文章编号:1671-4598(2024)05-0072-08 DOI:10.16526/j.cnki.11-4762/tp.2024.05.011 中图分类号:TP391.4 文献标识码:A

# 融合边缘信息多运动目标检测识别系统设计

严 飞[1,2],徐 龙[1],孟 川[1],李 楚[1],刘 佳[1,2]
(1.南京信息工程大学 自动化学院,南京 210044;
2.江苏省大气环境与装备技术协同创新中心,南京 210044)

**摘要**:运动目标检测算法易受到噪声和亮度变换的影响,从而出现虚假目标,很难满足在实际应用中准确性和实时性的需求;针对上述问题,提出了一种融合边缘信息的多运动目标实时检测算法,并在FPGA平台上构建了视频图像处理系统;该算法首先使用高斯滤波增强算法来提高抗噪性能;然后,采用三帧差分法减少虚假目标的出现概率;同时,对图像进行边缘检测,并将边缘信息与帧差法的结果相融合;最后,运用多目标识别算法完成对运动目标的识别和标记;该算法实现充分利用了FPGA的并行计算和流水线操作,经实验测试检测识别系统能够在1 280×720@60 Hz视频场景下实现对最多16个运动目标的实时检测。

**关键词**:FPGA;图像检测;多目标;边缘检测;实时性

# Design of Multi Moving Target Detection and Recognition System Based on Edge Information Fusion

YAN Fei[1,2], XU Long[1], MENG Chuan[1], LI Chu[1], LIU Jia[1,2]
(1. Automation College, Nanjing University of Information Science & Technology, NanJing 210044, China;
2. CICAEET, Nanjing University of Information Science & Technology, Nanjing 210044, China)

**Abstract**: Motion target detection algorithms are easily affected by noise and changes in brightness, leading to the presence of false targets, making it difficult to meet the requirements of accuracy and real-time performance in practical applications. To address these issues, this paper proposes a real-time multi-object motion detection algorithm that incorporates edge information and constructs a video image processing system on FPGA platform. Firstly, the algorithm utilizes a Gaussian filtering enhancement algorithm to improve noise resistance. Subsequently, the three-frame difference method is employed to reduce the probability of false targets. Simultaneously, the edge detection is applied to the images, and the edge information is fused with the results of the frame difference method. Finally, the multi-object recognition algorithm is used to identify and label the moving targets. The algorithm fully utilizes the parallel computation and pipeline operations of FPGA. Experimental tests demonstrate that the detection and recognition system can achieve the real-time detection of up to 16 moving targets in 1280×720@60 Hz video scenes.

**Keywords**: FPGA; image detection; multiple objects; edge detection; real-time performance

## 0 引言

近年来,随着计算机视觉和图像处理领域的迅速发展,运动目标检测系统在众多实时应用中扮演着重要的角色[1]。这些系统能够实时、准确地检测和跟踪场景中的运动目标,为视频监控[2-3]、交通管理[4]、无人机导航[5]等领域提供了关键的解决方案。同时,它们还可为目标跟踪系统[6]提供初始目标坐标,从而实现持续的跟踪和定位。

目前,主流的运动目标检测算法包括光流[7]、背景差分法[8]、帧间差分法[9]、ViBe算法[10]、角点提取法以及基于深度学习的算法,如YOLO[11]和CNN[12]等。光流法通过比较连续帧之间像素灰度值的变化来计算像素的运动向量以检测运动目标。然而,它的计算复杂度较高,对硬件资源的需求也较大,难以硬件实现。背景差分法则是将每个视频帧与背景模型进行比较,若差异超过预定阈值,则将其视为运动目标。然而,一旦背景发生变化,就需要及时更新背景模型。ViBe算法主要用于静态背景下的运动目标检测,算法涉及到复杂的像素更新和比较操作硬件实现需要大量的资源,功耗较高。角点提取法通过寻找图像中明显变化的像素点来确定目标的位置和边界,对于平滑区域或缺乏纹理的区域往往无法提取有效的角点。深度学习算法目前在精度上具有相当高的性能,但其复杂度较高,在实际嵌入式应用中受制于功耗、速度、成本等因素,在硬件上实现相对困难。帧间差分算法拥有较好的实时性并且更新速度快、算法简单、计算量小,同时FPGA具有独特的并行处理能力,可实现对高清视频图像的实时处理,因

收稿日期:2023-10-13; 修回日期:2023-12-06。
基金项目:国家自然科学基金项目(61605083);江苏省产业前瞻与关键核心技术重点项目(BE2020006-2)。
作者简介:严 飞(1983-),男,博士,副教授。
通讯作者:刘 佳(1981-),女,博士,教授。
引用格式:严 飞,徐 龙,孟 川,等.融合边缘信息多运动目标检测识别系统设计[J].计算机测量与控制,2024,32(5):72-79.

此越来越多的研究团队选择帧差算法并通过 FPGA 硬件平台搭建运动目标检测系统。邢凯等采用了背景差分法和帧间差分法，并结合包围盒技术，成功实现单一运动目标的准确检测[13]。刘汝卿等进一步实现了运动目标的实时跟踪和动态显示，通过实验测出在一定范围内可稳定跟踪运动目标[14]。黄明政等在帧间差分法的基础上，提出了动态帧差法实现了对低速和高速运动目标的自适应实时检测[15]。刘存领等基于帧差法实现对多运动目标检测，检测结果通过以太网传输至上位机实时显示[16]。然而，目前已有的基于帧间差分算法的检测系统易出现漏检、鬼影的现象，从而无法准确提取运动目标，并且难以处理高帧率的高清视频图像，实时性仍有待提高。除此之外，以上系统均只支持通过摄像头采集的视频源，无法更好适应实际应用需求。

针对上述问题，本文在三帧差分算法的基础上设计了一种融合边缘信息的多运动目标检测识别算法。一方面，该算法将三帧差分与图像边缘信息相结合，使图像的边缘信息不易受噪声和亮度的影响，从而有效提高了对运动目标检测的准确率。另一方面，基于 FPGA 灵活性高、运行速度快、兼容性强等特点，搭建了多运动目标检测识别系统，且支持上位机通过 HDMI 采集视频源。该系统能够实时检测出图像中的运动目标，并对检测得到的二值化图像进行目标识别，分割出各个目标并进行标记。

## 1　系统总体设计

为了实现多个运动目标的检测和标记，提出了一种基于 FPGA 的多运动目标检测识别系统，该系统主要由图像采集与显示模块、图像缓存模块、运动目标检测模块、多目标识别模块四部分组成。整体流程框图如图 1 系统流程总体框图所示。

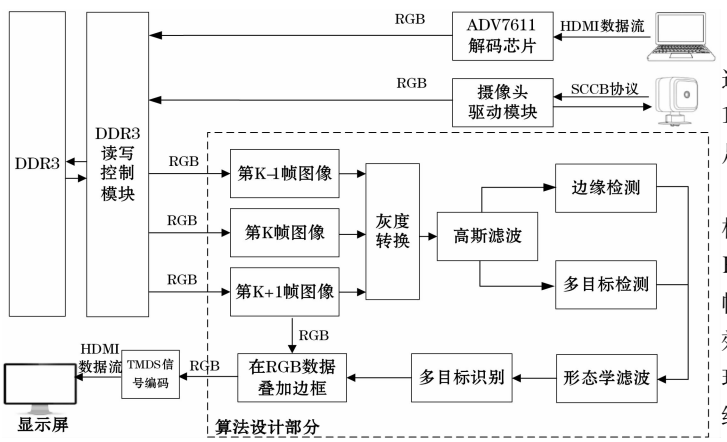



图 1　系统流程总体框图

图像采集与显示模块通过 FPGA 控制输入视频图像时序，并将算法处理后的结果图像进行 TMDS 编码，再使用 HDMI 接口输出至 LCD 屏实时显示；图像缓存模块对不同帧率的视频源均进行三帧缓存；运动目标检测模块将缓存的图像数据进行预处理、Sobel 边缘检测、三帧差分法、形态学滤波等处理生成仅包含运动目标信息的二值图像；多目标识别模块将检测的运动目标进行分割、定位，滤除干扰噪点并对各个目标进行标记。

### 1.1　图像采集与显示模块

系统采集的图像分为两种来源：一种是摄像头直接采集，另一种是上位机通过 HDMI 输出的视频源。该模块不仅能够满足更多的应用场景，还增加了系统的灵活性和多样性。

摄像头采集视频数据的具体过程如下：FPGA 通过 SCCB 接口协议向摄像头发送一系列配置命令，其中包括设置图像分辨率、曝光时间、帧率以及其他图像参数等。一旦 FPGA 向摄像头发送采集指令，便触发了图像采集过程。摄像头开始将图像数据转换为数字信号，并通过数据接口将信号传送给 FPGA。视频源帧率为 50 fps。

上位机发送的 HDMI 视频流需要进行解码和编码操作。本系统采用 ADV7611 视频解码芯片作为硬核解码器。每次上电只需在 FPGA 内部使用 SCCB 协议来复位芯片并配置相关寄存器，即可稳定地将 HDMI 协议码流解码为 VGA 时序，稳定获取上位机发送的 60 fps 视频源。

### 1.2　图像缓存模块

系统接收由 HDMI 输入的视频流为 1 280×720@60 hz，每秒数据量带宽 $Vedio_{bw}$ 计算如下：

$$Video_{bw} = H_p \times V_p \times f_r \times pp \tag{1}$$

其中，$H_p$ 表示每行像素点个数，$V_p$ 表示像素点行数，$f_r$ 表示帧率，$pp$ 表示单个像素点位宽。

由公式（1）计算可得系统最大输入视频流带宽约为 1.04 GB/s。调用核心板自带 4 片 DDR3 对视频流进行缓存，其内存为 16 Gbit，数据位宽为 64 bit，则 DDR3 带宽 $DDR_{bw}$ 可以计算为：

$$DDR_{bw} = data_{width} \times rate_t \tag{2}$$

其中：$data_{width}$ 为数据位宽，$rata_t$ 为内部时钟速率。

由于系统内部采用双沿采样数据传输，所以内部时钟速率 $rata_t$ 为 1 600 MHz。带宽 $DDR_{bw}$ 由公式（2）计算为 102.4 GB/s。系统的图形缓存模块在内存与带宽方面均满足算法运行的需求。

为了方便控制 DDR3 芯片，本系统通过调用 MIG IP 核为用户提供 APP 端接口。同时，为了兼容摄像头和 HDMI 时序，本系统在 DDR3 内部划分 4 个缓存区域用于帧缓存，并采用跨 Bank 乒乓读写[17]方式提升 DDR 读写效率。如图 2 所示，写端口可以将任意时序视频流按帧循环写入 4 个缓存区域，并且输出端口始终为固定的三帧连续视频流：$F_{k-1}(x,y)$、$F_k(x,y)$ 和 $F_{k+1}(x,y)$。因此，视频序列不会发生帧撕裂的情况，后续的图像处理操作只需按顺序使用三帧视频流即可。

本系统的图像缓存模块能够实现 HDMI 接口或摄像头采集的视频源输入，经 DDR3 缓存后进行多路视频流输出，解决了输入视频时序与输出端 VGA 时序不匹配的问题。

## 2　运动目标检测模块

### 2.1　融合边缘信息运动目标检测算法

将边缘信息与运动目标检测算法相结合[18]可以有效抑

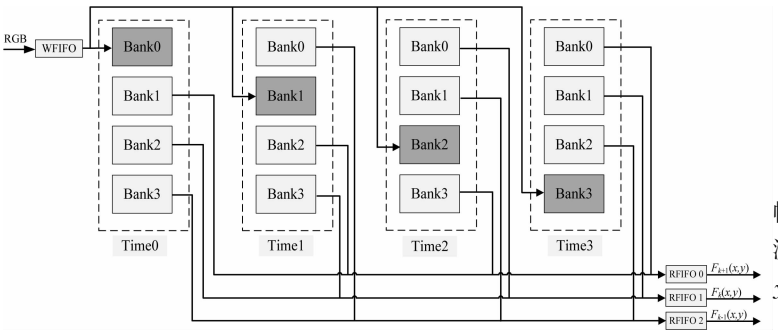



图 2 跨 Bank 乒乓读写示意图

制噪声和增强边缘，减少误报和漏报的情况，更好地适应动态场景，提高对移动对象的检测能力。在此基础上提出一种改进运动目标检测算法，具体算法设计框图如图 3 所示。首先，对图像进行预处理操作，以降低处理的复杂度、减少不必要的细节和抑制干扰。接着，执行三帧差分运算并进行膨胀操作，以缓解三帧差分对目标边缘部分产生的欠检测问题。最后，将边缘检测结果与帧差结果结合，并应用形态学处理完成对运动目标的检测。

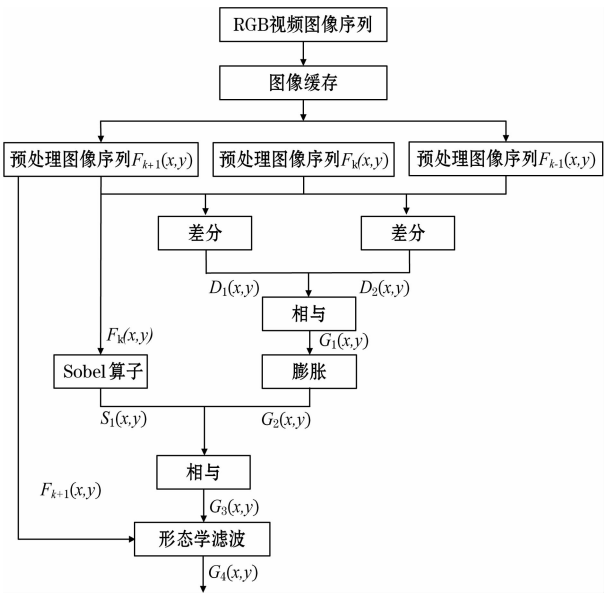



图 3 运动目标检测算法框图

预处理：3 个读 FIFO 同时从 DDR3 的读缓存区域读取连续的三帧图像序列，先进行预处理操作，将 RGB 视频流转灰度，以减少处理的复杂度。接着进行高斯滤波操作，通过平滑化图像，使边缘更加明显，从而减少不必要的细节和干扰，增强算法的抗干扰能力。

三帧差分：预处理后得到的连续三帧图像序列 $F_{k-1}(x,y)$、$F_k(x,y)$ 和 $F_{k+1}(x,y)$，相邻两帧差分处理并根据阈值 $T_2$ 得到二值化图像 $D_1(x,y)$ 和 $D_2(x,y)$，接着将两帧差所得的二值图像做与运算获得最终的二值图像，其计算公式为：

$$D_1(x,y)=\begin{cases}1, |F_k(x,y)-F_{k-1}(x,y)|>T_2\\0, |F_k(x,y)-F_{k-1}(x,y)|\leqslant T_2\end{cases} \quad (3)$$

$$D_2(x,y)=\begin{cases}1, |F_k(x,y)-F_{k+1}(x,y)|>T_2\\0, |F_k(x,y)-F_{k+1}(x,y)|\leqslant T_2\end{cases} \quad (4)$$

$$G_1(x,y)=\begin{cases}1, D_1(x,y)\cap D_2(x,y)=1\\0, D_1(x,y)\cap D_2(x,y)\neq 1\end{cases} \quad (5)$$

膨胀：对帧差结果进行膨胀操作可以有效解决三帧差对目标的边缘部分产生欠检测的问题，同时使检测到的运动目标在图像中更加明显。二值图像 $G_1(x,y)$ 的膨胀操作如式（6）。

$$G_2(x,y)=\max_{(s,t)\in E}\{G_1(x-s,y-t)\} \quad (6)$$

式中，$G_2(x,y)$ 示膨胀后图像中坐标为的像素值，$E$ 为膨胀操作卷积核，$(s,t)$ 是卷积核中的坐标。

结合 Sobel 算子：三帧差分算法是基于像素差分的计算，可能无法准确地检测出运动物体的边缘信息，从而导致检测结果的边界模糊。由于 Sobel 算法有助于抑制高噪声，能够得到更准确的运动物体轮廓，同时可以解决三帧差分算法对快速移动的物体和光照变化等因素较为敏感的问题，因此，将 $G_2(x,y)$ 与第帧 Sobel 算子结果 $S_1(x,y)$ 做与运算得到 $G_3(x,y)$，如式（7）所示：

$$G_3(x,y)=\begin{cases}1, G_2(x,y)\cap S_1(x,y)=1\\0, G_2(x,y)\cap S_1(x,y)\neq 1\end{cases} \quad (7)$$

形态学滤波：为了同时消除图像中细小的噪声点和物体的突出部分，使物体的轮廓更平滑，便于多目标识别模块准确识别出运动目标的个数，所以对 $G_3(x,y)$ 使用“开”运算，公式表示为，

$$G_4(x,y)=(G_3(x,y)\ominus A)\oplus B \quad (8)$$

式中，$A$ 是“与”运算模板，$B$ 是“或”运算模板，$\ominus$ 表示“与”运算，$\oplus$ 表示“或”运算。

## 2.2 运动目标检测算法的 FPGA 实现

FPGA 在数字电路设计和应用中的最大优势就是可以使用并行和流水线操作。流水线操作可以在一个时钟周期内执行一部分计算，并将结果传递给下一个级别进行下一步计算。并行操作则可以使多个模块同时执行相同或不同的操作，每个模块可以处理不同的数据。这种并行性和流水线操作的组合能让 FPGA 同时处理多个数据流，并在时钟周期内产生部分结果，从而提高整体的计算效率。

本系统采用并行和流水线的组合操作，在进行图像处理时分割成多个逻辑小块。如图 4 所示，包括：转灰度、高斯滤波、帧间差分、Sobel 边缘检测、形态学滤波和目标分割框选等。在 t5 时刻，本系统所有模块均进行数据的处理计算，同时在模块的内部进行的复杂计算，如卷积操作则使用多级流水线操作，使得每个阶段能够在一个时钟周期内完成部分计算。通过将图像处理任务划分成多个阶段，在多个模块之间进行并行操作，从而大大提高了系统整体的处理效率和实时性能。

### 2.2.1 图像灰度处理

将彩色视频图像转换为灰度视频图像可有效降低数据的处理量，却会导致显示效果下降，因此本系统将灰度和彩色视频流分开，灰度图像用于目标检测，而彩色图像用

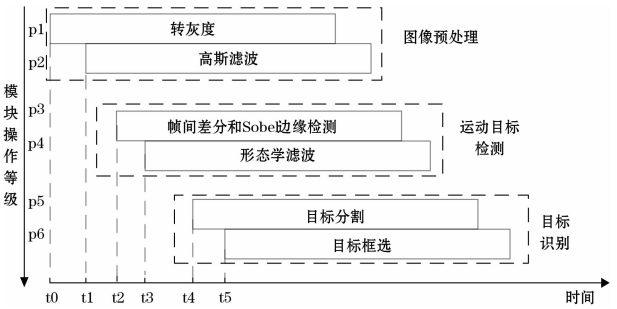



图 4　图像算法并行流水线处理

于跟踪和显示，以平衡处理效率和显示质量。

在 FPGA 中，通过 DDR3 缓存的视频流，每秒包含 60 帧彩色图像。直接处理彩色视频图像会涉及大量计算和较长的运行时间。为了降低计算复杂度并减少色彩相关性的影响，需要对 DDR3 缓存后的 24 位彩色视频图像进行灰度处理[19]，以获得 8 位灰度视频图像。灰度处理的公式如下：

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \tag{9}$$

式中，$R$、$G$、$B$ 表示红、绿、蓝分量，$Y$、$Cb$、$Cr$ 表示灰度分量、色调、饱和度。

由于 FPGA 芯片无法计算浮点数，需对灰度计算的参数扩大 $2^8$ 倍再右移 8 bit，将其转换成 FPGA 易处理的整数运算，计算公式如下：

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \left\{ \begin{bmatrix} 77 & 150 & 29 \\ -43 & -85 & 128 \\ 128 & -107 & -21 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 32\,768 \\ 32\,768 \end{bmatrix} \right\} >> 8 \tag{10}$$

在电路设计方面采用流水线设计方式计算 $Y$、$Cb$ 和 $Cr$，只需要 3 个时钟即可输出结果。

2.2.2　高斯滤波处理

FPGA 接收的视频图像不可避免地含有噪声。高斯滤波作为一种线性平滑滤波器，在抑制图像输入时的随机噪声方面表现出色[20]，因此在图像处理和信号处理领域得到广泛应用。对于灰度图像，高斯滤波使用 3×3 的卷积核，沿图像从左到右、从上到下进行逐一扫描。再通过计算卷积核范围内像素的加权平均值，替代该范围内中心像素，实现图像的平滑处理。

算法模块采用了三级流水线设计，按照自上而下的顺序扫描图像。在 FPGA 上实现这一计算过程时，首先建立了一个行缓存结构，用于存储待处理的三行像素数据。调用两个 FIFO 行缓存，将数据组合成一个 3×3 的矩阵，然后通过 3 个输出端同时进行并行输出到计算模块。这样，一次卷积计算处理涵盖了三行三列的数据，最后，输出值经过向右移位 4 位的归一化处理，得到了中心像素点的高斯滤波结果。通过这种方式将视频图像根据行、场时序进行遍历，不仅可以减少板上资源的消耗，同时提高了滤波处理的速度。

2.2.3　Sobel 图像边缘提取

Sobel 边缘检测[21]是一种广泛应用的图像处理算法，能够有效地提取图像中的边缘信息。与 Roberts 和 Prewitt 算子相比，Sobel 算子在边缘提取性能上表现更优越。相较于 LOG 和 Canny 算子[22]，Sobel 算子具有更低的运算复杂度，更易于在硬件上实现，并且具备更好的实时性[7-9]。因此，本系统选择采用 Sobel 算子来获取图像的边缘信息。

Sobel 算子的本质是对图像进行像素级的卷积运算，通过计算每个像素点与其周围像素点的灰度值差异来确定该像素点的梯度值。具体来说，Sobel 算子由两组 3×3 大小的卷积核，即 $k_1$ 和 $k_2$ 组成，其中一个用于横向，另一个用于纵向边缘检测。使用 $Z$ 表示原始图像，那么横向和纵向的边缘亮度变化值 $G_x$ 和 $G_y$ 可以通过将这两组卷积核与原始图像进行卷积运算得到，公式表示如下：

$$\begin{cases} G_x = k_1 Z = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} Z \\ G_y = k_2 Z = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} Z \end{cases} \tag{11}$$

然后，$G_x$ 和 $G_y$ 进行均方根运算得到近似的梯度值 $G$。再通过设置的阈值 $T_1$ 判断该点是否为边缘像素点，具体公式如下：

$$G = \sqrt{G_x^2 + G_y^2} \tag{12}$$

$$\begin{cases} G > T_1, \text{边缘} \\ G \leqslant T_1, \text{非边缘} \end{cases} \tag{13}$$

如图 5 所示，基于 FPGA 的 Sobel 算子使用两个行缓存 FIFO，输出值通过寄存器打拍组合成 3×3 矩阵。整个计算过程可以分解为两个子计算：$G_x$ 与 $G_y$ 加减法计算。$G_x$ 是指 $G_{x1}$ 与 $G_{x2}$ 做差的绝对值，其中 $G_{x1}$ 与 $G_{x2}$ 分别表示横向加法计算结果与减法计算结果，$G_y$ 同理。$G_x$ 和 $G_y$ 求平方和后经过 CORDIC IP 核计算输出结果，最后根据设定的阈值判断是否为边缘。

2.2.4　形态学滤波处理

基本形态学滤波有腐蚀和膨胀两种，也叫作“与”运算和“或”运算。图像先进行“与”运算再进行“或”运算的操作方式被称为“开运算”。其中“与”运算可腐蚀图像的边缘部分，“或”运算可扩张图像中目标点的边缘部分。因此，本系统采用“开运算”先腐蚀后膨胀的方法去除二值图像中的小噪声点或孤立的像素，具体 FPGA 逻辑设计如图 6 所示。

分别使用两个 RAM 行缓存将数据组成 3×3 矩阵，然后对矩阵内的九个二值元素执行“与”和“或”操作。为了确保电路在高频时钟下稳定运行，采用了二级流水线处理，“与”和“或”运算在每个级别中分别处理 3 个元素的组合运算，以降低组合逻辑所消耗的时间。

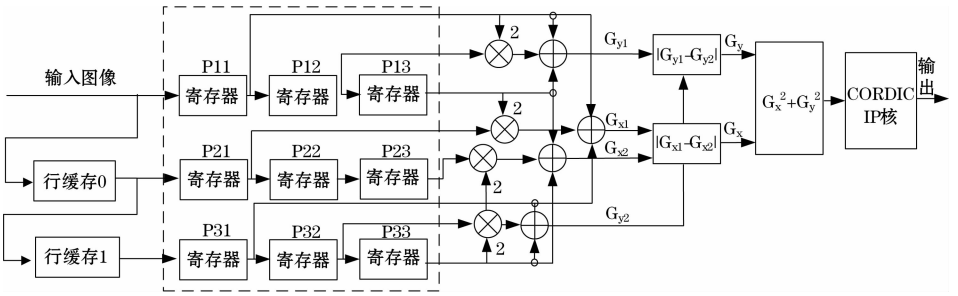



图 5 Sobel 算法设计框图

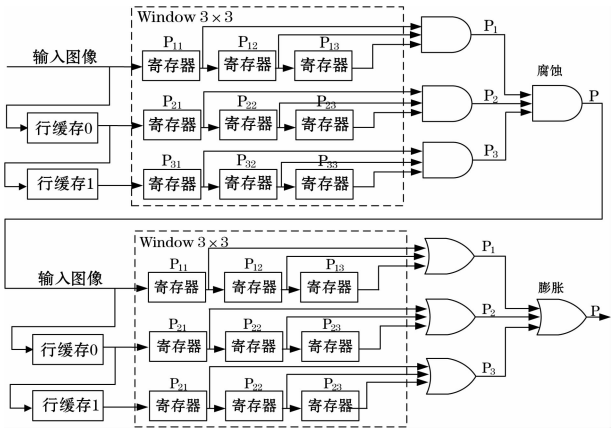



图 6 形态学滤波设计框图

## 3 多目标识别模块

### 3.1 多目标识别算法

针对检测后的各目标内部空洞易造成目标标记误判的情况，在连通域标记思想的基础上提出一种基于已有目标阈值判决的多目标识别算法，该算法的计算步骤如下：

1）首先根据各目标在图像中的大小，选择合适的阈值 $T_3$，用于判决新检测的运动像素点与已识别目标的关系。检测到的运动像素点 $P$ 会根据图片由左到右，再由最上行至最下行依次进行判决。

2）若运动像素点 $P$ 与图像中所有已识别目标边界 $N(i)$ 之间距离的最小值 $\min(dist(P,N(i)))$ 大于预设的阈值 $T_3$，则判决结果 $C(P,i)$ 为 1，反之为 0。

3）根据每一个已识别的目标对运动像素点 $P$ 的判决结果，若仅有一个已识别目标判决结果 $C(P,i)$ 为 0，那么此时运动像素点 $P$ 与该已识别目标属于同一目标。如图 7（a）所示，目标的边界随像素点 $P$ 更新。

4）若已识别目标对运动像素点 $P$ 的判决结果 $C(P,i)$ 均为 1，则此时像素点被视为新目标，如图 7（b）所示。

5）若多个已识别目标对运动像素点的判决结果 $C(P,i)$ 为 0，则此时同时判决为 0 的运动目标将被定为同个目标，像素坐标进行合并，坐标随之更新，如图 7（c）所示。

### 3.2 基于 FPGA 的多目标识别模块

在 FPGA 中实现多目标识别算法主要通过分割模块和加框模块共同完成。如图 8 目标分割仿真波形图所示，分割模块通过调用 49 位宽 16 位深的二维数组 target _ pos 来存储至多 16 个运动目标的上、下、左、右坐标，其中识别目标 $x$ 和 $y$ 方向的最大值和最小值各占 12 位宽，最高位是指示位 flag，若指示位为 1 代表已存储运动目标坐标，否则该寄存器为空。在一帧的数据使能信号 per _ frame _ clken 拉高时，数组 target _ pos 根据检测的运动目标像素值 per _ img _ Bit 实时计算更新，target _ cnt 存储已识别目标个数。

由于数组无法在 Vivado 平台实现跨模块传输，所以当场同步信号 per _ frame _ vsync 是上升沿时，数组 target _ pos 内的 16 组数据将会通过 0～15 计数器 boundary _ cnt 依次赋值给寄存器 boundary _ data，并发送给加框模块，以标记各目标的轮廓。具体代码如下所示：

$$boundary\_data <= target\_pos[boundary\_cnt]$$

加框模块则在收到寄存器中的数据后，将计数器作为存储地址存储在位宽、深度相同的二维数组，该数组中的数据会在下一个上升沿出现时进行更新，否则数据保持不变。在场同步信号 per _ frame _ vsync 的下降沿，分割模块数组中的数据会进行清空，以存储下一帧识别的运动目标信息。

对于判决操作，若数组 *target _ pos* 中深度为 $i$ 的一组数据至运动像素点的距离大于阈值 $T_3$，则判决的最终结果 new _ target _ flag 的第 $i$ 位为 1，否则将对应位置 0，由此不断计算获得一帧中识别的目标数 *target _ cnt*。实现目标数计算的逻辑代码如下所示：

$$target\_cnt <= (target\_flag = 16'hffff)?(target\_cnt+1):target\_cnt$$

在场消隐期间，加框模块根据计算获得的目标坐标

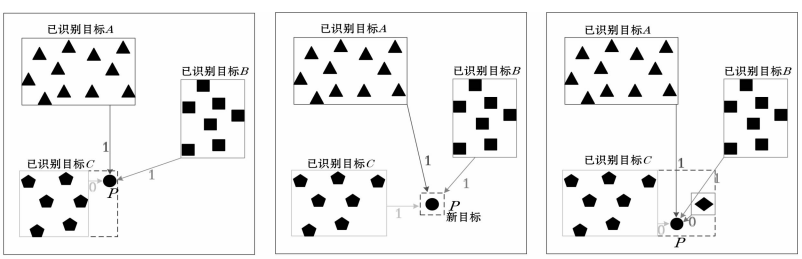



图 7 目标分割示算法示意图

boundary _ data 对 RGB 图像进行框选。考虑到光照等因素产生的噪点会影响最终目标个数的判断，因此在识别算法中增加了筛选功能，并根据待检测目标的大小选择合适的筛选阈值 $T_4$，若边框长度小于阈值 $T_4$ 则将对应边框筛除。本系统将阈值设置为 20，通过筛选功能进一步提高了目标个数的识别准确率。

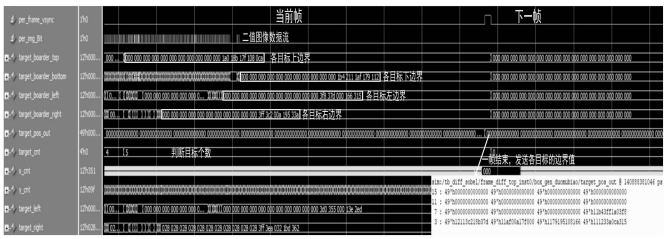


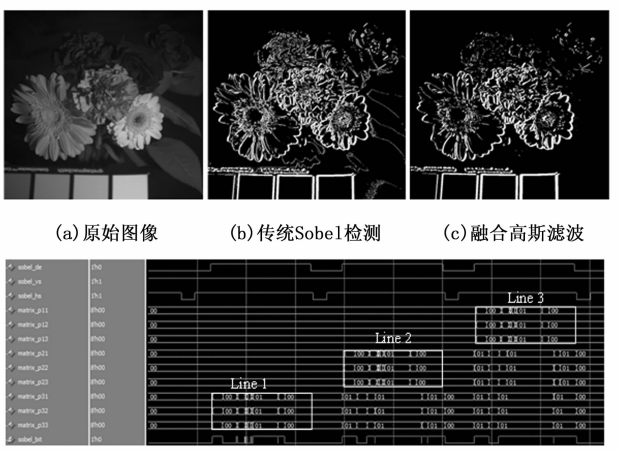

图 8　目标分割仿真波形图

## 4　验证与分析

### 4.1　模块仿真验证

通过建立 Modelsim 仿真平台，对各个模块进行仿真验证并生成可视化图像，具体结果如下所示。

与传统 Sobel 边缘检测算法的对比实验结果如图 9 所示。其中，图 9（a）是原始图像，图 9（b）是传统的 Sobel 边缘检测结果，图 9（c）是系统增加高斯滤波预处理后的 Sobel 边缘检测检测结果。相比于传统 Sobel 边缘检测算法，系统融合后的结果具有更好的边缘定位精度，并有效过滤掉不重要的信息。此外，图 9（d）还展示了相应的时序仿真波形。在此仿真中，设定阈值 $T_1$ 为 70，并通过时钟信号每个时钟周期内获得一行中的连续 3 个像素。一旦第一行的数据结束，数据会依次流向两个行缓存 FIFO，以获得 3×3 图像模板，最终的计算结果为 sobel _ bit = 1’b1，则被判断为边缘信息。整个过程总共用时 13 个时钟周期即 87.1 ns。

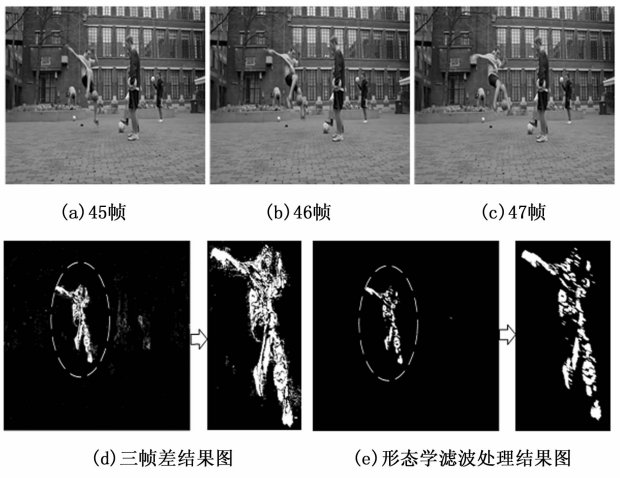



(a) 原始图像　(b) 传统Sobel检测　(c) 融合高斯滤波

(d) Sobel边缘检测仿真图

图 9　Sobel 边缘检测结果对比

图 10 展示了形态学滤波处理的结果。图 10（a）～（e）分别为连续三帧图像、三帧差和开运算结果图。图 10（e）中形态学滤波处理在保留大部分运动目标信息的同时有效去除了目标周围的微小噪点。



(a) 45帧　(b) 46帧　(c) 47帧

(d) 三帧差结果图　(e) 形态学滤波处理结果图

图 10　形态学滤波运算结果

图 11 展示了增加目标筛选功能前后的目标检测效果。如图 11（a）所示，无目标筛选功能的检测结果图中包含不希望被识别的像素点（虚线处），而添加目标筛选功能后的结果图 11（b）成功过滤掉无效运动元素，提高了目标检测的准确率。

(a) 无筛选功能　(b) 增加筛选功能

图 11　增加目标筛选结果对比

### 4.2　系统测试与分析

系统使用 Xilinx 公司提供的 XC7K325TFF900-2 型号 FPGA 芯片，并基于 Modelsim 构建了仿真平台。在实验中，由上位机连续发送 PETS2009 数据集，并进行了一系列对比实验，包括传统的 2 帧差分算法、3 帧差分算法以及本文提出的算法，以验证运动目标检测性能。

(a) 第387帧　(b) 第388帧　(c) 第389帧

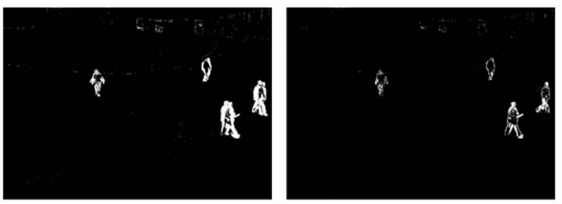
图 12　连续 3 帧图像

其中，图 13（a）展示了通过 2 帧差分计算得到的跟踪结果，图 13（b）则展示了经过 3 帧差分计算得到的跟踪结果，图 13（c）呈现了采用改进目标检测算法并融合 Sobel

算子后得到的二值化图像，图 13（d）展示了本文提出的系统算法的检测结果图像。

如图 13（a）所示，2 帧差分法在检测快速移动目标时存在明显的重影问题，导致准确性不高。而对于图 13（b），传统的 3 帧差分法虽然有效地解决了目标重影问题，却导致目标边缘模糊且不连续，易将同一目标误判为多个小目标。这两种检测方法都容易受到背景干扰的影响，从而影响运动检测的准确性。

然而，如图 13（c）所示，本文提出的多目标检测方法在目标检测时能够更好地保留目标的边缘信息，同时有效去除了噪声、重影以及背景干扰的影响，表现出了较高的鲁棒性。最后，图 13（d）在 RGB 图像上展示了本文算法在多运动目标跟踪方面的表现结果。



(a) 2帧差分算法　(b) 3帧差分算法

(c) 改进目标检测方法　(d) 多运动目标检测结果

图 13　仿真实验结果对比

### 4.3　FPGA 板上跟踪结果分析

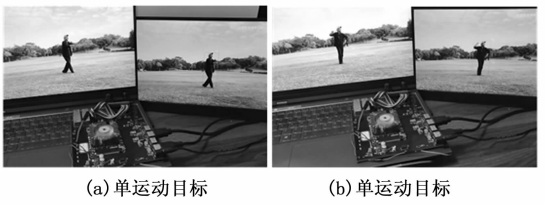
在 FPGA 板上进行改进的目标检测算法测试时，能够与上位机通过 HDMI 接口发送的视频源和摄像头采集的视频源兼容。基于 HDMI 接口发送视频源的目标检测实验包括两组：单目标检测和多运动目标检测。

如图 14（a）和（b）所示，图片内左侧显示为上位机通过 HDMI 向 FPGA 发送的视频源，右侧为 FPGA 检测结果输出，检测目标使用红色矩形框标记。图 14（c）和（d）展示了两个场景下的多运动目标检测结果。如图 14 所示，本系统能够有效地检测出 HDMI 接口发送的视频源中快速移动的目标。

摄像头采集的视频分辨率为 1 280×720，经过 FPGA 的乒乓操作后，在 LCD 显示屏上以 60 Hz 的帧率显示。图 15 展示了基于摄像头采集的视频图像多目标运动检测实验结果。如图 15 所示，本系统能够实时、准确地检测 CMOS 摄像头采集的视频图像中多个目标，处理一帧 1 024×768 的图像需要 16.7 ms，对于处理高清视频有巨大的优势。

### 4.4　帧率和资源占有率分析

本系统可支持 720 p@60 Hz 视频输入，实际上板处理速度 60 fps。文献［13］、文献［16］均为基于帧差算法实



(a) 单运动目标　(b) 单运动目标

(c) 多运动目标　(d) 多运动目标

图 14　基于 HDMI 接口运动目标检测结果图

(a) 出现多运动目标　(b) 出现多运动目标

(c) 出现多运动目标　(d) 单运动目标

图 15　基于摄像头检测结果图

现的运动目标检测系统。表 1 给出了本系统与上述系统的实时性对比实验结果。如表 1 所示，本系统可处理更高分辨率的视频流，并且在帧率上具有明显的优势。因此，本系统算法采用流水线处理方式，在输出图像的同时即能得到目标检测的结果，实时性较好。

表 1　基于帧差法目标检测系统实时性对比

| 检测系统 | 分辨率 | 检测速度/($f$/s) |
|---|---|---|
| 文献[13] | 1 280×720 | 15 |
| 文献[16] | 640×480 | 26 |
| 本系统 | 1 280×720 | 60 |

系统的资源使用情况如表 2 所示，主要逻辑资源 LUT 和 BRAM 的使用率均低于 10%，易于在不同 FPGA 芯片上实现，整个算法层面为轻量级加速，降低了系统应用的成本。

表 2　FPGA 资源使用情况表

| 资源类型 | 占有量 | 总量 | 占有率/% |
|---|---|---|---|
| LUT | 14 352 | 203 800 | 7.0 |
| LUTRAM | 864 | 64 000 | 1.4 |
| FF | 12 064 | 407 600 | 3.0 |
| BRAM | 27.5 | 445 | 6.2 |
| IO | 87 | 500 | 17.4 |

## 5　结束语

本文分别从图像的采集与显示、图像缓存、运动目标检测、运动目标识别等方面介绍了融合边缘信息的多运动目标检测识别系统的设计方法。为了提高算法的运行速度，在图像灰度转换、高斯滤波、形态学滤波、Sobel 边缘检测、形态学滤波的处理方式采用了并行流水线操作，在差分运算中使用乒乓操作进行视频图像的存取。通过实验验证了系统的功能和性能的准确性，实现了对多个运动目标的检测和识别，检测帧率为 60 fps 能够满足视频传输的实时性要求。此外，本系统对于 FPGA 资源消耗低，有利于系统的应用和后续功能的扩展，适应未来需求的变化。对于不同场景多个运动目标的检测识别具有较强的实用性和应用价值。目前，该系统的目标跟踪准确性受到 Sobel 阈值 $T_1$、帧差阈值 $T_2$、多目标判决阈值 $T_3$ 以及筛选阈值 $T_4$ 的影响。需要进一步研究如何将每个阈值与跟踪场景相关联，并实现自适应调整。

**参考文献：**


[1] 冯天桥，毛征宇，彭向前. 基于 FPGA 的运动目标实时检测系统设计 [J]. 电子技术应用，2023，49 (6)：99 - 103.

[2] 雷文静，王连明. 一种基于 FPGA 的多路视频网络监控系统设计 [J]. 现代电子技术，2019，42 (22)：48 - 52.

[3] 方　勇，张建军. 面向智能视频监控中的运动目标检测方法研究 [J]. 计算技术与自动化，2022，41 (4)：79 - 83.

[4] 彭　湃，耿可可，王子威，等. 智能汽车环境感知方法综述 [J]. 机械工程学报，2023，59 (20)：1 - 23.

[5] 陆渊章，戴红霞，胡　莹，等. 基于机器视觉的无人机协同目标跟踪算法研究 [J]. 电子器件，2020，43 (5)：1096 - 1099.

[6] 余田椿，孙先松. FPGA 的图像识别与目标跟踪系统设计 [J]. 单片机与嵌入式系统应用，2019，19 (9)：30 - 34.

[7] 任朝宇，赵冬娥，张　斌，等. 基于多尺度 CLG 光流法的多目标检测方法 [J]. 计算机与现代化，2022 (4)：33 - 37.

[8] 贾晓琪，闫俊伢，杨丽英. 基于背景差分法的行驶中的汽车检测 [J]. 山西电子技术，2021 (5)：47 - 49.

[9] 丁文龙，尹　朵，邱　崧. 改进的三帧差分运动目标识别算法 [J]. 智能计算机与应用，2022，12 (3)：180 - 182.

[10] 王欣宇，陈广锋，李　侠. 基于改进 ViBe 算法的运动目标检测 [J]. 东华大学学报 (自然科学版)，2023，49 (1)：95 - 102.

[11] 季娟娟，王　佳，陈亚杰，等. 基于改进 YOLO v4 的热轧带钢表面缺陷检测 [J]. 计算机工程与设计，2023，44 (9)：2786 - 2793.

[12] 代恒军. 基于改进的 Faster R-CNN 图像目标检测方法研究 [J]. 信息技术与信息化，2023 (8)：91 - 94.

[13] 邢　凯，李彬华，陶　勇，等. 基于 FPGA 的运动目标实时检测跟踪算法及其实现技术 [J]. 光学技术，2020，46 (2)：158 - 166.

[14] 刘汝卿，李　锋，蒋　衍，等. 基于 FPGA 的运动目标实时检测系统设计 [J]. 计算机测量与控制，2022，30 (4)：56 - 59.

[15] 黄名政，李彬华. 动态帧差法及其 FPGA 设计与实现 [J]. 光学技术，2023，49 (2)：231 - 237.

[16] 刘存领，林青松，于洪泽. 基于 FPGA 的运动目标远程监视系统设计 [J]. 计算机测量与控制，2023，31 (5)：21 - 27.

[17] 刘荷花，蔡江辉，王晓燕，等. 基于 FPGA 的视频监测存储系统设计 [J]. 仪表技术与传感器，2022 (4)：64 - 68.

[18] 苏文强，王永雄. 融合边缘信息的显著性目标检测 [J]. 小型微型计算机系统，2023，44 (10)：2291 - 2299.

[19] 严　飞，马　可，刘　佳，等. 无人机目标实时自适应跟踪系统 [J]. 计算机工程与应用，2022，58 (10)：178 - 184.

[20] 康　宇，赵冬青，上官鹏，等. 基于 FPGA 的图像边缘保护高斯滤波算法实现 [J]. 电子设计工程，2021，29 (6)：94 - 98.

[21] 张　琪，王　艺，陈兆飞. 基于 Sobel 算子图像预处理的目标检测算法 [J]. 电子技术与软件工程，2020 (5)：151 - 153.

[22] 王嘉俊，段先华. 改进 Canny 算子在水面目标边缘检测中的研究 [J]. 计算机时代，2020 (1)：35 - 38.


(上接第 71 页)


[17] LU S，LUO B，PATEL T，et al. Making disk failure predictions {SMARTer}! [C] //18th USENIX Conference on File and Storage Technologies (FAST 20)，2020：151 - 167.

[18] LU R，XU E，ZHANG Y，et al. Perseus：a {Fail-Slow} detection framework for cloud storage systems [C] //21st USENIX Conference on File and Storage Technologies (FAST 23)，2023：49 - 64.

[19] BAI A，CHEN M，PENG S，et al. Attention-based bidirectional LSTM with differential features for disk RUL prediction [C] //2022 IEEE 5th International Conference on Electronic Information and Communication Technology (ICEICT)，IEEE，2022：684 - 689.

[20] ANANTHARAMAN P，QIAO M，JADAV D. Large scale predictive analytics for hard disk remaining useful life estimation [C] //2018 IEEE International Congress on Big Data (BigData Congress)，IEEE，2018：251 - 254.

[21] GARGIULO F，DUELLMANN D，ARPAIA P，et al. Predicting hard disk failure by means of automatized labeling and machine learning approach [J]. Applied Sciences，2021，11 (18)：8293.

[22] ZHANG J，WANG J，HE L，et al. Layerwise perturbation-based adversarial training for hard drive health degree prediction [C] //2018 IEEE International Conference on Data Mining (ICDM)，IEEE，2018：1428 - 1433.

[23] CHHETRI T R，KURTEVA A，ADIGUN J G，et al. Knowledge graph based hard drive failure prediction [J]. Sensors，2022，22 (3)：985.