

基于改进特征跟踪的客流量统计

金 鑫^a, 梁雪春^a, 宫宁生^b

(南京工业大学 a. 自动化与电气工程学院; b. 电子与信息工程学院, 南京 211816)

摘 要: 为实现多目标的精确跟踪, 提出一种基于视频客流量的跟踪计数方法。利用图像差分法提取运动目标, 对差分图像进行形态学处理, 使用连通域分析创建人体结点, 得到目标的形心, 采用向前优先搜索像素点的原则搜索下一帧图像的所有像素点, 通过搜索到的像素点确定目标的新中心位置, 根据目标中心与计数线的关系进行计数。实验结果表明, 该方法简单可行, 统计数据具有较高的正确率。

关键词: 视频监控; 客流量统计; 差分法; 形态学处理; 连通域分析; 多目标跟踪

Passenger Flow Statistics Based on Improved Feature Tracking

JIN Xin^a, LIANG Xue-chun^a, GONG Ning-sheng^b

(a. College of Automation and Electrical Engineering;

b. College of Electronics and Information Engineering, Nanjing University of Technology, Nanjing 211816, China)

【Abstract】 In order to solve the problem about multitarget tracking, this paper proposes a new video-based traffic tracking counting method. The frame difference method is used to extract the moving target and the image needs morphological processing. Connected domain analysis is applied to create the human node and get the node's centroid. According to the nodes' centroids which are created in the list, first search forward is used. All target pixels are searched in the next frame which are used to determine the new position of the object in the video. Experimental results show that the method is simple and feasible, and achieves statistical data with high accuracy.

【Key words】 video surveillance; passenger flow statistics; difference method; morphologic processing; connected domain analysis; multitarget tracking

DOI: 10.3969/j.issn.1000-3428.2012.15.049

1 概述

人体运动的视觉分析是计算机视觉领域的重要课题之一。视频运动跟踪统计算法包括基于视频运动对象区域灰度等特征的跟踪算法、基于轮廓特征的跟踪算法、基于运动估计的跟踪算法和基于视觉学习的跟踪算法等。其中, mean shift 算法^[1]、snake 算法^[2]和基于特征的跟踪是常用的跟踪算法。mean shift 算法是对目标的颜色进行建模, 具有良好的实时性和跟踪性能。但 mean shift 根据颜色进行跟踪匹配, 当跟踪物体颜色与背景颜色相差不大时, 会出现跟踪失败。snake 活动轮廓模型(Active Contour Model, ACM)算法是一种较好的基于轮廓跟踪算法, 但 snake 及其改进算法与初始轮廓给定的位置密切相关, 若给定的初始形状和位置不佳, 则不易收敛或形状演化不正确, 且要取得理想的轮廓效果, 活动轮廓算法演化时间较长, 不适合实时操作。

本文结合文献[3]采用基于特征的跟踪算法^[4-5], 为实现跟踪并不需要跟踪目标的每一点, 只需跟踪目标上的某一特征点就可以实现对整个目标的跟踪。这个特征点可以是目标的质心, 也可以是目标上的任意一点。但这种算法在处理多目标跟踪时经常会出现跟踪匹配错误, 为了解决多目标精确跟踪的问题^[6], 本文对其算法进行了改进, 提出了根据方向

向量向前优先搜索像素的方法用于搜集目标像素, 利用计算特征点方法求出图像对象的位置, 根据跟踪目标的方向向量变化寻找下一帧图像中跟踪目标的位置, 对人体运动目标跟踪计数和客流量统计精度进行分析和比较。

2 人体运动检测

2.1 人体运动区域提取

帧差法是最常用的运动区域检测^[7]方法之一, 其基本原理是将同一背景下不同时刻 2 幅图像做差分或相减运算, 从相减后的图像中, 发现运动物体的差异信息, 用以指导动态监测、运动目标监测和跟踪、图像背景消除及目标识别等工作。

设输入的图像序列为 $F=\{f_i(x,y), x \in N, y \in M, i=1,2,3,\dots\}$, 其中, (x,y) 为二维视场($N \times M$)上的一点; $f_i(x,y)$ 为第 i 帧图像中 (x,y) 点的灰度值; i 为图像序列帧号; $G_i(x,y)$ 是第 i 帧帧差图像中 (x,y) 的值; T 为阈值。经相邻帧图像实时相减得到差分图像序列其定义如下:

$$\begin{aligned} G &= \{G_i(x,y), x \in N, y \in M, i=1,2,3,\dots\} \\ G_i(x,y) &= 255, |f_i(x,y) - f_{i-1}(x,y)| \geq T \\ G_i(x,y) &= 0, |f_i(x,y) - f_{i-1}(x,y)| < T \end{aligned} \quad (1)$$

由于采样时间间隔极短, 相邻帧图像之间的照明条件基

基金项目: 江苏省博士后科研资助计划基金资助项目(1001027B); 江苏省高校自然科学基金资助项目(09KJB510002)

作者简介: 金 鑫(1989—), 男, 硕士研究生, 主研方向: 模式识别, 图像处理; 梁雪春, 教授、博士; 宫宁生, 教授

收稿日期: 2011-09-23 **修回日期:** 2011-11-21 **E-mail:** superjx@foxmail.com

本不变, 因此对连续相邻 2 帧图像或间隔帧图像进行相减, 能消除大面积的背景和一些固定的噪声点, 获得运动目标的边界信息。图像的帧差效果如图 1 所示。



(a) 视频图

(b) 帧差图

图 1 图像的帧差效果

2.2 形态学处理

为了能更好地进行目标定位, 对差分后得到的二值图像首先进行开运算。即先腐蚀将目标与杂点隔离, 消除细小目标、孤立的点和小区域, 然后通过膨胀使图像恢复原来的大小, 再进行闭运算, 即先膨胀, 将内部小间隙和空洞填满, 并将目标边界平滑, 然后通过腐蚀运算使图像恢复原来的大小。这样在不改变目标大小的情况下, 能够有效地提高目标检测效果。中值滤波器是一种非线性平滑滤波器, 在一定条件下可以克服线性滤波带来的图像细节模糊问题, 而且对滤除噪声干扰及图像扫描噪声非常有效。形态学处理结果如图 2 所示。



(a) 开运算

(b) 闭运算

(c) 中值滤波

图 2 形态学处理结果

2.2.1 图像膨胀腐蚀

图像腐蚀作用是消除物体边界点, 使边界向内部收缩的过程, 可以把小于结构元素的物体去除。这样选取不同大小的结构元素, 就可以去除不同大小的物体。

图像腐蚀算法步骤如下:

(1) 扫描原图(如图 3 所示), 找到第 1 个像素值为 1 的目标点。

(2) 将预先设定好形状以及原点位置的结构元素(结构元素如图 4 所示)的原点移到该点。

(3) 判断该结构元素所覆盖的像素值是否全部为 1:

1) 是, 则腐蚀后图像中的相同位置上的像素值为 1。

2) 不是, 则腐蚀后图像中的相同位置上的像素值为 0。

(4) 重复步骤(2)和步骤(3), 直到所有原图中像素处理完成, 腐蚀结果如图 5 所示。

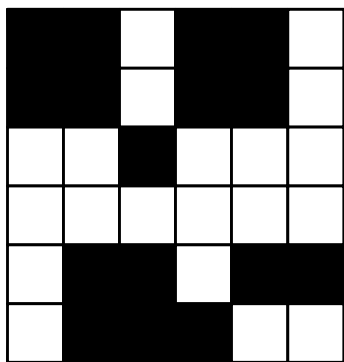


图 3 原图



图 4 结构元素

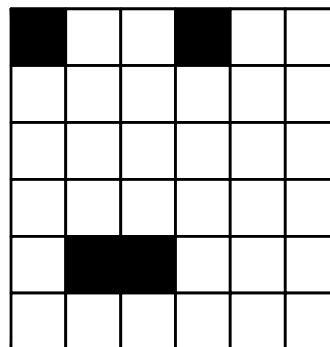


图 5 腐蚀结果

图像膨胀其作用正好和腐蚀相反, 它是对二值图像的边界点进行扩充, 将与物体接触的所有背景点合并到该物体中, 使边界向外部扩张的过程。如果 2 个物体之间的距离比较近, 则膨胀运算可能会把 2 个物体连通到一起, 膨胀对填补图像分割后物体中的空洞很有用。

图像膨胀算法步骤如下:

(1) 扫描原图, 找到第 1 个像素值为 0 的背景点。

(2) 将预先设定好形状以及原点位置的结构元素的原点移到该点。

(3) 判断该结构元素所覆盖的像素值是否存在为 1 的目标点:

1) 是, 则膨胀后图像中的相同位置上的像素值为 1。

2) 不是, 则膨胀后图像中的相同位置上的像素值为 0。

(4) 重复步骤(2)和步骤(3), 直到所有原图中像素处理完成, 膨胀结果如图 6 所示。

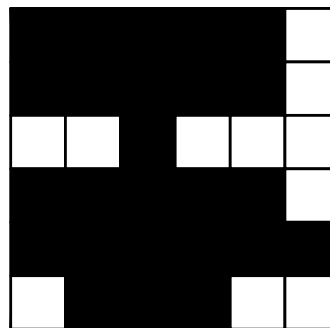


图 6 膨胀结果

2.2.2 中值滤波

中值滤波器的原理是: 采用一个含有奇数个点的滑动窗口, 用窗口中各点灰度值的中值代替中心点的灰度值。对于奇数个元素, 中值是指按大小排序后中间的数值; 对于偶数个元素, 中值是指排序后中间 2 个元素灰度值的平均值。

中值滤波算法步骤如下:

(1) 获取源图像的首地址及图像的宽和高。

(2) 开辟一块内存缓冲区, 用于存放结果图像, 并初始化为 0。

(3) 逐个扫描图像中的像素点, 将其领域各元素的像素值从大到小进行排序, 将求得的中值赋值给目标图像中与当前点对应的像素点。

循环步骤(3), 直到处理完源图像的全部像素点。

由于中值滤波器是直接取滤波模板内中间像素点的值作为当前像素点的灰度值, 因此在二值图像中只有 0 和 255

2 个值, 可以采用更简单的方法, 比如计算当前点周围的黑点的个数, 大于一定阈值当前点为黑, 否则当前点为白。本文采用了这个方法, 得到了一个清晰的滤波图像。

3 人体运动目标的跟踪计数

3.1 人体信息结点的创建

通过连通域分析^[8]收集人体所有帧差像素, 通过连通域的分析可以去掉干扰点, 并能有效分析出人的轮廓信息。其主要思想就是通过 9 宫格判断周围 8 个点是否为有效点(黑点), 若是就保存下来, 再判断下个点, 依次迭代下去直到一帧全部扫描结束。人体信息结点的创建算法实现如下:

Step1 用 10×10 的方格(根据视频悬挂高度决定)分割屏幕, 并用这个方格进行扫描屏幕, 这样做是为了减少系统开销, 如果在这个方格中出现有效黑点, 就将这个方格记录下来并将其放到一个数组中, 其实是将方格左上角的点记录下来, 放到数组中去。这样循环下去屏幕上所有格子有有效点的都将被装到数组里, 假设这个结构体数组为 $Point[i]$ 。

Step2 从这个数组 $Point$ 任取一个元素(也就是一个点), 将其放到另一个数组中去, 这个数组为 $zone[m][n]$, 二维数组 $zone$ 的目的是存放一个连通域的所有有效点(格子)。m 表示域的个数; n 表示每个域最多能存放有效点的个数。

如将刚才放到 $zone[i][0]$ 的点与 $Point$ 里的每个点作比较, 如果它们之间的距离小于 10 个像素点, 那么就认为它们之间连通。将中间的点与其周围的 8 个点作比较, 如果条件成立(它们之间的距离小于 10 个像素点), 那么就将 $Point$ 里的这些点放到 $zone[i][j]$ 里, 依次循环下去, 那么人的连通域就可以找到了。

Step3 排除干扰点, 有些干扰点就会被放到其他 $zone[i]$ 里, 最后判断 $zone[i]$ 里有效点的个数, 如果小于一定数值就默认他不是一个人连通域, 将其释放掉。最后留下来的就是一些人的连通域。

Step4 根据得到的连通域的像素点计算 $zone[i]$ 连通域的宽度。本文通过测量得到一个标准人的宽度为 W (约 30 个像素)。根据这个数据就可以将人进行分割。假设 d 表示连通域的宽度; n 表示人数; u 表示人宽度的最小值。

将人进行分割算法如下:

```
n=d/W
if(n%W ≥ u)
n=n+1
```

Step5 在分割之后, 所有连通域进行一次新的组合, 现在每一个 $zone[i]$ 代表一个人, 根据 $zone[i]$ 里的像素可以算出整个连通域的 $zone[i]$ 的中心值, 即人的形心位置 (x, y) 。将这些值都统一存放在 $person[i]$ 的结构数组里, 这个过程就成功创建了所有人的结点。

在最后 Step4 分割这一部分其实就解决了并排多人行走跟踪的问题。

3.2 向前优先目标匹配

根据文献[9-10], 为了实现多人准确跟踪, 本文采用了向前优先搜索帧差像素的原则。通过采用方向偏移标志 $Direction$ (1、2、3、4 分别代表上下左右 4 个方向), 确定物体的移动方向, 最后在移动的方向上找取人的下一帧像素点, 这样就能准确确定人的位置信息。方向偏移标志

$Direction$ 值的确定通过第 $n-2$ 帧图像中对象位置 (x_0, y_0) 与第 n 帧图像中相应对象位置 (x_1, y_1) 进行比较。

若 $|x_1 - x_0| - |y_1 - y_0| > 0, x_1 - x_0 > 0$, 则 $Direction = 4$ 。

若 $|x_1 - x_0| - |y_1 - y_0| > 0, x_1 - x_0 < 0$, 则 $Direction = 3$ 。

若 $|x_1 - x_0| - |y_1 - y_0| < 0, y_1 - y_0 > 0$, 则 $Direction = 2$ 。

若 $|x_1 - x_0| - |y_1 - y_0| < 0, y_1 - y_0 < 0$, 则 $Direction = 1$ 。

算法实现如下:

Step1 在已经创建结点的基础上, 根据标志 $Direction$ 的值, 采取向前优先的原则的跟踪。具体的实现是通过半径为 r (约 15 个像素)的圆依次向前移动, 且每次向前移动 $t(5)$ 个像素的距离, 共移动 $k(3)$ 次, 每次移动后将在圆范围内的有效像素点都读取出来。

Step2 从这些像素点中找到 x 的最大最小值 $minx$ 、 $maxx$ 和 y 的最大最小值 $miny$ 、 $maxy$ 。这样就可以根据 $maxy - miny$ 的值的大小来判断前后是几个人。在通过测试后发现一个人的前后宽度大约在 W 个像素左右。这样就可以通过前后的宽度来判断后面是否还有人。

Step3 如果 $maxy - miny > H$ (人的前后厚度, 约 40 像素), 那么假设后面还有一个人。先把一个人的前后宽度里面的点全部找出来, 根据这些点来获得新的结点圆心坐标。

Step4 那么多于一个人宽度的点还要释放出来, 这些点也许就是后面人的像素点。将释放出来的像素点置成未被读取状态, 以便后面的人创建信息。

在 Step3 就很好地解决了前后多人一起走的情况, 将前后靠在一起的人分割开来。

3.3 计数

当被跟踪物体是在创建区 $S1(S2)$ 创建时, 必须将这个结点 $person[i]$ 的 in_or_out 元素置 1(0)。当结点到达区域 $S2(S1)$ 时, 先判断是否是已有结点, 若是就再根据 in_or_out 的值判断是否计数。若 $person[i].y > \text{监控区域上(下)边界值}$, 而且 $person[i].in_or_out == 1(0)$, 则将 $person[i]$ 结点删除, $IN(OUT)$ 计数器加一, 其他出边界的情况则只删除结点不计数。

4 实验结果与分析

本文是在 Linux 环境下用标准 C 语言实现, 系统界面如图 7 所示。视频的分辨率为 288×352 像素, 帧率为 25 f/s。



图7 系统界面

4.1 客流量统计精度比较

通过视频播放, 分别使用 mean shift 跟踪算法和改进的基于特征的跟踪算法对某家商店从下午 1 点-5 点之间的客流量进行数据统计, 客流量统计结果如表 1 所示(M 表示 mean shift 算法; F 表示改进的特征跟踪算法)。可见, 改进的特征跟踪算法的精度要达到 95% 以上, 且精度要高于 mean shift 跟踪算法。

表 1 客流量统计结果

测试门	实际进店人数	系统统计的进店人数		进店统计准确率/(%)		实际出店人数	系统统计的出店人数		出店统计准确率/(%)	
		M	F	M	F		M	F	M	F
正门	1 829	1 646	1 802	90	99	1 828	1 626	1 789	89	98
收银门	1 058	973	1 029	92	97	1 044	1 200	1 089	87	96

改进算法优点在于解决了因客流拥挤进入视频监控区而造成的跟踪匹配错误,提高了跟踪和计数的精确度。但仍然存在误判,主要原因是门口光线与门内光线强度差造成对跟踪的影响,使得人体的跟踪像素点发生变化,导致跟踪匹配错误,其次就是人群中混有类似人大小(人的宽度、厚度)的移动物体,有时会出现误判,但一般的不同于标准人尺度大小的移动物体能够准确识别出来。

4.2 算法的实时性比较

改进的跟踪算法在跟踪时仅需要进行特征点计算,跟踪效果能满足实时要求。不同算法的平均跟踪时间比较如表 2 所示,比较结果以 10 s 视频设置 30 个关键帧图像匹配为例。

表 2 不同算法的平均跟踪时间比较

算法	平均运行时间/s
mean shift 算法	0.3
snake 算法	0.5
改进的特征跟踪算法	0.1

实时跟踪效果如图 8 与图 9 所示,左半边是视频监控图,右半边是视频处理后的跟踪图,图中的黑圆圈代表被跟踪的人。

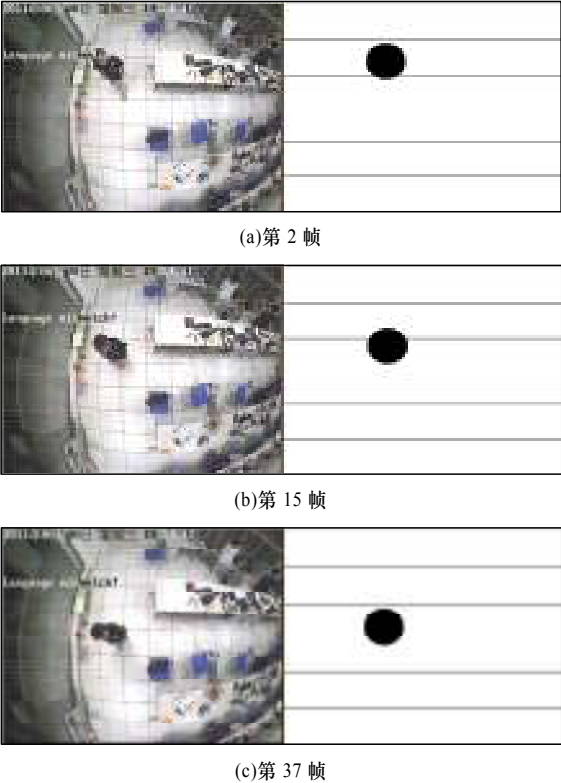


图 8 单人跟踪效果示意图

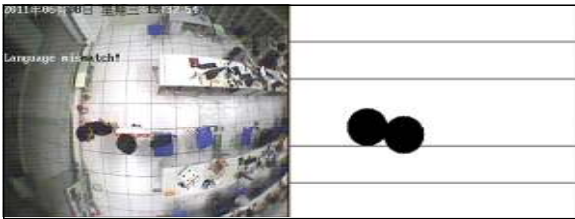


图 9 多人跟踪效果图

5 结束语

本文对人体视频跟踪进行研究,提出一种改进的特征跟踪算法。引入方向向量,用其预测下一帧人体位置,并通过准确的人体分割获得多人跟踪结果。实验结果表明,该方法的识别准确率在 95%以上,具有较好的实用价值。今后将研究如何跟踪混合在人群中、与人体相似的移动物体。

参考文献

[1] 袁 霄,王丽萍. 基于 MeanShift 算法的运动人体跟踪[J]. 计算机工程与科学, 2008, 30(4): 46-49.

[2] 刘皓挺,姜国华,王 丽. 一种基于 Snake 模型的多目标跟踪算法[J]. 计算机工程与应用, 2006, 23(12): 76-79.

[3] 陈 明,陈一民,黄诗华. 增强现实中视频对象跟踪算法[J]. 计算机工程, 2010, 36(12): 229-231.

[4] Hager G D, Dewan M, Stewart C V. Multiple Kernel Tracking with SSD[C]//Proc. of IEEE Conference on Computer Vision and Pattern Recognition. [S. l.]: IEEE Press, 2004.

[5] Dai Yaping, Yu Guanghui, Hirasawa K. New Development on Tracking Algorithm with Derivation Measurement[EB/OL]. (2010-11-21). <http://ieeexplore.ieee.org/iel5/7658/20951/00972008.pdf>.

[6] 魏娟丽,翟社平,王万诚. 视频序列中人体运动目标的检测与跟踪研究[J]. 计算机应用与软件, 2006, 23(4): 139-141.

[7] 李国辉,焦 波,涂 丹. 视频序列中二值图像的快速目标检测方法[J]. 计算机工程, 2009, 35(18): 217-218, 226.

[8] 赵宏伟,冯 嘉. 一种实用的运动目标检测和跟踪算法[J]. 吉林大学学报: 工学版, 2009, 39(2): 386-390.

[9] Yilmaz A, Javed O, Shah M. Object Tracking: A Survey[J]. ACM Computing Surveys, 2006, 38(4): 2678-2684.

[10] Green R D, Guan Ling. Quantifying and Recognizing Human Movement Patterns from Monocular Video Images——Part I: A New Framework for Modeling Human Motion[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2004, 14(2): 179-190.

编辑 刘 冰