

基于FPGA的多运动目标检测算法实现

朱鹏程*, 王夫康, 曹 允, 窦 亮, 王 岩

(中国电子科技集团公司第五十五研究所, 南京 210016)

摘 要: 针对目标检测对实时性要求越来越高的情况, 提出了一种基于现场可编程门阵列(Field Programmable Gate Array, FPGA)实现的多运动目标检测算法。该方法通过帧差法提取运动目标, 然后基于距离阈值对形态学处理后的图像进行目标分割, 最后对运动目标标记显示。系统通过CMOS摄像头采集视频, 采用FPGA平台外接DDR3高速大容量缓存, 实现了视频图像的采集、存储、目标检测和显示。实验结果表明, 提出的多目标检测系统能够有效的实时检测出多个运动目标, 并且在分辨率为1 024×600的情况下帧率达到38 fps。

关键词: 现场可编程门阵列; 运动目标检测; 帧差法; 阈值

中图分类号: TN79+.1 **文献标志码:** A **文章编号:** 1005-488X(2023)01-0074-06

Implementation of Multi-moving Target Detection Algorithm Based on FPGA

ZHU Pengcheng, WANG Fukang, CAO Yun, DOU Liang, WANG Yan

(The 55th Institute of China Electronics Technology Group Corporation, Nanjing 210016, CHN)

Abstract: In view of the increasingly high real-time requirement of target detection, a multi-target detection algorithm based on field programmable gate array was proposed. Firstly, the moving targets were extracted through the frame difference method, and then the target segmentation of images after morphological processing was performed based on the distance threshold. Finally, the moving targets were marked and displayed. The system was based on CMOS camera, FPGA, external DDR3 high-speed large-capacity cache, and realized the functions of video image acquisition, storage, target detection and display. The experimental results showed that the target detection system could effectively detect multiple targets in real time, and could reach 38 fps at a resolution of 1 024×600.

Key words: FPGA; moving target detection; frame difference method; threshold

引 言

运动目标检测和跟踪一直是计算机视觉领域

中的研究重点, 传统的基于PC软件执行的算法日趋完善, 随着视频图像越来越清晰, 需要处理的数据量越来越大, 目标检测系统的灵活性和实时性越

收稿日期: 2022-11-30

作者简介: 朱鹏程(1999—), 男, 硕士研究生, 主要从事FPGA图像处理相关方向的研究; (E-mail: 1031286842@qq.com)

王夫康(1991—), 男, 硕士, 工程师, 主要从事显示、光电探测技术研究;

曹 允(1975—), 男, 研究员级高工, 长期从事显示电路技术的研究。

* 通讯作者

来越差,无法适用在一些需要小体积、快响应的场合。为此可以采用嵌入式系统实现图像处理算法,利用FPGA(Field Programmable Gate Array)强大的逻辑处理能力和并行计算能力,可实现高帧率高清视频的目标检测与跟踪。

目前,运动目标检测算法主要有帧间差分法、背景差分法、光流法、卡尔曼滤波法。光流法通过分析光流场变化来检测运动目标,算法复杂度高,且容易受光照强度变化、噪声和背景扰动等因素的干扰^[1-3];卡尔曼滤波法通过上一帧的观测值预测下一帧运动物体的位置,算法复杂度高,且涉及除法运算,对毫无规律的运动进行预测时精度差强人意^[4];背景差分法利用当前图像帧和背景帧作差,与阈值进行比较分割目标^[5-6],虽然能检测出运动目标,但如何构造背景帧是一个难点,而且不同场景下构造背景帧的方法可能不再适用;帧间差分法直接对相邻的两帧或多帧图像进行差分,通过选取合适的阈值对差分后的图像二值化,从而检测出运动物体^[7],运算简单易于FPGA实现。

基于以上分析,充分利用FPGA强大的并行处理能力,同时结合帧间差分法,设计了一种多目标检测算法,实现了对多个运动目标的实时检测。整个系统分为图像采集模块和目标检测模块,图像采集模块负责实时视频图像的采集和格式转换,目标检测模块对采集到的视频进行帧间差分算法,并对差分后的图像进行二值化处理和形态学处理,凸显运动目标,最后对检测出的多个运动目标进行加框显示。以下从多运动目标检测算法,FPGA实现,实验结果和分析等几个方面进行介绍。

1 多运动目标检测算法

整个系统实现运动目标的检测标记功能,需要对摄像头采集到的图像进行一系列的处理,包括灰度化算法、差分二值化、形态学处理、基于距离阈值的目标分割算法、标识算法等。

1.1 图像灰度化算法

整个系统的数据输出设置为24位RGB888格式,因此将摄像头的输出数据配置成16位RGB565格式,然后拼接成RGB888格式,拼接方法为{R[4:0],R[2:0],G[5:0],G[1:0],B[4:0],B[2:0]},其中R、G、B分别为RGB565格式中的红、绿、蓝分量。为得到更易处理的灰度图像,可以将RGB格式转换为

YCbCr格式,其中Y分量即为灰度分量。一种通用的转换公式如下:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.172 & -0.339 & 0.511 \\ 0.511 & -0.428 & -0.083 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

为转化为FPGA易处理的整数运算,使用扩大256倍再右移8 bit的方式,转换方式如下:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \left\{ \begin{bmatrix} 77 & 150 & 29 \\ -43 & -85 & 128 \\ 128 & -107 & -21 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 32768 \\ 32768 \end{bmatrix} \right\} \gg 8$$

1.2 差分二值化算法

文章采用的帧间差分算法对相邻两帧进行作差,假定 $f(x,y,k)$ 和 $f(x,y,k-1)$ 表示视频图像序列第 k 帧和第 $k-1$ 帧中的某一像素点 (x,y) 处的灰度值,利用下式进行判断:

$$g(x,y,k) = \begin{cases} 1, & |f(x,y,k) - f(x,y,k-1)| > \Delta \\ 0, & \text{otherwise} \end{cases}$$

式中, Δ 为预设的阈值,取值为0~255中的任意整数,用来将差分后的图像二值化,阈值设置与实际应用场景有关,阈值的选取会影响到实际检测结果,方便起见,这里设置 Δ 为固定值。 $g(x,y,k)$ 为二值化之后像素点 (x,y) 的布尔值,当相邻两帧同一位置的灰度值相差大于阈值时,判定处在该像素点位置的物体是运动的,否则判定物体是静止的。两帧的帧差法实现简单,易于硬件实现,加之两帧图像相差时间短,算法实时性较好,而且对环境光强变化不敏感,抗干扰能力强。

1.3 形态学处理

形态学处理主要是针对二值化图像,通过特定大小的窗口在整张图像滑动,利用邻域运算后的结果代替原来的布尔值。形态学处理的基本运算主要包括四种:膨胀、腐蚀、开运算和闭运算^[8],在图像中如果存在一些不必要的结构,进行形态学处理,并不会影响基本形状,下面简要介绍一下这几种形态学运算。

(1) 膨胀

图像膨胀的作用是扩张图像的边界点,使高亮部分区域更大,便于提取整体目标。作为 Z^2 中的集合A和B,用模板B对图像A的膨胀定义为:

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\}$$

(2) 腐蚀

腐蚀的作用是消除图像的边界点,减小高亮区

域,使图像沿着边界向内收缩。用模板 B 对图像 A 的腐蚀定义为:

$$A \ominus B = \{z | (B)_z \in A\}$$

(3) 开运算

开运算操作即对图像先腐蚀后膨胀,开运算能够去除孤立点噪声,平滑较大物体的边界,用模板 B 对图像 A 的开运算定义为:

$$A \circ B = (A \ominus B) \oplus B$$

(4) 闭运算

闭运算操作即对图像先膨胀后腐蚀,闭运算能够填充前景图像中的空洞,弥合小裂缝,用模板 B 对图像 A 的闭运算定义为:

$$A \bullet B = (A \oplus B) \ominus B$$

帧差法检测运动物体时会产生很多空洞和细小的椒盐噪声,为此可采用开运算对差分判断后得到的二值化图像进行处理,减少噪声和断裂点对检测结果的干扰。为方便 FPGA 处理和减少资源消耗,采取一个 3×3 的模板对二值图像进行处理,则腐蚀操作就是 3×3 模板中的 9 个像素点的布尔值相与,膨胀操作就是对 3×3 模板中的 9 个像素点的布尔值相或。

1.4 基于距离阈值的运动目标分割算法

在差分运算之后,判定差分绝对值大于阈值的像素点为运动目标,针对帧差后图像存在分散空洞的现象,提出了一种基于间距的多目标标记算法,考虑两个被认定为运动目标的像素点,记 A 的像素坐标为 (x_A, y_A) , B 的像素坐标为 (x_B, y_B) ,设置距离阈值为 T,若满足 $|x_A - x_B| \leq T$ 且 $|y_A - y_B| \leq T$,则认定两个像素点同属一个运动目标,否则认定不属于同一个运动目标。设置最多能同时检测到 8 个不同的运动目标,若实际运动目标个数大于 8 个,则选取标记算法中按照自然数顺序排列的前 8 个目标进行加框标记,否则按照顺序对符合要求的运动目标进行标记。

在帧消隐期,检查各个标记框边界信息,若标记框存在交叉或者包含的情况,则将这种情况下的标记框进行合并,并将相互交叉或包含的目标记为同一个目标,经过整理后得到各个运动目标最小外接矩形的边界坐标。

1.5 标识算法

在经历消隐期的标记框整理之后,在下一帧到来前得到了上一帧运动目标最小外接矩形的边界

信息,由于相邻两帧相差时间很短,因此可以用上一帧的检测结果代替当前帧,在当前帧进行显示时,在上一帧多个运动目标外接矩形的位置加上红色标记框。

2 算法 FPGA 实现

本系统基于 FPGA 设计了一个实时多运动目标检测系统,整个系统主要由摄像头配置模块、算法处理模块、视频显示模块 3 部分组成,总体设计框图如图 1 所示。

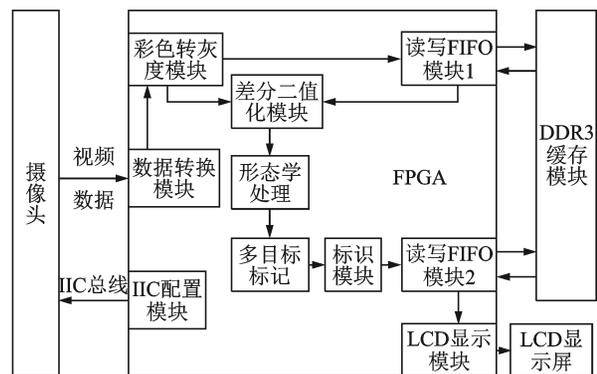


图 1 多运动目标检测系统设计框图

Fig.1 The framework of multi-moving target detection system

2.1 摄像头配置模块

文章采用 CMOS 摄像头采集视频数据,其感光阵列达到 2592×1944 (即 500 万像素),使用的是两线式 SCCB 接口总线,对寄存器数据进行配置,包括:数据输出格式、白平衡、增益、曝光、分辨率等,配置数据在 SCL 的上升沿发送给 SDA,使摄像头工作在预期的工作模式,文章设置摄像头输出 1024×600 像素的视频,摄像头时钟为 84 MHz,视频帧率为 38 fps,输出为 RGB565 格式,每个像素点的数据通过两个时钟输出。

2.2 算法处理模块

2.2.1 差分二值化模块

在图像灰度化处理之后,需要存储像素数据进行差分运算,由于存储一帧位宽为 16 bit、像素为 1024×600 的图像需要资源较多,研究通过外挂 DDR3 芯片进行存储。DDR3 共开辟出 4 块存储区,其中两块存储区使用乒乓操作存储灰度化之后的视频流,另外两块存储区同样使用乒乓操作存储加框标记之后的 RGB 格式视频流。如图 2 所示,通过

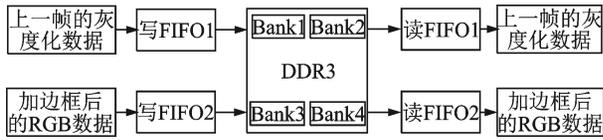


图2 DDR3的4端口控制模块

Fig.2 4 port control module of DDR3

一个DDR3的4端口控制模块和4个FIFO即可控制两个视频流的读写,读FIFO1输出上一帧的灰度化数据与当前帧的灰度化数据相差一个时钟,通过延时即可将两帧灰度数据同步,进行差分和二值化计算;读FIFO2输出当前帧加框标记后的RGB格式视频流给显示模块;DDR3通过存储器接口生成器(Memory Interface Generator, MIG)IP核进行控制。

2.2.2 形态学处理模块

FPGA实现膨胀腐蚀处理时,使用 3×3 的模板窗口,因此需要同时取相邻3行的数据。文中通过两个Shirt_Ram IP核实现,在得到模板内的9个数据后,先进行与操作,后进行或操作,得到开运算结果。在实现算法设计时,为提高运算速度,采取两级流水的方式来实现。以腐蚀操作为例,第一级流水先将 3×3 模板中每一行的3个二值数进行“与”运算;第二级将第一级“与”后的3个结果再进行“与”运算,两个像素时钟便可得到最终结果。

2.2.3 多目标标记模块

记已记录的目标数为 cnt ,第 $i(1 \leq i \leq 7)$ 个目标的边界坐标信息为 $(x_{imin}, y_{imin}), (x_{imax}, y_{imax})$,其中 x_{imin} 、 y_{imin} 、 x_{imax} 和 y_{imax} 分别为第 i 个目标外接矩形的最小横坐标、最小纵坐标、最大横坐标、最大纵坐标,通过这几个参数即可确定第 i 个目标最小外接矩形的位置。算法流程如下:

1) 在每一帧有效像素来临前,给8个运动目标的边界信息置零, $cnt=0$ 。

2) 实时计算每个目标外接矩形的邻域:

$$target_bottom[i] = \begin{cases} y_{imax} + T, y_{imax} \leq 599 - T \\ 599, y_{imax} > 599 - T \end{cases}$$

$$target_right[i] = \begin{cases} x_{imax} + T, x_{imax} \leq 1023 - T \\ 1023, x_{imax} > 1023 - T \end{cases}$$

$$target_top[i] = \begin{cases} y_{imin} - T, y_{imin} \geq T \\ 0, y_{imin} < T \end{cases}$$

$$target_left[i] = \begin{cases} x_{imin} - T, x_{imin} \geq T \\ 0, x_{imin} < T \end{cases}$$

其中, $target_left[i]$ 、 $target_top[i]$ 、 $target_right[i]$

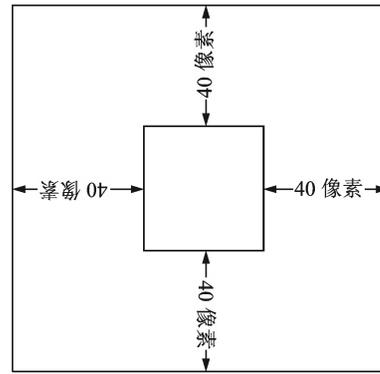


图3 目标邻域

Fig.3 Neighborhood of target

和 $target_bottom[i]$ 分别为第 i 个目标外接矩形域的最小横坐标、最小纵坐标、最大横坐标、最大纵坐标。

3) 在视频流到来时根据像素点的坐标 (x, y) 和8个目标的邻域信息进行投票,若当前像素点的坐标处在目标的邻域内,则该目标投票值为0,否则投票值为1。

4) 在下一个时钟,若当前像素点不处在任何一个目标的矩形邻域内,判断此时 cnt 的大小:若 $cnt < 8$,则令 $cnt = cnt + 1$,同时将当前像素点的坐标设置为第 $cnt + 1$ 个目标的边界,即:

$$x_{(cnt+1)min} = x, x_{(cnt+1)max} = x;$$

$$y_{(cnt+1)min} = y, y_{(cnt+1)max} = y。$$

若 $cnt \geq 8$,则不进行任何操作。若当前像素点在记录的目标矩形邻域内,则根据当前像素点坐标对第 j 个记录的目标边界按下式进行更新(满足前 $j-1$ 个目标对当前像素点投票值为1,第 j 个目标对当前像素点投票值为0):

$$x_{jmin} = \min(x_{jmin}, x);$$

$$x_{jmax} = \max(x_{jmax}, x);$$

$$y_{jmin} = \min(y_{jmin}, y);$$

$$y_{jmax} = \max(y_{jmax}, y);$$

记两个目标为A和B,其中 $x_{Amin} \leq x_{Bmin}$,当满足 $x_{Bmin} \leq x_{Amax}$ 且 $y_{Bmin} \leq y_{Amax}$ 时,称两个目标是交叉关系,这时给两个运动目标加框会出现交叉和包含的情况。针对这种情况,文章设计了一个边框合并算法,如图4所示,整个算法分为三个阶段,在帧消隐期来临时开始对8个矩形框进行合并:

在第一阶段,需要向FIFO中写入两种交叉关系,首先写入新的目标标记值,即将 $\{0, cnt\}(1 \leq cnt \leq 8)$ 写入FIFO;写完后将所有交叉关系对 $\{la, lb\}$ 写入FIFO,其中 $la \geq lb$,同时以标记值为地址将8个目标

- | | | |
|-----------------|---|------------------------------|
| 1. 写入各个边框的交叉关系。 | 2. 对所有交叉关系进行初步整理, 将目标标记值都替换成具有交叉关系的最小值。 | 3. 根据初整理关系更新外接矩形边界值, 得到最终参数。 |
|-----------------|---|------------------------------|

图 4 算法的三个阶段

Fig.4 Three stages of algorithm

的边界信息写入坐标 RAM 中。

在第二阶段, 从 FIFO 中读出交叉关系对 $\{la, lb\}$, 通过状态机找到和 la 有交叉关系的最小值 $temp1$, 以及 lb 对应的最小值 $temp2$, 若 $temp1 = temp2$, 则以 la 为地址将 $temp1$ 写入交叉关系 RAM 中; 若 $temp1 < temp2$, 则以 $temp2$ 为地址将 $temp1$ 写入交叉关系 RAM 中; 若 $temp2 < temp1$, 则以 $temp1$ 为地址将 $temp2$ 写入交叉关系 RAM 中, 这样等价关系 RAM 中每个以标记值为地址存的都是与其有交叉关系的最小值。

在第三阶段, 对交叉关系 RAM 进行重新整理, 重新按照自然数顺序将交叉关系 RAM 中初标记结果替换成新的标记值, 同时根据交叉关系更新坐标 RAM 中新目标的边界值, 对于两个具有交叉关系的目标 A 和 B, 将边界信息更新为:

$$x_{Cmin} = \min(x_{Amin}, x_{Bmin});$$

$$x_{Cmax} = \max(x_{Amax}, x_{Bmax});$$

$$y_{Cmin} = \min(y_{Amin}, y_{Bmin});$$

$$y_{Cmax} = \max(y_{Amax}, y_{Bmax});$$

在经历 3 个阶段的处理之后, 交叉关系 RAM 中每个不同初标记值都替换为按照自然数排列的新标记值, 且坐标 RAM 中按照自然数顺序存储了所有目标的边界信息。

2.2.4 标识模块

通过顺序读取坐标 RAM 即可获取上一帧图像中各个运动目标的最小外接矩形信息, 将各目标的最小外接矩形的像素值设置为 $\{8'd255, 8'd0, 8'd0\}$, 其余像素点的像素值保持不变, 即可实现用红框实时框选多个运动目标的效果。在加框标记后, 将 24 位的 RGB 数据截取成 16 位的 RGB565 格式缓存入 DDR3 中, 截取方式为 $\{R[7:3], G[7:2], B[7:3]\}$, 其中 R、G、B 分别为 RGB888 格式中的红、绿、蓝分量。

2.3 视频显示模块

LCD 显示模块读取 DDR3 中加框后的 RGB565 格式数据, 重新扩展为 24 位的 RGB888 格

式数据, 通过一块 LCD 显示屏实时显示运动目标检测后的结果, 文章显示屏的驱动时钟为 50 MHz, 分辨率为 1024×600 , 由于 DDR 控制采用了乒乓操作, 读取速率比存储速率快, 输出视频的帧率可以达到 60 fps。

3 实验结果与分析

整体硬件平台基于赛灵思 FPGA 开发, 另外还包括一块分辨率为 1024×600 的 LCD 屏和 CMOS 摄像头, 系统通过摄像头采集到的视频分辨率为 1024×600 , 帧率为 38 fps, LCD 显示屏显示视频的帧率为 58 Hz。文章在照明条件较好的环境下进行测试, 背景和运动物体的亮度比大致为 5:1, 通过对两个运动物体进行检测跟踪, 评估整个系统的准确性和实时性。

传统的帧差法检测结果如图 5 所示, 文章提出的多目标帧差法检测结果如图 6 所示。可以看出, 文章中系统相对传统的帧差法能够准确检测出视频序列中更多的运动目标。影响本系统目标识别率(检测出运动物体的概率)和跟踪效果的因素主要是两个阈值, 帧间差分后的图像需要根据阈值分



图 5 单目标检测结果

Fig.5 Results of single moving target detection

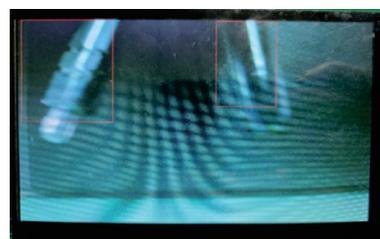


图 6 多目标检测结果

Fig.6 Results of multiple moving target detection

割,这个阈值需要根据具体实验检测环境的亮度来设定,合并标记框的阈值则需要根据实际检测的运动物体大小来设定。由于文章检测的运动物体相对较大,这里固定距离阈值为40,测试不同差分阈值情况下的目标识别率和跟踪效果,每个阈值统计20次检测结果,测试结果如表1所示。可以看出,当设置差分阈值为20时,目标识别率最高,且检测出来的目标有很好的连通性,当阈值逐渐增大时,差分二值化后的图像会出现许多空洞,目标很容易被分割成多个目标。

表1 不同阈值的检测结果

Tab.1 Detection results of different thresholds

阈值	目标识别率/(%)	单目标被认为多目标的概率/(%)
20	100	0
40	100	10
60	100	60
80	80	100
100	5	100

在相同实验条件下,采用PC对视频进行帧差分法处理,检测出的多个目标通过连通域进行标记,处理一帧像素为1 024×600的图像大约需要1.204 s,而文章提出的系统处理一帧相同大小的图像仅需要一帧时间,即0.026 s,可见文章设计的系统实时性较高,尤其在处理高清视频时优势巨大。整个系统的资源利用消耗情况如表2所示。可见,文章提出的运动目标检测方法硬件实现所消耗的资源总体较少,还可以使用丰富的资源来完善检测算法和扩展系统功能。

4 结 论

文章利用FPGA能够并行处理的优势,融合帧间差分法、基于距离阈值的目标分割算法和边框合并算法,结合CMOS图像传感器、外部存储器和显示器,设计了一个多运动目标实时检测系统。实验

表2 FPGA的资源使用情况

Tab.2 Resource utilization of FPGA

资源类型	占用量	总量	占用率/(%)
LUT	9 152	20 800	44
LUTRAM	576	9 600	6
FF	6 656	41 600	16
BRAM	20	50	40
IO	95	250	38

结果表明:该系统集成度高、功耗低、成本低以及实时性高,且能够准确检测出运动目标。但由于文章设计系统中的连续两帧差分图像存在典型的空洞及残缺问题,单个运动目标的差分图像不连续,一个目标可能被检测成多个目标。尽管已经设计了标记框合并算法减小这种影响,但仍会造成阈值设置过小一个目标被识别成两个,阈值设置过大两个相邻小目标被识别成一个的情况,如何根据环境亮度和检测目标的大小自适应调整阈值还需要进一步的研究。

参 考 文 献

[1] 胡以静,李政访,胡跃明.基于光流的运动分析理论及应用[J].计算机测量与控制,2007,15(2):219-221.

[2] 张 能.基于FPGA的运动目标跟踪系统设计与实现[D].上海:东华大学,2014.

[3] 杨叶梅.基于改进光流法的运动目标检测[J].计算机与数字工程,2011,39(9):108-111.

[4] 彭丁聪.卡尔曼滤波的基本原理及应用[J].软件导刊,2009,8(11):32-34.

[5] Zhang R, Jian D. Object tracking and detecting based on adaptive background subtraction[J]. Procedia Engineering, 2012, 29(1): 1351-1355.

[6] Mohamed S S, Tahir N M, Adnan R. Background modelling and background subtraction performance for object detection [C]. 2010 6th International Colloquium on Signal Processing & its Applications, Malacca, Malaysia, 2010: 1-6.

[7] 陈鑫元,李 筠,杨海马,等.自适应阈值图像二值化及形态学处理的FPGA实现[J].电子测量技术,2016,39(7):67-71.

[8] 李 鹏.基于FPGA的运动目标检测算法实现[D].西安:西安电子科技大学,2020.



(上接第73页)

[9] 何立夫,陆佳政,刘 毓,等.输电线路山火可见光-红外多光源精准定位技术[J].高电压技术,2018,44(8):2548-2555.

[10] 陆晓杰,韩贵丞,姚 波,等.热红外相机双目测距系统标定技术研究[J].激光与红外,2020,50(5):567-572.

[11] 陶重彝,乔 荔,孙云飞,等.基于双目视觉的六旋翼无人机立体匹配算法[J].激光与红外,2018,48(9):1181-1187.

[12] 杨晓文,曹山海,韩 燮.基于边界特征的三维模型分割[J].

计算机工程与应用,2019,55(4):214-218,232.

[13] 左克铸,邹媛媛,李鹏飞,等.基于迭代的等厚激光拼焊焊缝边界特征点识别方法[J].应用激光,2018,38(3):430-435.

[14] 卫保国,张玉兰,周佳明.图像匹配中的特征点筛选方法[J].计算机工程与应用,2021,57(3):208-214.

[15] 高东东,徐晓婷,李 博.红外/白光混合补光系统在智能交通中的应用研究[J].红外与激光工程,2018,47(9):337-343.