

基于空间梯度直方图的行人检测算法

孙 昀 刘富强 李志鹏

(同济大学嵌入式系统与服务计算教育部重点实验室/宽带无线通信与多媒体实验室,上海 201804)

摘要 为了快速准确地对行人进行检测,提出了一种采用空间梯度直方图作为特征的行人检测方法。该方法首先使用直方图相似性和 Fisher 准则来衡量所有定义特征的分辨能力,然后选择一些具有强分辨能力的特征来表征行人,使该方法获得了较高的检测率;最后用 SVM 分类器用来训练从目标和背景中获得的经过筛选的特征。实验结果表明,该方法对行人的检测是快速且有效的。

关键词 空间梯度直方图 行人检测 SVM 特征选择

中图分类号:TP391.41 文献标识码:A 文章编号:1006-8961(2008)10-1825-04

Pedestrian Detection Using Spatial Histograms of Oriented Gradients

SUN Yun, LIU Fu-qiang, LI Zhi-peng

(The Key Laboratory of Embedded System and Service Computing Supported by Ministry of Education/Broadband Wireless Communication and Multimedia Laboratory, Tongji University, Shanghai 201804)

Abstract This paper presents a pedestrian detection approach using spatial histograms of oriented gradient feature. In order to detect pedestrians faster and more accurately, histogram similarity and Fisher criterion are employed to measure discriminability of all features and then we selected some discriminative features to identify the pedestrian. SVM classifier is constructed to train the selected features from the target and surrounding background. Experimental results show that the proposed approach is efficient and rapid in pedestrian detection.

Keywords spatial histograms of oriented gradients, pedestrian detection, SVM, feature selection

1 引言

在计算机视觉领域里,行人检测一直是一个富有挑战性的课题。因为行人除了有不同的衣着和姿势以外,对行人进行检测的主要困难还在于处理的速度和算法的鲁棒性。目前已有许多方法用于行人检测,如 Leibe 等人使用基于“局部特性的编码”来进行行人检测^[1]。Oliver 等人利用边缘图像来对不同的形状模型进行匹配(ASM)^[2]。Viola 等人利用 AdaBoost 算法训练检测器,同时使用行人的动作和外貌信息进行检测^[3]。Mikolajczyk 等人综合了位置方向直方图和二进制

梯度,对人体的各个局部进行匹配^[4],其中包括了行人的正面和侧面。

最近 Dala 和 Triggs 提出了一种基于梯度直方图(Hog)特征的行人检测算法^[5],并取得了良好的效果。Zhu 和 Avudan 改进了这个方法^[6],即使用 Adaboost 算法去选择冗余的 Hog 特征,从而加速了整个处理过程,并取得了较好的检测结果。但在层叠式训练器训练的过程中,由于该算法在每层中仅随机地选取了 5% 的总特征作为输入特征,因而会导致所选取的特征并非是全局最优的。如果初始的 5% 的特征中没有“较好”的特征,那么系统的检测结果不可能达到准确。

本文将特征选择作为一个全局问题,采用

Fisher 准则作为评估标准,并用支持向量机(SVM)作为分类器,以便使行人检测具有良好的实时性和较高的准确率。

2 空间梯度直方图

目标表征和特征选择一直是目标检测中最精髓的问题。对于行人检测而言,由于肢体的外形变化和自由移动,其中 Hog^[5] 特征被证明是一组较好的表征形状和轮廓特性的特征,所以本文用扩展的 Hog 特征作为行人的模型,同时通过定量分析这些特征来选择一些最“具可分性”的特征。

2.1 梯度直方图的计算

本文沿用文献[5]中提取梯度直方图的基本思路,首先将一个“块”(block)分成 4 个“片”(cell),然后分别计算“片”中像素的梯度,并将其分为 9 个方向,同时统计每“片”的梯度直方图,最后每个“块”可得到 4 × 9 维的特征向量。值得注意的是,这里采用 0° ~ 180° 作为梯度的范围,这将使特征的性能达到较高的水平。

“积分图”能快速计算类 Harr 小波的矩形特征。与之相对应,本文采用“积分直方图”来快速计算梯度直方图,即首先计算每个像素的梯度值以及梯度方向(本文中有 9 个方向);然后将计算出来的梯度值在各个方向上计算积分图;最后计算每个梯度积分图中“片”所在区域的特征值,以便组成 Hog 特征直方图。在对“块”进行归一化的问题上,为了适应“积分直方图”的特征,并加快计算速度,本文采用文献[5]中所提到的 L1 归一化方法。

2.2 空间梯度直方图

在文献[5]的方法中,对于 Hog 特征的定义只有简单的一个尺寸(16 × 16 pixels 大小的 105 个“块”),而这种 Hog 特征仅仅能够表征目标某些局部的梯度特征,对于更大范围的梯度特征,乃至全局的梯度特征就无能为力了。如果要用不同尺寸的“块和片”作为特征,那么整个算法的计算量将要大增加。受 Boosting 算法的启发,本文采用特征选择的方法来解决这个问题,即对大量的特征进行选择时,由于只是从中挑选某些最具代表性的特征作为目标的特征进行训练和识别,从而大大降低了训练和识别的时间。

根据 64 × 128 pixels 大小的检测窗口,本文使用的“块”的大小从 (16 × 16) pixels 到 (64 × 128)

pixels,长宽比例分别取 1:1、1:2 和 2:1。另外,根据“块”的大小选用 {4,6,8} 中的一个作为平移量,这样就得到了一组交叠稠密的“块”,称之为空间梯度直方图,标记为 $Hog(x,y,w,h)$ 。这里 x 和 y 代表“块”的起点(左上点)坐标, w 和 h 代表“块”的宽和高。在 (64 × 128) pixels 大小的窗口中总共定义了 3 766 个“块”,每个“块”都将得到一个 4 × 9 维的特征向量。使用这样一组多尺度的“块”有以下两个好处:(1)对一类特殊的物体种类而言,它的特征涵盖了一个较大的范围,而原始的 105 个固定大小的“块”则仅仅描述了少量的特征信息;(2)在 3 766 个“块”中的某些可能对应于检测物体的语意上的部分,例如,行人检测中的头、肩或腿。而固定大小的“块”则难以建立这样的对应。

3 特征选择

在进行特征选择时,首先,将正样本的所有特征 $SAM^{Hog(x,y,w,h)}(P_j)$ 取均值作为样本的模板 $MOD^{Hog(x,y,w,h)}$ 。若有 n 个目标样本,则

$$MOD^{Hog(x,y,w,h)} = \frac{1}{n} \sum_{j=1}^n SAM^{Hog(x,y,w,h)}(P_j) \quad (1)$$

然后,衡量每个样本(包括正负全体样本)的特征直方图和模板直方图之间的相似性。参考文献[7],本文采用相交直方图方法来计算两个直方图的相似性,相交的结果就代表了样本与模板之间具有相同方向的梯度值大小,即

$$D(H_1, H_2) = \frac{\sum_{i=1}^K \min(H_1^{(i)}, H_2^{(i)})}{\sum_{i=1}^K H_1^{(i)}} \quad (2)$$

这里 H_1 和 H_2 代表了两个直方图, K 为直方图的级数。

对于任意一个样本的特征,其与模板之间的相似性为

$$z_j = D(SAM^{Hog(x,y,w,h)}(P_j), MOD^{Hog(x,y,w,h)}) \quad (3)$$

而所有特征的相对于模板的相似性为

$$F^{Hog(x,y,w,h)} = \{z_1, z_2, \dots, z_m\} \quad (4)$$

对于任何一个 Hog 特征,从理论上说虽然其有区分目标物体的能力,但是为了能够更容易地区分目标,通常希望选择一些具有较强区分能力的特征,用来表征目标物体。因此,本文使用 Fisher 准则来衡量每个 Hog 特征的区分能力。对每个 $Hog(x,y,$

w, h) 都能得到一组样本与模板的相似度集合 $F^{Hog}(x, y, w, h)$ 。对于样本库, 用 ω_1 代表正样本子集, ω_2 代表负样本子集, S_b 代表类间离散度矩阵, 其可由以下公式得到:

$$S_b = (m_1 - m_2)^2 \quad (5)$$

这里 $m_i = \frac{1}{N_i} \sum_{z \in \omega_i} z_j, j \in \omega_i, i \in \{1, 2\}$ 。

S_1, S_2 分别代表正负两类类内离散度矩阵, 可由以下公式计算得到:

$$S_i = \frac{1}{N_i} \sum_{z \in \omega_i} (z_j - m_i)^2 \quad j \in \omega_i, i \in \{1, 2\} \quad (6)$$

类内离散度矩阵 S_w 为正负两类离散度矩阵之和, 即

$$S_w = S_1 + S_2 \quad (7)$$

其 Fisher 准则就是类间离散度矩阵 S_b 与类内离散度矩阵 S_w 的比值, 即

$$J(f_j) = \frac{S_b}{S_w} \quad (8)$$

Fisher 准则越大, 特征的区分能力就越强。

一组有效的特征不仅需要具有强的区分能力, 同时特征与特征之间也要相互独立。根据空间 Hog 特征的特点, 本文采用两个“块”交叠的面积来衡量两个特征之间的空间相关性(如图 1 所示)。当交叠面积与较大“块”面积之比小于阈值 T 时, 则认为两个特征相互独立。

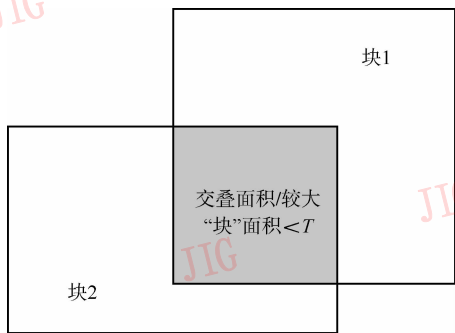


图 1 两个相互独立的特征
Fig. 1 Two independent features

SVM(支持向量机)作为良好的模式识别算法, 其是利用了数据样本中两类之间最大的距离作为划分的超平面。本文用线性的 SVM($C = 0.1$) 进行训练。这里, SVM 首先被用来判断所选的特征是否足够, 然后用所选择的特征训练 SVM 并将训练结果作为分类器。

采用迭代的特征选择算法获得一组特征子集

F_{select} , 具体的算法如下:

(1) 根据最大 Fisher 准则找到特征 f_{max} , 则 $F_{select} = \{f_{max}\}$, $F_{ori} = F \setminus \{f_{max}\}$, F_{ori} 为所有剩下特征的集合。

(2) 在 F_{ori} 中, 根据最大 Fisher 准则找到 \hat{f} , 并测试 \hat{f} 与 F_{select} 中所有的特征的相关性, 如果它们相互独立, 则 $F_{select} = F_{select} \cup \{\hat{f}\}$, $F_{ori} = F_{ori} \setminus \{\hat{f}\}$

(3) 训练分类器 C , 同时用测试样本进行评估, 并记录当前正确率(并可同时记录误检率和漏检率)

$$R_{accuracy}^{cur} = 1 - \frac{1}{k} \sum_{i=1}^k |C(p_i) - y_i|$$

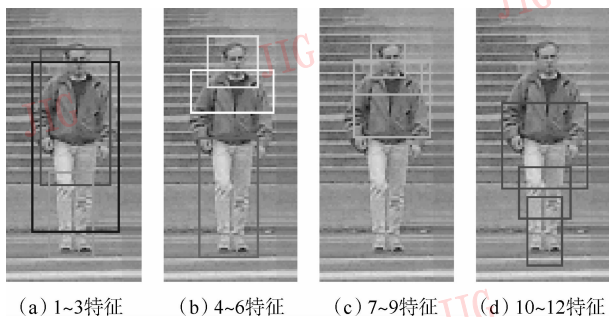
这里 $C(p_i)$ 是分类器的输出, $C(p_i) \in \{0, 1\}$ 。 p_i 为样本, y_i 表示对应的正负样本集(其中 1 为正样本, 0 为负样本)。

(4) 如果 $R_{accuracy}^{cur} - R_{accuracy}^{pre} \leq \varepsilon$ (ε 是一个很小的常量, $R_{accuracy}^{cur}$ 表示当前循环的正确率, $R_{accuracy}^{pre}$ 表示上次循环的正确率), 则退出算法, $F_{selected}$ 为选择的特征。否则, $R_{accuracy}^{pre} = R_{accuracy}^{cur}$, 转到步骤(2)继续循环。

4 实验结果及算法分析

本文的主要贡献就是对空间梯度直方图特征进行了有效的选择, 下面来观察一下所选特征的性质。

由于本文选择的特征具有最大的可区分性, 因此图 2(a) 所示的 3 个“最好”的“块”, 可以表征人体的大致轮廓(如: 身体、腿、头)。图 2(b) ~ 图 2(d) 显示了随后选择的“块”位置。与文献[2]相比, 本文选择的“块”则更具有“语义”。在文献[2]方法中, 仅仅只有一些“块”表征了人的身体、腿和头等重要的部位, 因为其在每个循环中只抽取了 5% 的特征进行选择, 这在一定程度上影响了检测结果。如果对所有的特征用 Adaboost-SVM 算法进行



(a) 1~3特征 (b) 4~6特征 (c) 7~9特征 (d) 10~12特征

图 2 所选特征的位置

Fig. 2 Position of the selected feature

训练,则又极其耗费训练时间,而本文的算法能有效地平衡这对矛盾。

为了更好地了解算法的性能,还和文献[5]、[6]算法进行了比较,本文沿用了文献[5]中衡量系统性能的 DET(检测错误率折中)曲线来进行对比。图3画出了这3种算法的漏检率/误检率曲线(其中两条引自相应文献)。该曲线通过改变 ε 的值,并在测试集上通过统计漏检率和误检率得到。从图中可以看到,本文算法在误检率为 10^{-4} 时,可获得较高的检测率,大约为90%,该精度是令人比较满意的。

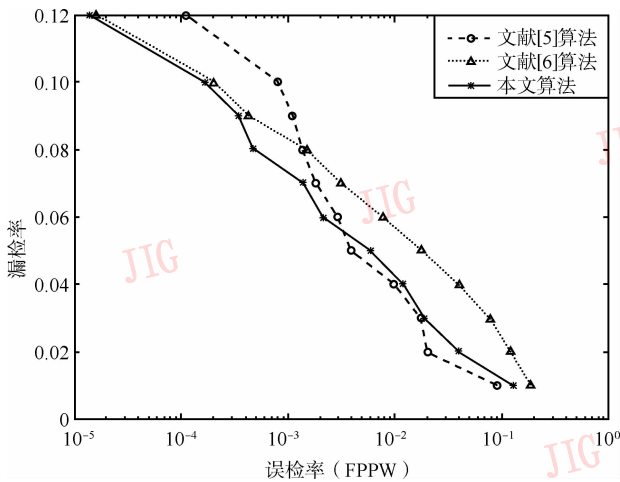


图3 3种算法的漏检率/误检率比较曲线

Fig.3 Curve of three algorithm miss rate/false rate

在常用的 PC 机上实现了本文算法。其运行环境为 Pentium IV 3.0 GHz 的 CPU 和 1 GMB 的内存。表1是3种算法对 240×320 大小的图片进行处理的时间比较。由表1可以看到,本文算法基本满足了实时性的要求,虽然略慢于文献[6]算法,但远远快于文献[5]的算法,约为其运算时间的1/5。

表1 对 240×320 大小的图片进行处理的时间比较

Tab.1 Time required to evaluate 240×320 image

	粗扫描		细扫描	
	时间(ms)	窗口数	时间(ms)	窗口数
文献[5]算法	1 000	4 000	未提及	
文献[6]算法	26	800	106	12 800
本文算法	76	1260	194	8 640

5 结 论

本文提出了一种基于空间梯度直方图的行人检测算法。该方法不仅能够自动选择一些具有“高区分度”的特征,而且通过理论分析和实验测试证明,本文算法具有较快的速度和较好的检测率(如图4所示)。下一步研究方向应该考虑如何解决拥挤情况下的行人检测。



图4 检测结果

Fig.4 Detection result

参考文献 (References)

- 1 Leibe B, Seemann E, Sziele B. Pedestrian detection in crowded scenes[A]. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition [C], San Diego, CA, USA, 2005, 1: 878 ~ 885.
- 2 Oliver Sidla, Yuriy Lypetsky. Pedestrian detection and tracking for counting applications in crowded situations[A]. In: Proceedings of IEEE Conference on Video and Signal Based Surveillance [C], Washington, DC, USA, 2006, 1: 70 ~ 76.
- 3 Viola P, Jones M J, Snow D. Detecting pedestrians using patterns of motion and appearance[A]. In: Proceedings of IEEE International Conference on Computer Vision [C], Nice, France, 2003, 1: 734 ~ 741.
- 4 Mikolajczyk K, Schmid C, Zisserman A. Human detection based on a probabilistic assembly of robust part detectors[A]. In: Proceedings of European Conference on Computer Vision [C], Prague, Czech Republic, 2004, 1: 69 ~ 82.
- 5 Dalal N, Triggs B. Histograms of oriented graients for human detection[A]. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition [C], San Diego, CA, USA, 2005, 1: 886 ~ 893.
- 6 Zhu Qiang, Avudan Shai. Fast human detection using a cascade of histograms of oriented gradients[A]. In: Proceedings of Conference on Computer Vision and Pattern Recognition [C], New York, USA, 2006, 2: 1491 ~ 1498.
- 7 Swain M, Ballard D. Color indexing[J]. International Journal of Computer Vision, 1991, 7(1): 11 ~ 32.