; 采用基于单计数线算法完成双向流量统计. 三个不同走廊环境的实验表明, 本文算法具有不依赖帧间运动信息的特点, 可快速稳定的对空间受限情景下的行人进行计数.

关键词: 智能监控; 目标检测; 行人计数; HOG

Fast Pedestrians Counting Algorithm Based on HOG

LI Feng-Song¹, ZHANG Yun-Chu^{1,2}, YANG Hong-Juan^{1,2}, WANG Ye-Peng¹

¹(School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan 250101, China)

²(Shandong Provincial Key Laboratory of Intelligent Buildings Technology, Shandong Jianzhu University, Jinan 250101, China)

Abstract: A rapid pedestrian detection and counting algorithm based on HOG is proposed for getting the pedestrian flow passing the confined space. The time efficiency is increased remarkable by restricting window shrink range and setting ROI. Counting method based on single line is presented for getting tow directions people number. Experiments in three different passageway show that the proposed method don't depend on moving information, besides it has the ability of fast and stable in confined space pedestrian counting.

Key words: intelligent monitoring; object detection; pedestrian counting; HOG

空间受限情景是指形如建筑物走廊或出入口等室内空间, 行人受环境限制只能沿特定方向往返的情况. 对此类情景下的行人流量进行统计, 可以为商场、展馆、交通枢纽等公共场所的日常运行和应急处理提供决策依据. 基于图像处理技术的行人计数方法因具有灵活高效的特点而逐渐成为研究热点.

行人计数算法是根据摄像机的视角配置方式而设计的, 可以划分为基于垂直观察视角^[1-3]和基于倾斜观察视角^[4-6]的两类方法. 目前对前者的研究已经相对成熟, 然而垂直观察视角具有监控范围狭窄及视角陌生的不足, 降低了视频监控的使用价值, 导致这类算法在智能监控中很少使用. 本文研究基于倾斜视角的行人计数算法, 系统环境设置如图 1 所示, 摄像机吊顶安装在走廊上方俯拍走廊环境.

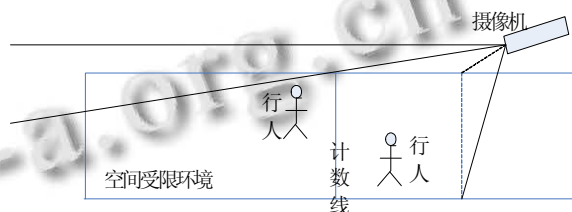


图 1 环境设置示意图

目前的研究有基于帧间信息的运动目标检测的拌线计数方法^[4], 这类方法具有良好的实时性, 但对目标的运动性依赖强, 且易受光照突变的干扰; 另有基于帧内信息的静态图像目标检测计数方法^[5], 该类方法不依赖运动信息且对光照变化不敏感, 其中较为经典的目标检测算法是 Dalal^[7]的基于方向梯度直方图 (HOG- Histograms of Oriented Gradients) 的算法, 它是

① 基金项目: 国家自然科学基金(61303087); 住房和城乡建设部科学技术项目(2011-K9-20)

收稿时间: 2013-10-09; 收到修改稿时间: 2013-10-31

一种优异的滑动窗搜索检测算法,但算法复杂度高导致系统实时性差,围绕着实时性的改进,文章^[8-9]通过积分直方图计算 HOG 特征并构建级联分类器,在检测精度略有下降的情况下提高了检测速度.文章^[10]采用并行技术,通过 GPU 加速,构建了一个实时行人检测系统.

本文提出了一种基于 HOG 行人检测的快速计数算法,在不影响检测精度的前提下充分利用空间受限这一客观条件,在足以满足计数线计数需求的情况下限制图像的缩放尺度并设置检测带控制搜索窗的空间搜索范围,从而减少了 HOG 算法中搜索窗的运算数量,提升了行人检测的时间效率.计数方面提出一种基于单计数线的空间位置判别法,具有对过往行人的双向计数能力.实验证明本文所提方法具备快速稳定的计数能力,且在行人间互遮挡不严重时检测计数精度较高.

1 算法原理

1.1 基于 HOG 的行人检测

HOG 描述子计算是通过统计每个像素的梯度方向直方图实现的,如图 2 所示,直方图统计的最小单元是 Cell,计算 Cell 内各像素点的梯度方向并将无符号角度范围 $0-180^\circ$ 离散为 9 个分量统计直方图,即从每个 Cell 中提取 9 维的特征向量.以 Block 为单位进行归一化处理从而降低局部光线变化的干扰,通常相邻 Block 有 $1/2$ 的重叠,且每个 Block 包含 4 个 Cell,进而提取的特征向量为 36 维,最终串联所有 Block 的特征向量构成的高维向量就是当前搜索窗样本的 HOG 描述子.

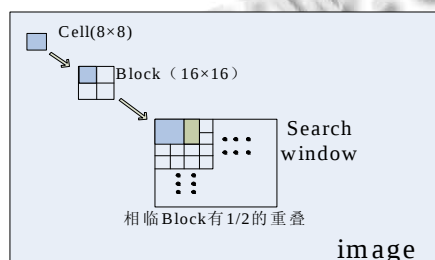


图 2 HOG 描述子

基于 HOG 的行人检测是采用滑动搜索窗的二值分类过程.首先采集大量正负样本,归一化到共同大小后提取各自的 HOG 描述子,并对 SVM(Support

VectorMachine, 支撑向量机)分类器进行分类模型的训练.然后对待检图像采用滑动搜索窗进行多尺度检测,即采用与归一化后的样本等大小的矩形窗(搜索窗,远小于待检测图像)覆盖到图像中,提取覆盖区域像素的 HOG 描述子,由训练好的 SVM 分类器对其进行人体与非人体的二值判决.为兼顾图像中任意位置及不同大小的行人的检测,搜索窗先以较小步长纵横滑动遍历图像任意区域,然后按某一比例将图像缩小后继续用该搜索窗进行遍历搜索,直到图像被逐级缩小到近似窗体大小为止.由于窗体的密集遍历会导致同一目标行人同时被多个矩形包围盒包围的情况,如图 3 左所示,因此需对包围盒按位置进行聚类分析,对属于同一类的窗体求平均融合为一个包围盒,如图 3 右所示,最后获得检测到的行人目标位置及区域的包围盒.

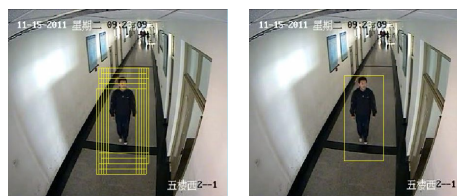


图 3 窗口融合

由于检测过程近似“穷搜”过程,计算量大,耗时多,影响了检测过程的实时性.而实际检测过程中行人的尺度变化及空间存在范围有限且可事先获知,因此本文首先根据计数线前后行人尺度的变化规律,直接将图像缩小到搜索窗刚好包围检测线附近最大行人的尺寸上,并总结出在这一尺寸上按 1.05 比例逐级缩放 4 次即可满足计数过程中对行人检测的需要这一规律,将搜索窗的搜索过程集中在最有效的 5 幅子图像中进行,从而显著降低了搜索窗的运算数量.其次在计数线附近隐式的设置检测计数区域,将搜索窗的空间搜索范围约束在计数线附近以进一步提高系统的时间效率.

1.2 行人跟踪及跨越计数线判别

为避免长距离跟踪的不稳定性,本文跟踪过程选择计数线附近 20 像素以内进行.跟踪方法采用目标质心点的距离匹配,即根据相邻两帧同一目标质心位置的偏移距离有限这一事实,在第 n 帧图像中以第 $n-1$ 帧图像中的目标质心位置为中心小范围搜索,所搜到的质心与第 $n-1$ 帧中的为同一目标.

跨线计数首先通过对目标质心点的实时跟踪,判断目标是否跨越计数线.如图 4 所示,计数线 H 是一条线段,将二维平面分割成 I、II 两个区域,通过判断行人质心点 P 的位置在不同时刻其所属区是否发生变化来判断行人是否跨线.其关键是质心点的空间区域判别,借助模式识别中的线性判别函数的概念,用直线方程的一般式生成线性判别函数 $D=Ax+By+C$,将质心点的空间坐标 $P(x,y)$ 带入判别函数其结果是:

$$D \begin{cases} > 0 (< 0) & \text{质心位于 I 区} \\ = 0 & \text{质心位于直线 } H \text{ 上} \\ < 0 (> 0) & \text{质心位于 II 区} \end{cases} \quad (1)$$

当目标跨线时其判别函数 D 的正负属性必然变化,从而获知行人是否跨线.

其次,判断行人的过往方向,在没有任何先验知识时,质心点位于 I 区(或 II 区)时其判别函数 D 的正负号无法断定,因此设置一个已知位置属性的参考点 $R(r_x, r_y)$ 参与运算,如图 4 所示将 R 点设置在 I 区,于是与 R 的判别函数值具有共同正负号的质心点位于 I 区,反之位于 II 区.最终通过判断行人是否跨线及跨线前后质心点与参考点位置属性的对比情况实现对过往行人的带方向计数.

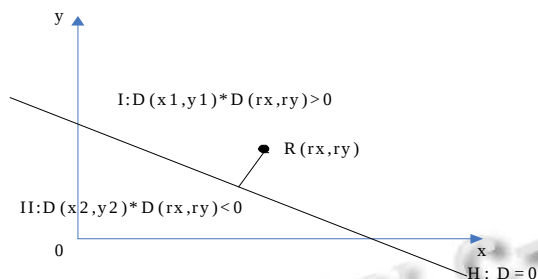


图 4 线性判别函数

2 系统设计与实现

本系统流程如图 5 所示,其中关键步骤设计与实现如下.

2.1 OpenCV 中 HOG 算法时效分析及行人检测加速策略

OpenCV 是 Intel 建立的高效的计算机视觉函数库,其中包含了 HOG 算法的实现.通过上述 HOG 的算法原理可见其对一副图像的扫描处理要经历不同缩放级别的子图像、子图像中的滑动窗口、滑动窗口中的滑动 Block、滑动 Block 中的 Cell、Cell 中的各个像素点

等多个层次.为了避免重复计算带来的时间消耗,OpenCV 采用了一种 HOG 缓存思想来加速算法运算,一方面是一次性计算图像各个像素点的梯度幅值和量化角度并存入堆内存中形成梯度幅值图像和量化角度图像以备后续计算中查找使用,另一方面将计算过的 Block 信息存入到内存查找表中,当窗口滑动到下一个位置时,重叠部分的 Block 信息可以直接从内存中读取而无需再次计算.尽管如此,该过程的时间消耗依然很庞大,表 1 统计了各部分的时间消耗.

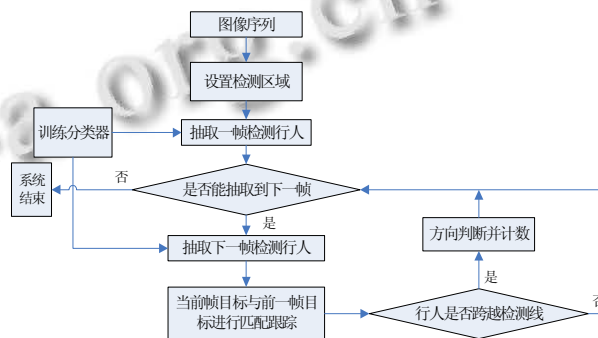


图 5 系统流程图

表 1 HOG 各环节时间消耗

功能	耗 时 (ms)
原图像中各像素点的梯度幅值和量化角度计算	25.4
原图像和所有 22 个子图像的梯度幅值和量化角度计算	297.8
第一个搜索窗的 HOG 特征提取及 SVM 判决	0.63
具有重叠 Block 的搜索窗 HOG 特征提取及 SVM 判决	0.05
原图像中所有 1551 个搜索窗的运算	88.7
原图像和 22 个子图像中的所有 14274 个搜索窗的运算	868.9
窗口融合	0.15
整幅图像行人检测	120.3

注:该统计结果是对一幅 352×288 的 RGB 三通道彩色图像测试所得

由表 1 可见,虽然一个检测窗的运算时间很短,然而累计时间却非常长.而且每级子图像的梯度幅值图像和量化角度图像的计算耗时也比较严重,其中大部分时间消耗对最终的计数结果没有影响.本文在借

鉴了 OpenCV 的 HOG 缓存思想的基础上将 HOG 算法的运算过程集中在最有效的 5 幅子图像中进行,从而避免了在无效子图像中的梯度幅值和量化角度计算上的耗时,并且因子图像的减少而显著降低了搜索窗的运算数量.

2.2 SVM 分类器的训练

本文使用 SVM 训练 48×96 的搜索窗. 先用样本集训练一个初级分类器对实验场景进行实验, 对误报为正样本的区域及其周边一定范围的区域进行样本提取后将其作为负样本添加到原有负样本集中进行再训练, 同样对于漏报的正样本也要将其添加到原有正样本集中再训练, 通过周期性反复训练, 获得一个适合当前应用场景的最佳分类器. 本文最终选定正样本为包含 MIT 行人数据库在内的 1255 张行人图像, 负样本 18018 张分别来自遍历楼道产生的负样本、行人肢体负样本和 INRIA 公共训练负样本.

3 实验结果及分析



场景 1 场景 2 场景 3

图 6 场景效果

搭建平台测试算法结果, 摄像机吊顶安装, 俯拍走廊环境, 见图 1. 以采集的楼道环境视频为测试样本, 视频图像大小为 352×288 , 帧率为每秒 15 帧, 具有光照变化、光照不均等特点. 在 3.2 GHz CPU 和 2GB 内存的 i3 PC 上进行实验, 软件平台为 WindowsXP+VS2008+OpenCV2.3.1.

图 6 是测试用的 3 个不同场景的效果图, 系统每帧处理速度约为 57ms, 可满足帧率为 15f/s 的系统的实时性需求, 相比直接使用 OpenCV 函数构建的每帧耗时 1 千多毫秒的系统, 时间效率上有了明显提高, 主要是运算量由穷举 14274(表 1 所示)个搜索窗减少为有针对性的 667 个搜索窗所致.

表 2 是三个不同场景的计数统计, 基于原 HOG 组成的系统由于实时性差, 无法有效实施跟踪从而导致系统计数失败的结果. 基于本文 HOG 组成的

计数系统影响计数精度的是漏计数, 漏掉的原因有: 1)行人的奔跑造成跟踪丢失导致漏计数; 2)行人遮挡导致对行人的检测失败造成漏计数. 虽然三个实验场景都面临行人的过往造成的光线变化, 但 HOG 描述子计算过程中的 Block 归一化降低了局部光线变化的干扰, 使得一般光线变化不直接影响计数精度.

表 2 计数结果

场景描述		场景 1: 光照变化, 行人较稀疏, 无簇拥人群出现。	场景 2: 光照变化, 光照不均, 行人稀疏, 无簇拥人群出现。	场景 3: 光照变化, 光照不均, 有簇拥人群出现。
实 际 人 数	由上至下	16	14	19
	由下至上	20	10	6
原 HOG 系 统 计 数	由上至下	0	0	2
	由下至上	0	0	1
	计数精度	0%	0%	12%
本 文 HOG 系 统 计 数	由上至下	16	13	13
	由下至上	18	10	5
	计数精度	94.4 %	95.8%	72%

4 结 语

本文基于 HOG 静态图像目标检测算法搭建行人计数系统, 针对 HOG 算法耗时较大问题, 测试了 OpenCV 实现过程中各环节的时间占有情况. 根据搜索窗运算总耗时比较严重而单个搜索窗运算效率又无法进一步提高的情况, 结合空间受限环境行人计数的需求, 限制了 HOG 的图像缩放范围并设置了检测带, 既减少了搜索窗的数量, 同时也降低了梯度幅值和量化角度的计算量. 在本文的测试环境中将原本需要 1203ms 完成的行人检测过程在 57ms 内完成, 时间效率明显提升. 计数方面提出用单直线划分平面空间, 线性判决函数判决质心位置属性的双向计数方法. 通过三个不同环境的实验证明本文方法具有不易受光线变化干扰且不依赖帧间运行信息的特点, 对空间受限情景具有快速稳定的计数效果.

参考文献

- 1 Rossi M, Bozzoli A. Tracking and counting moving people. IEEE International Conference Image Processing. 1994. 212–216.
- 2 Barandiaran J, Murguia B, Boto F. Real-time people counting using multiple lines. 9th International Workshop on Image Analysis for Multimedia Interactive Services. 2008. 159–162.
- 3 Albiol A, Mora I, Naranjo V. Real-time high density people counter using morphological tools. IEEE Trans. on Intelligent Transportation Systems, 2001, 2(4): 204–218.
- 4 Liu X, Tu P H, Rittscher J, Perera A, Krahnstoever N. Detecting and counting people in surveillance applications. IEEE Conference on Advanced Video and Signal Based Surveillance. 2005. 306–311.
- 5 Zeng C, Ma H. Robust head-shoulder detection by PCA-based multilevel HOG-LBP detector for people counting. IEEE Conference on Pattern Recognition. 2010. 2069–2072.
- 6 Sidla O, Lypetskyy Y, Brandle N, et al. Pedestrian detection and tracking for counting applications in crowded situations. IEEE International Conference on Video and Signal Based Surveillance. 2006. 70–70.
- 7 Dalal N. Finding People in Images and Videos. Institut National Polytechnique de Grenoble-INPG, 2006.
- 8 Zhu Q, Yeh MC, Cheng KT, Avidan S. Fast human detection using a cascade of histograms of oriented gradients. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2006. 1491–1498.
- 9 李同治, 丁晓青, 王生进. 利用级联 SVM 的人体检测方法. 中国图象图形学报, 2008, 13(3): 566–570.
- 10 Wojek C, Dorkó G, Schulz A, Schiele B. Sliding-windows for rapid object class localization: A parallel technique. Pattern Recognition, 2008: 71–81.