

基于 PERCLOS 判据的驾驶员疲劳监测系统^①

金 梅^② 薛静芳^③ 张立国 刘 强

(燕山大学电气工程学院 秦皇岛 066000)

摘 要 针对疲劳驾驶监测常用算法运算效率低的问题,提出了一种基于单位时间内人眼闭合时间所占比例(PERCLOS)判据的驾驶员疲劳监测方案。针对传统顺序执行算法运算效率低的问题,提出了基于流水线算法的人脸检测及人眼跟踪算法。针对传统顺序式算法对硬件要求高的问题,本系统运行在低端现场可编程门阵列(FPGA)上,设备成本大幅降低。并利用 PERCLOS 判据作为判断驾驶员是否疲劳的依据。实验表明,本文所提出的基于流水线算法的图像处理算法使系统拥有了较高的实时性和较低的资源占用,实现了 60 fps 的处理速度,系统的准确度达到了 95% 以上。

关键词 疲劳监测;流水线算法;人脸检测;人眼跟踪;单位时间内人眼闭合时间所占的比例(PERCLOS)判据

0 引 言

随着社会与科技的发展,汽车越来越普及,成为了社会生产力发展的强劲动力,为人们出行提供了便捷。但随着汽车普及,安全问题也随之而来。据国家统计局统计,2019 年,全国共发生 247 646 起交通事故,导致 62 763 人死亡,造成直接财产损失 134 618 万元,而由疲劳驾驶引发的事故则占 20% ~ 40%。因此,针对驾驶员疲劳度检测系统的研究能产生巨大的经济效益和社会效益。但用于疲劳驾驶检测的传统顺序式算法对硬件性能要求高,导致设备成本高昂。如文献[1]提出一种基于 MTCNN-PFLD-LSTM 深度学习模型的疲劳驾驶检测算法。文献[2]采用计算机视觉的方法利用 Dlib 提取面部特征点的坐标,通过计算眼睛纵横比来实现瞌睡检测。文献[3]提出通过从用户端摄像头提取面部特征,计算人眼睛的开合程度、嘴部开合度、头部偏移程度判断疲劳的融合算法。上述文献及文献[4-8]

所述的方法均采用顺序式算法,对处理器的性能及工作频率要求高,且若要达到较高的处理帧率,需要采用较高端的图形处理器(graphic processing unit, GPU)或者中央处理器(central processing unit, CPU),导致成本高昂。

现场可编程门阵列(field programmable gate array, FPGA)得益于其低成本及并发计算的特点,在信号处理和图像处理等领域发挥了重要的作用。本文通过 FPGA 开发板、摄像头搭建驾驶员疲劳监测系统,将传统顺序式算法改进为流水线算法,多个处理步骤并行执行,充分利用 FPGA 并发特点实现实时人脸图像处理及眼部特征提取,并利用单位时间内人眼闭合时间所占的比例(percentage of eyelid closure over the pupil over time, PERCLOS)判据判断驾驶员疲劳状态。FPGA 并发计算的特性及本文对算法的流水线改进,使得系统延时大幅降低,处理速度大幅提高。并且本系统可运行在低端 FPGA 上而准确度和性能几乎不受影响,从而大幅降低设备成本。

^① 河北省中央引导地方科技发展专项(199477141G)资助项目。

^② 女,1977 年生,博士,副教授;研究方向:机器人控制及应用技术,机器视觉,智能信息处理;E-mail: meijin297@126.com。

^③ 通信作者,E-mail: 1939984670@qq.com。

(收稿日期:2021-11-05)

1 算法的流水线改进

1.1 基于 PERCLOS 判据的疲劳监测方案

本文基于 PERCLOS 判据的疲劳监测流程图如

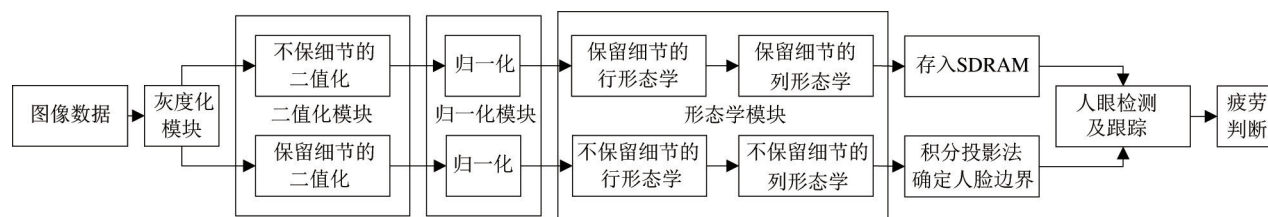


图 1 基于 PERCLOS 判据的疲劳监测流程图

用于后续人脸检测。经二值化模块后,输出的数据长度变为 8 位。完成形态学处理后,保留细节的图像存入同步动态随机存储器 (synchronous dynamic random-access memory, SDRAM),与此同时,不保留细节的图像进入人脸定位模块,在存储完成时,人脸定位也完成。紧接着读取 SDRAM 中的图像,在人脸范围寻找眼睛,并运用 PERCLOS 判据判断驾驶员是否疲劳。

1.2 流水线算法

流水线算法将一个算法拆分成多级算法步骤,每级占用一定的时钟周期。每级流水线并行运行,但由于后一级的输入是前一级的输出,所需的处理时间为第一个数据进入第一级流水线、到第一个处理结果输出所需的时钟周期。

FPGA 使用流水线算法能获得非常快的处理速度,且所需处理的数据越多,算法步骤越多,流水线路的优势也越大。

1.3 灰度化流水线

摄像头输出 16 位 RGB565 格式数据,因此先将 RGB565 格式数据通过高位补低位的方式转换为 RGB888 格式^[9],再利用 RGB888 格式数据计算灰度值。RGB 转灰度常用公式为

$$Gray = (R \times 299 + G \times 587 + B \times 114) / 1000 \quad (1)$$

为了进一步提高算法的处理速度,将式(1)算法中的除法改进为移位算法。移位比除法快,所以可以将系数缩放成 2 的整数幂。

图 1 所示。

摄像头输出的数据为 16 位 RGB565 格式,数据同时流经 2 个二值化模块,通过设定不同的阈值,配合形态学模块,一幅图像保留脸部细节,用于后续人眼检测及跟踪。另一个输出抹去脸部细节的图像,

所以 RGB 转灰度公式可分解为三级流水线,即提取并扩展颜色分量、计算括号内乘法、计算加法并移位除法。每级流水线消耗一个时钟周期。

1.4 形态学操作的流水线

流水线算法只能处理一维数据,因此,腐蚀/膨胀操作需分解为行方向 (1 × 3) 与列方向 (3 × 1) 2 个一维操作。

1.4.1 一维行形态学操作

摄像头输出的数据经二值化后,数据依然是连续地从二值化模块输出,而进行 1 × 3 的行腐蚀须同时获取 3 个像素数据。由于同一时刻只能收到 1 个数据,并且需要保证每个流经形态学操作模块的数据一个不漏全部被处理,因此不能用暂存数据再处理的方法。在此提出让数据流经一个 3 bit 移位寄存器,并且每个时钟周期将寄存器后两位数据前移一位,将新读取的二值化数据放入寄存器末位,以实现数据在寄存器内流动。从二值数据进入一维形态学模块以后的第 2 个时钟周期即可开始判断腐蚀/膨胀结果,输出结果落后判断一个时钟周期,即第 3 个时钟周期开始输出第 1 个结果,意味着此种方法对一幅图像进行行一维腐蚀/膨胀操作只相当于把图像数据延时了 2 个时钟周期,但经过这 2 个时钟周期的延时后,输出的图像已经是经过行形态学操作的图像^[10]。

1.4.2 一维列形态学操作

在行方向进行形态学的基础上对列方向进行形态学操作,原理和行形态学类似。但由于列形态学

需要同时获取位于相邻 3 行的像素,因此至少需要缓存 2 行加第 3 行的第 1 个像素,即需要一个 1281 位的寄存器。与行形态学类似,列形态学也是当像素处于中间位置时判断该像素,即第 641 个数据,意味着此模块只需要 784 个时钟周期(一行图像持续 784 个时钟周期,其中 640 个位像素时钟周期)便可开始输出结果,即完成一幅图像的列形态学操作相当于把图像数据延时了 784 个时钟周期。

通过结合行与列方向 2 个一维形态学操作,实

现了对图像的二维形态学操作。

1.5 人脸检测

由于摄像头输出的数据会同时进行 2 个阈值不同的二值化及形态学操作,分别输出保留脸部细节和不保留脸部细节的图像。为了节省资源,提高效率,本文优化了算法流程,实现了实时判断、投影结束立即得出人脸边界,并且只占用少量寄存器资源。人脸定位模块状态转移图如图 2 所示。

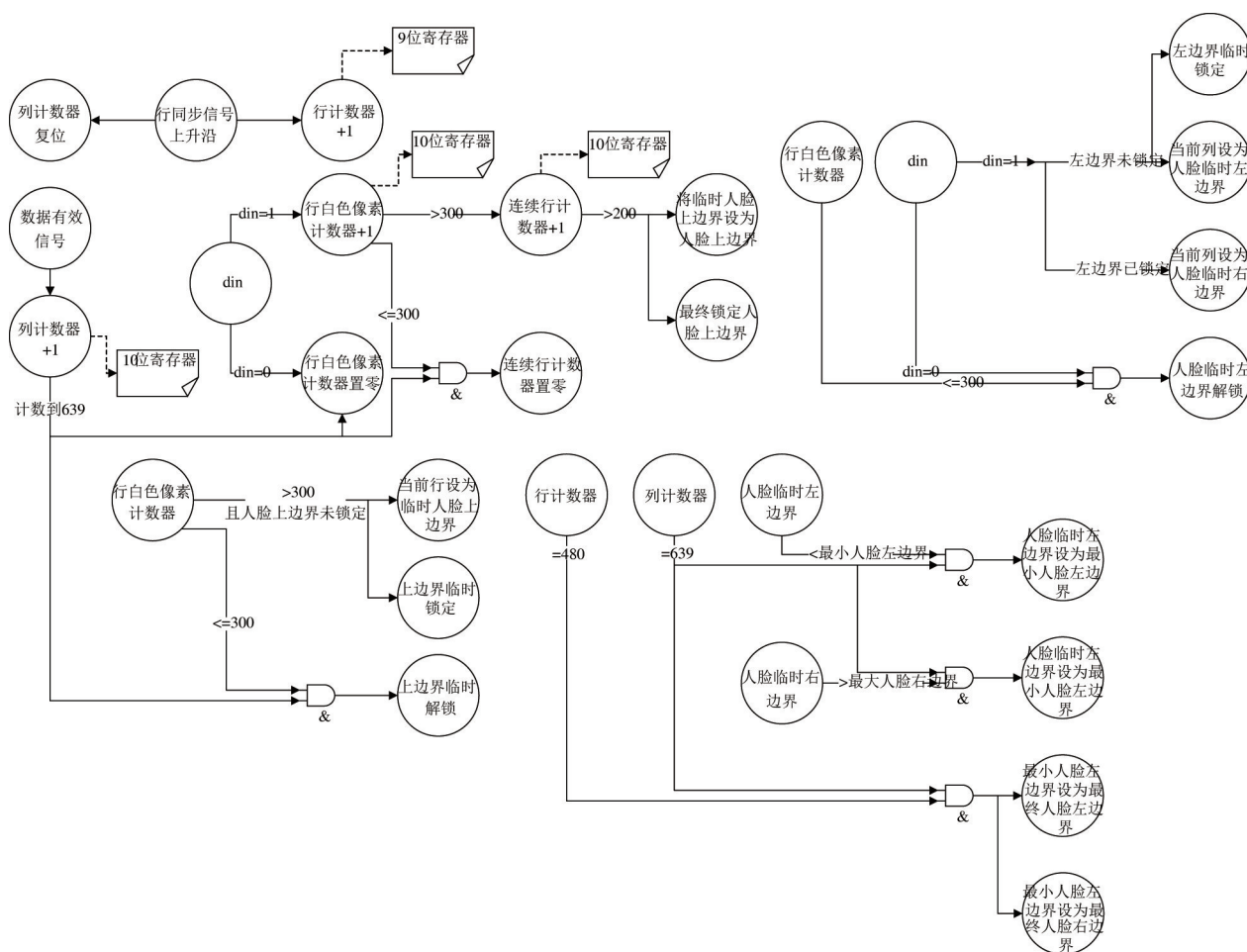


图2 人脸检测模块状态转移图

1.6 人眼定位

与人脸定位类似,人眼定位也利用投影法,但在投影之前需要确定眼睛的大致区域。根据“三庭五眼”比例关系^[11],以及人脸的最宽处一般位于眼睛与鼻子之间的区域这一规律,只需要找到人脸最宽处,就可以在人脸上下边界与最宽处之间利用投影法寻找眼睛,也能较好地避免“三庭”规律的缺陷。区

域分割及人眼定位结果如图 3 所示。

1.7 人眼跟踪

本文系统采用的摄像头输出帧率为 60 fps, 即相邻两帧间隔约 0.016 s, 而人眼在这段时间内的位移属于微位移。因此检测到眼睛后, 后续处理将在上一帧眼睛边界的基础上扩展边界大小, 并将扩展的边界作为下一帧检测眼睛的边界, 如此反复迭代,

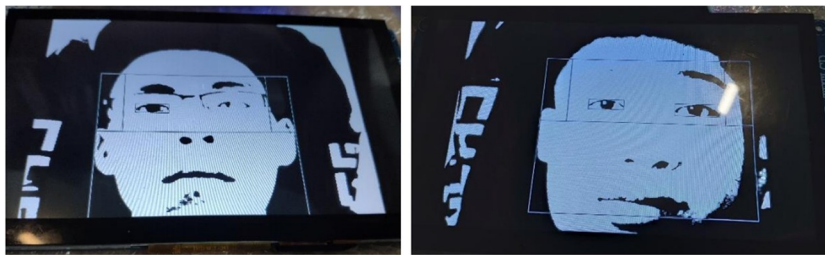


图3 人眼定位

实现跟踪。

1.8 疲劳判定

PERCLOS 判据^[12]的 P80 标准定义眼睛从最大闭合到闭合 80% 大小所用时间为 t_1 , 从闭合 80% 到 20% 所用时间为 t_2 , 从闭合到 20% 到睁开到 20% 所用时间为 t_3 , 从睁开 20% 到睁开 80% 所用时间为 t_4 。则 PERCLOS 指数 f 定义如下。

$$f = \frac{t_3 - t_2}{t_4 - t_1} \times 100\%$$

当 f 超过一定范围则认为驾驶员疲劳。虽然此方法理论上能达到非常高的准确度,但在实际操作时并不常用,主要有以下 2 个原因。

(1) 因为当驾驶员距离摄像头较远时,眼睛的最大变化量不够大,导致无法区分这 4 个阶段。

(2) 此法对系统速度要求极高,人眨眼一次只需要 0.2 ~ 0.4 s,以本文为例,处理速度为 60 fps,则每次眨眼只持续了 10 帧左右,闭眼过程和睁眼过程平均只持续了 5 帧左右,用如此少的帧数去描述闭眼与睁眼过程,精度较低。

因此,在实际应用时,常用一段时间内闭眼的时间占统计时间的百分比判断驾驶员的疲劳程度。即

$$f = \frac{t_c}{T} \times 100\%$$

其中 t_c 为闭眼持续的帧数, T 为总统计时间所持续的帧数。睁闭眼状态由眼睛长宽比判断。

2 实验结果

2.1 PERCLOS 值标定

根据实际测试效果,确定当眼睛的长宽比大于 4 时,认为眼睛处于闭合状态,并在此基础上进行 PERCLOS 值标定。

对 10 名测试者每人随机记录 5 个周期的闭眼

帧数数据,实验统计数据如表 1 所示。

表1 清醒时闭眼帧数结果

总帧数	清醒时闭眼帧数 平均值	疲劳时闭眼帧数 平均值	f
5400	754	1270	0.2

根据表 1 所测结果可知,在 5400 帧里,清醒时眼睛闭合帧数在 1200 帧以下,疲劳时在 1200 帧以上,即得出的对应关系如下。(1) $f < 0.2$, 驾驶员清醒;(2) $f > 0.2$, 驾驶员疲劳;(3) 驾驶员长时间闭眼报警时间设为 2 s,持续闭眼超过 2 s 则认为驾驶员极度疲劳。

2.2 疲劳检测测试

测试时设定的采样周期为 30 s,摄像头采集 900 帧图像,系统读取 1800 次,按照标定结果,疲劳时 $f > 0.2$,即闭眼帧数计数器计数值应大于 360。

根据表 2 所示测试结果,可知本系统的准确度约为 96% (大于 360 的次数除以 50)。

表2 疲劳时测试结果

闭眼帧数 测试者	采样组别				
	1	2	3	4	5
1	458	499	369	452	463
2	563	515	544	537	521
3	274	481	534	387	558
4	466	470	342	442	428
5	484	491	397	474	489
6	548	593	403	576	456
7	582	379	431	466	480
8	452	420	471	455	434
9	402	392	449	373	486
10	493	496	532	285	472

2.3 实验对比

实验使用 verilog 语言编写疲劳监测系统的逻辑电路,并在 AltraEPCE0F7C FPGA 芯片上部署,电路驱动时钟频率为 50 MHz,摄像头捕获画面的分辨率为 640 × 480 像素。

以文献[13]采用的关键点特征融合的方式作为对比组,与本文所叙述的 PERCLOS 疲劳计量方法进行对比。实验前预先统计测试视频序列中出现的疲劳次数,与实际检测出的疲劳次数对比,算法对比的实验结果如表 3 所示。

表 3 检测算法对比实验结果

算法	漏检次数	误检次数	正确检测率
PERCLOS	14	9	96.3%
面部关键点特征融合	10	21	90.5%

实验表明,PERCLOS 疲劳计量方法采用单一因素进行检测,故误检发生不频繁且正确率较高。面部关键点融合特征算法能够检测出较多的疲劳情况,但由于该算法涉及多个疲劳因素与判定方式,在定值判定的情况下容易发生误检测的情况。

将文献[14]提出的基于转向盘转角的疲劳驾驶检测方法,与本文提出的基于 PERCLOS 判据的驾驶员疲劳监测方法进行对比,利用实际道路条件下的车辆行驶数据对所建立的模型进行验证。实验对比结果见表 4。

表 4 判别模型实验结果对比

模型	误检次数	正确检测率	帧率/fps
基于 PERCLOS 判据的疲劳监测	11	95.6%	60
基于转向盘转角的疲劳驾驶监测	25	80.3%	25

实验表明,本文的驾驶员疲劳检测方案不仅在正确率上较基于转向盘转角的疲劳驾驶检测方案高,在速度上也更快。基于转向盘转角的疲劳检测方法导致误判的原因主要是,在少数路况简单的直线路段无论是在清醒状态还是疲劳状态,短时间内转向盘转角变化的幅度和频率都较小,使得提取的

静止百分比和角度标准差失效,从而导致误判的发生。

3 结 论

针对常规疲劳监测算法处理速度慢、实时性低、硬件成本高等问题,本文提出了一种基于 PERCLOS 判据的驾驶员疲劳监测系统。通过对常规算法进行流水线改造,保证了系统的高速处理,达到了实时性的要求;并且通过改进算法,使得本系统所占用的资源处于较低水平,这也意味着本系统能运行在低端 FPGA 上,从而节省硬件成本。在人脸定位方案中实现了以较低的资源占用完成人脸定位,且实现了每帧图像结束时即可得出人脸定位结果。此外,在人脸定位的基础上利用人脸比例关系,将眼睛区域缩小到人脸上边界与人脸最宽处之间的部分,从而缩小了检测人眼的范围,简化了程序,并利用系统处理速度快这一优势,使用迭代法实现了人眼跟踪。最后进行实验测试以及对比实验,测试结果表明本系统在保证实时性的情况下达到了 95% 以上的准确度。

参考文献

[1] 李小平,白超. 基于深度学习的司机疲劳驾驶检测方法研究[J]. 铁道学报, 2021, 4(6):78-87

[2] 孙玥,杨国为,陈雪鑫. 基于计算机视觉的疲劳及注意力检测算法[J]. 计算机与数字工程, 2021, 49(6):1195-1198, 1239

[3] 冯宇航,叶辉,黄文禧,等. 基于头部多姿态融合识别的疲劳检测与预警研究[J]. 电脑知识与技术, 2021, 17(13):12-14

[4] 盛振涛,李梦珂. 基于深度学习和微表情检测的防疲劳驾驶检测综述[J]. 信息与电脑(理论版), 2019(7):126-127

[5] 刘明周,蒋倩男,扈静. 基于面部几何特征及手部运动特征的驾驶员疲劳检测[J]. 机械工程学报, 2019, 55(2):18-26

[6] 孔研,常若松,刘天娇,等. 驾驶员主观疲劳状态与方向盘转角对事故倾向性的综合预测模型[J]. 人类工效学, 2018, 24(2):36-42

[7] 唐阳山,徐忠帅,杨语尧. 基于面部表情的驾驶员疲

- 劳特征提取[J]. 辽宁工业大学学报(自然科学版), 2018, 38(6):404-407, 412
- [8] 陈昕, 朱国华, 张驰, 等. 基于眼睛闭合状态的驾驶疲劳检测方法[J]. 辽宁工业大学学报(自然科学版), 2018, 38(3):182-186, 191
- [9] HE H, ZHANG X, JIANG F, et al. A real-time driver fatigue detection method based on two-stage convolutional neural network[J]. *IFAC-Papers OnLine*, 2020, 53(2): 15374-15379
- [10] CHAUDHURI A, ROUTRAY A. Driver fatigue detection through chaotic entropy analysis of cortical sources obtained from scalp EEG signals[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2019, 21(1):1-14
- [11] AWASEKAR P, RAVI M, DOKE S, et al. Driver fatigue detection and alert system using non-intrusive eye and yawn detection[J]. *International Journal of Computer Applications*, 2018, 180(44):1-5
- [12] ZHAO X P, MENG C N, FENG M K, et al. Fatigue detection based on cascade convolutional neural network[J]. *Journal of Optoelectronics · Laser*, 2017, 28(5): 497-502
- [13] 詹润哲, 姜飞. 基于面部关键点特征融合的驾驶员疲劳检测方法研究[J]. 宿州学院学报, 2018, 33(10): 115-117
- [14] 刘军, 王利明, 聂斐, 等. 基于转向盘转角的疲劳驾驶检测方法研究[J]. 汽车技术, 2016(4):42-45

Driver fatigue monitoring system based on PERCLOS criterion

JIN Mei, XUE Jingfang, ZHANG Ligu, LIU Qiang

(School of Electrical Engineering, Yanshan University, Qinhuangdao 066000)

Abstract

Aiming at the low efficiency of commonly used algorithms for fatigue monitoring, this paper proposes a driver fatigue monitoring program based on the percentage of eyelid closure over the pupil over time (PERCLOS) criterion. Aiming at the low efficiency of traditional sequential execution algorithms, a face detection and eye tracking algorithm based on pipeline algorithm is proposed. Aiming at the problem of high hardware requirements for traditional sequential algorithms, this system runs on low-end field programmable gate array (FPGA), and the equipment cost is greatly reduced. The PERCLOS criterion is used as the basis for judging whether the driver is fatigued. Experimental results show that the image processing algorithm based on the pipeline algorithm proposed in this paper enables the system to have higher real-time performance and lower resource consumption. A processing speed of 60 fps has been achieved, and the test results show that the accuracy of the system has reached more than 95%.

Key words: fatigue monitoring, pipeline algorithm, face detection, eyetracking, percentage of eyelid closure over the pupil over time (PERCLOS) criterion