

# 视频图像中动态目标检测系统的 FPGA 硬件设计<sup>①</sup>

李艳婷, 王水鱼

(西安理工大学 自动化与信息工程学院, 西安 710048)

**摘要:** 为解决传统视频图像数据处理时, 通常以软件为基础, 存在工作量大、存储时间长的问题, 提出了基于 FPGA 硬件结构处理的设计, 提高系统的处理速度和实时性. 针对图像处理问题, 采用背景差分法检测运动目标, 通过二值化方法剔除图像中的静止场景, 然后再做形态滤波, 消除空洞和噪声点, 最后提取出目标中心点, 达到检测运动目标的目的. 同时加入 sobel 边缘检测模块, 扩展运动目标检测体系结构, 在检测目标时, 系统能够根据物体的形状以及它的运动路径, 来识别对象并对其检测结果进行分类. 此外, 对一些特定环境, 设置适当禁区, 当目标中心点进入禁区时, 做出相应判断. 硬件设计和 Verilog HDL 程序编写完成后, 在 Modelsim 中做仿真验证, 可以得到正确的目标检测结果.

**关键词:** 运动目标检测; 背景差分法; Sobel 边缘检测; 质心提取; 硬件设计; 事件分析

## FPGA Hardware Design of Dynamic Target Detection System in Video Image

LI Yan-Ting, WANG Shui-Yu

(School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China)

**Abstract:** Traditional video image data processing usually requires software, but there are some problems, such as heavy workload and long storage time. To solve this problem, this paper proposes a hardware process configuration based on FPGA, to improve the processing speed of the system and real-time. For the image processing problems, it uses the background subtraction to detect moving targets and remove the static scenes from images by Binarization, then does morphological filtering to eliminate voids and noise points, finally, extracts the target center to achieve the purpose of detecting moving targets. It joins sobel edge detection module at the same time, extends moving target detection architecture. According to the shape of the object and its path of movement, system can identify objects and categorize their test results when detects the target. In addition, for some specific environment, it sets the appropriate restricted area, when the center of the target is into the restricted area, makes the appropriate judgment. After the hardware design and Verilog HDL program is completed, doing simulation in Modelsim can get the right target detection results.

**Key words:** moving detection; background subtraction algorithm; Sobel edge detection; centroid extraction; hardware design; event analysis

## 1 引言

视频图像处理系统主要分为两大部分<sup>[1]</sup>, 第一是基于应用软件的处理技术, 该技术是在 PC 平台上用各种应用软件进行图片处理, 处理算法通常是由软件串行计算来完成, 这类技术对系统的实时性要求不高; 第二类是基于硬件结构的处理技术, 采用可编程逻辑器件 FPGA 来控制实时显示系统, 其性能优越<sup>[2]</sup>, 资源

丰富, 采用并行处理方式, 提高了处理速度, 具有高集成度、高可靠性、灵活的编程能力、全新的开发设计思想等优势, 在处理实时图像信号这样巨大的信息量时, 能满足系统对实时性的要求.

随着工业自动化水平和科学技术的不断发展和提高, 人们对视频数据的要求也越来越高, 高清晰、实时性视频数据量越来越大, 视频的实时处理难度也在逐

① 收稿时间:2016-06-10;收到修改稿时间:2016-07-19 [doi:10.15888/j.cnki.csa.005629]

渐增大, 因此利用 FPGA 来对实时图像进行硬件处理成为一个新的发展趋势<sup>[3]</sup>. 目前对于动态目标检测已经开展了大量研究, 主要是指通过对运动目标进行匹配和图像特征分析, 检测视频序列中是否存在与背景有相对运动的前景目标.

## 2 目标检测方法

运动目标检测是智能视频监控的底层处理过程, 它为目标分类、目标跟踪等其它后续处理步骤奠定了基础. 运动目标检测的方法主要有: 帧间差分法、光流法、背景差分法.

### (1) 帧间差分法

对视频图像序列中, 逐像素比较相邻两帧或三帧图像之间的差异<sup>[4]</sup>, 当监控场景中出现异常物体运动时, 相邻两帧之间会有明显差异, 帧间差分法就是利用这种特性对相邻两帧相减, 得到两帧图像亮度差的绝对值, 然后通过选定的阈值来分析视频图像的变化情况, 进而检测运动目标.

### (2) 光流法

给图像中的每一个像素点赋予一个速度矢量, 这就形成了一个图像运动场, 在运动的一个特定时刻, 图像上的点与三维物体上的点一一对应, 这种对应关系可由投影关系得到, 根据各个像素点的速度矢量特征, 可以对图像进行动态分析<sup>[4]</sup>. 如果图像中没有运动物体, 则光流矢量在整个图像区域是连续变化的. 当图像中有运动物体时, 目标和图像背景存在相对运动, 运动物体所形成的速度矢量必然和邻域背景速度矢量不同, 从而检测出运动物体及位置.

### (3) 背景差分法

背景差分法, 原理较为简单, 在背景已知的情况下, 利用当前帧图像的像素灰度值与背景图像的像素灰度值进行差分运算, 灰度值不同的像素点将会剩余下来, 然后设定阈值, 提取出不同于原始背景图像的标区域, 从而达到前景提取的目的.

运动目标的检测主要是为了得到感兴趣的运动目标的位置和大小. 经过研究可知<sup>[5]</sup>, 帧间差分法受光线变化影响小, 算法实现简单, 但检测时间的选择难以控制, 适用于多个运动目标和摄像机移动的场景, 一般不能完整的检测出运动目标, 实际应用在硬件平台上较为困难; 光流法的优点在于光流法不仅携带了运动物体的运动信息, 而且还携带了有关景物三维结

构的丰富信息, 它能够在不知道场景的任何信息情况下, 检测出运动对象, 但存在的问题是光流法采用迭代法计算求解, 算法复杂度高、计算量非常大, 计算耗时且实时性差, 除非有特殊的硬件支持, 否则很难实现运动目标的实时检测; 背景差分法是目前运动检测中最常用的一种方法, 与帧间差分法相比, 背景差分法可以获得完整的特征数据, 得到较为精确的目标图像, 且适用于静止背景. 考虑到本设计主要是针对一些特定区域, 如银行、ATM 机等场所, 该区域的背景是静态的, 且图像采集设备也是固定的, 因此背景差分法, 检测速度快而准确, 易于硬件实现, 可以满足实时性高的需求.

## 3 总体结构

图 1 表示, 将采集进来的视频图像数据, 缓存在 SDRAM 中, 在 FPGA 的控制下即可对其进行目标检测和图像处理, 处理完成后在 VGA 完成视频显示.

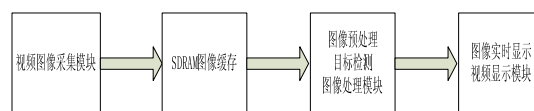


图 1 系统功能图

## 4 运动目标检测

基于硬件 FPGA 的运动目标检测和跟踪算法的实现主要是围绕着数据的读取控制来实现的, 背景差分法就是对保存的背景图像和当前帧图像做差分运算并二值化, 再经过形态学处理, 进而提取出运动目标并跟踪<sup>[6]</sup>. 图 2 为运动目标检测在 FPGA 中实现的体系结构图.

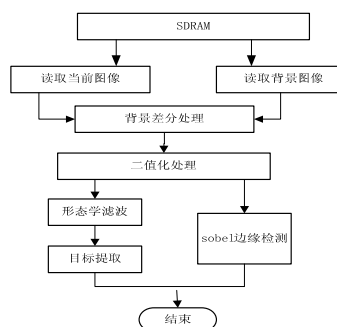


图 2 运动目标检测实现框图

### 4.1 背景差分运算

首先, 从 SDRAM 缓存中读取视频数据<sup>[7]</sup>, 两个输

入源分别是从 SDRAM 取出的视频流和采集保存的背景图像; 因此, 设计一个数据读取模块, 主要包括: 地址发生控制器和两个 FIFO; 一个用来缓存当前帧图像数据, 另一个用来缓存背景图像数据, 此处用 FIFO 缓存是为了解决时钟不匹配的问题。

地址发生控制器根据 FIFO 的空满状态, 不停地产生读取当前帧图像的地址和背景图像的地址, 从对应的地址中读取当前帧图像数据和背景图像数据, FIFO 的深度可取为两行视频图像的大小, 足够大能保证连续读取的同时也可以进行后面的差值运算, 即进行流水线处理<sup>[7]</sup>。当两个 FIFO 都不为空时, 就启动背景差值计算模块, 将 FIFO 中缓存的当前帧数据和背景图像数据做差值运算, 并根据给定的阈值进行二值化处理。最后将二值化结果送入形态学处理模块进行滤波去噪, 从而得到运动目标。在这种硬件处理结构下, 减少了数据的反复读写并减小数据处理量和存储空间, 提高系统的实时性, 减小占用硬件资源。图 3 为数据读取模块读取数据流程图。

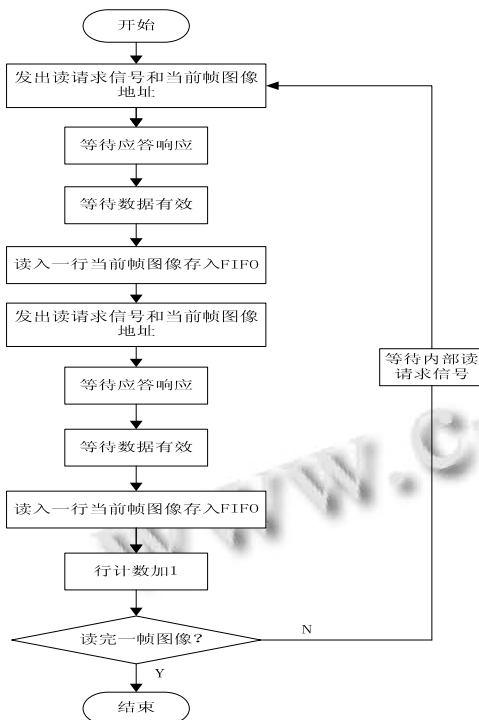


图 3 数据读取流程图

采用流水线结构即可实现背景差分计算并二值化处理。背景差分运算的实现过程<sup>[7]</sup>: 对背景图像和当前帧图像中相对应的像素灰度值进行相减并取绝对值, 当结果大于阈值 T 时, 则判断为运动目标像素点, 并

将该点灰度设为 1, 当结果小于阈值 T 时, 认为该点为背景像素, 将其值设为 0。这样就得到了二值图像。数据读取模块 RTL 视图如图 4 所示。

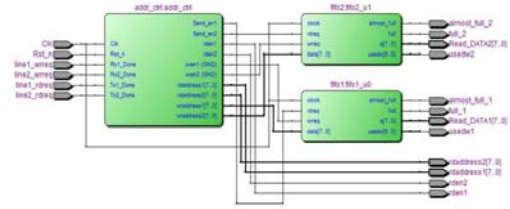


图 4 数据读取模块 RTL 视图

背景差分后的二值图像仿真图如图 5 所示, 可得到“0”、“1”像素点, 分别代表无运动目标和有运动目标。

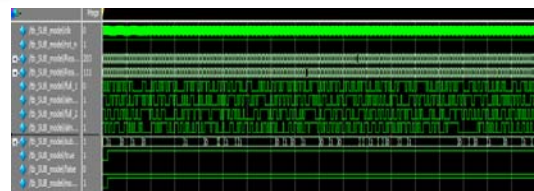


图 5 二值化处理图像

#### 4.2 形态学滤波

背景差分并二值化处理之后, 进入形态学滤波处理模块。形态学滤波处理主要分为开运算和闭运算。由文献[8]可知, 开运算可以较好地消除图像中的孤立点和噪声。

开运算是先利用腐蚀运算去除二值图像中孤立的点或者噪声, 然后通过膨胀运算对目标断裂部分进行修复, 得到的图像区域便是运动目标区域。

进行滤波处理时采用 3\*3 形态学滤波算子, 取算子如式(1), 值全为 1<sup>[9]</sup>, 在做腐蚀和膨胀处理时, 不影响像素处理结果:

$$S = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (1)$$

运算过程: 腐蚀运算即对像素做“与”运算, 膨胀运算对像素做“或”运算。用结构元素 S 依次对输入的 9 个二值图像像素进行扫描。腐蚀过程: 只要有一个像素的值是 0, 结果就为 0, 全为 1, 结果才为 1。膨胀过程: 只要有一个像素的值是 1, 结果为 1, 全为 0, 结果才为 0。

硬件实现过程中, 调用 Quartusii 提供的移位寄存

器 IP 核 altshift\_taps, 将单行的串行图像进行 3 线并行输出. altshift\_taps 的参数设置为 3 个 taps 输出, taps 间的间隔宽度为 640(VGA 显示像素为 640\*480), 数据宽度为 8 位(灰度图像). 图 6 为腐蚀运算的硬件结构图, 图 7 为膨胀运算的硬件结构图, 图 8 为开运算模块结构图(腐蚀运算的输出作为膨胀运算的输入).

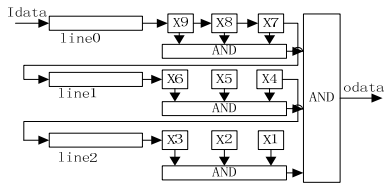


图 6 腐蚀运算硬件结构图

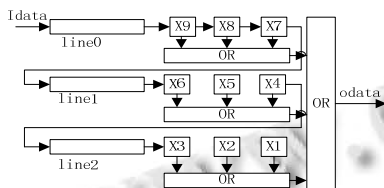


图 7 膨胀运算硬件结构图

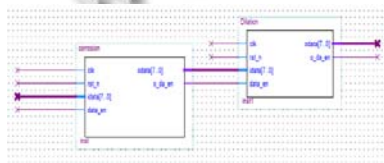


图 8 开运算模块结构图

### 4.3 运动目标提取

形态学滤波处理后, 得到的二值图像中像素值为“1”对应的区域即为运动目标区域. 考虑到运算量、系统实时性, 本设计将采用形心定位方法进行运动目标提取.

形心定位即选取运动目标区域的形心作为运动目标的特征. 对每一帧视频图像计算其形心, 就可以确定系列图像中目标连续的位置关系, 从而正确判断出后期运动目标是否进入禁区. 运动目标区域的形心提取公式<sup>[10]</sup>如式(2)、(3)所示:

$$X = \frac{\sum_{i=0}^N \sum_{j=0}^M f(x_i, y_j) \cdot x_i}{\sum_{i=0}^N \sum_{j=0}^M f(x_i, y_j)} \quad (2)$$

$$Y = \frac{\sum_{i=0}^N \sum_{j=0}^M f(x_i, y_j) \cdot y_i}{\sum_{i=0}^N \sum_{j=0}^M f(x_i, y_j)} \quad (3)$$

式(2)、(3)中,  $f(x,y)$ 为经形态学滤波处理后的二值

图像, 其值只可能取 0 或者 1, 设计时<sup>[10]</sup>用计数器 num\_counter 对  $f(x,y)$ 中值为“1”的像素个数进行计数, 用 sum\_X, sum\_Y 两个寄存器存储像素值为 1 的像素的坐标累加值, 用计数器 X\_counter 和 Y\_counter 表示当前像素的坐标, 当读取的值  $f(x,y)$  为 1 时, 进行如式(4)、(5)、(6)的计算:

$$num\_counter = num\_counter + 1 \quad (4)$$

$$sum\_X = sum\_X + X\_counter \quad (5)$$

$$sum\_Y = sum\_Y + Y\_counter \quad (6)$$

当整幅图像扫描完成后, 就得到当前图像的形心, 如式(7)、(8)所示.

$$oX\_center = \frac{sum\_X}{num\_counter} \quad (7)$$

$$oY\_center = \frac{sum\_Y}{num\_counter} \quad (8)$$

图 9 为运动目标形心提取流程图, 图 10 为运动目标形心提取模块仿真时序图, 在仿真时序图中可以看出该二值图像的形心被提取出来.

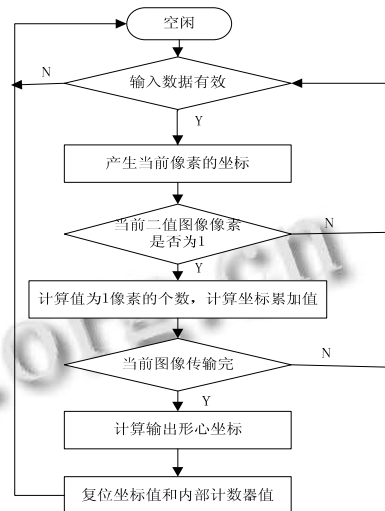


图 9 运动目标形心提取流程图

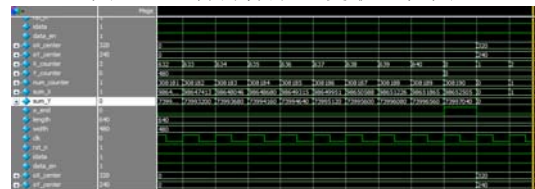


图 10 运动目标形心提取模块仿真时序图

### 4.4 Sobel 边缘检测

Sobel 边缘检测是一种较成熟的微分检测算法, 使用此算法对图像进行边缘检测, 能获得较好的边缘检测效果且计算简单. Sobel 算子原理框图如图 11 所示.

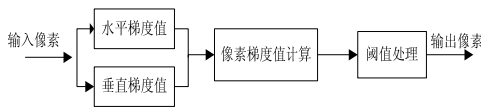


图 11 sobel 算子原理框图

实现过程: Sobel 算子在水平方向梯度矩阵 X 和垂直方向梯度矩阵 Y 分别如式(9)、(10)所示. 3\*3 的 9 个像素模板如式(11)所示<sup>[11]</sup>.

$$X = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (9)$$

$$Y = \begin{bmatrix} y_1 & y_2 & y_3 \\ y_4 & y_5 & y_6 \\ y_7 & y_8 & y_9 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (10)$$

$$F = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{bmatrix} \quad (11)$$

F 矩阵分别与 X 矩阵、Y 矩阵进行卷积运算, 得到水平梯度值  $G_x$  和垂直梯度值  $G_y$ , 如式(12)、(13)所示:

$$G_x = F \cdot X = (p_3 + 2p_6 + p_9) - (p_1 + 2p_4 + p_7) \quad (12)$$

$$G_y = F \cdot Y = (p_1 + 2p_2 + p_3) - (p_7 + 2p_8 + p_9) \quad (13)$$

再根据水平梯度值  $G_x$ , 和垂直梯度值  $G_y$ , 计算出像素的梯度值 G, 如式(13)所示:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (13)$$

最后根据式(14)与阈值 T 进行比较, 如果大于阈值, 便视其为边缘像素点.

$$g(x, y) = \begin{cases} 1 & G \geq T \\ 0 & G < T \end{cases} \quad (14)$$

基于 FPGA 的硬件实现 Sobel 边缘检测算法包括: 数据缓冲模块, 梯度计算模块, 加法模块, 开方模块, 阈值比较模块.

根据上述边缘检测 Sobel 算子的描述, 在 QuartusII 中设计了如下的 Sobel 边缘检测梯度运算结构图, 如图 12 所示.

数据缓冲模块调用移位寄存器 shift\_register, 实现 3 行图像数据的缓存, 参数设置和上面形态学处理中的参数一样; 梯度运算模块, 调用 6 个 ALTMULT\_ADD 乘法运算, 分别为 3 个水平梯度运算乘法器和 3 个垂直梯度运算乘法器; 加法模块, 调用 PARALLEL\_ADD 加法器对 3 行数据并行相加; 开方运算模块, 调用 ALTSQRT 即可计算出 |G| 的数值大小;

阈值比较模块, 设置阈值 T, 当计算出的 |G| 大于 T 时, 输出像素值为 1, 即为黑色, 反之则为 0, 即为白色, 以此检测出边缘像素. 图 13 为 Sobel 边缘检测 RTL 视图, 图 14 为 Sobel 边缘检测时序仿真图.

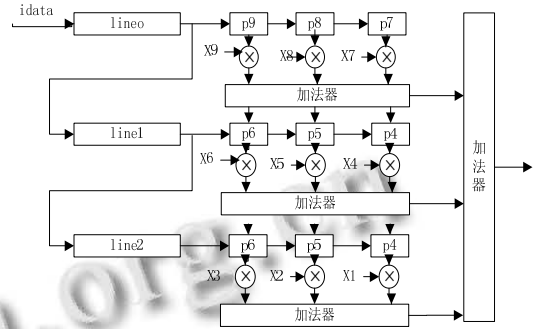


图 12 Sobel 边缘检测梯度运算结构图

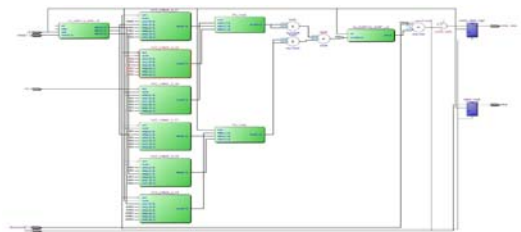


图 13 Sobel 边缘检测 RTL 视图



图 14 Sobel 边缘检测时序仿真图

### 5 禁区入侵判断

本文主要是针对特定区域, 如银行、ATM<sup>[12]</sup>机等设置禁区, 判断运动目标是否进入禁区, 以及停留时间过长时, 做出相应警报.

像素点组成了视频帧, 禁区<sup>[13]</sup>也是由一些被特殊规定的像素点组合而成的, 判断禁区中是否有目标进入, 也就是判断禁区点是否与目标中心点重合, 为了满足效率和实时性的要求, 只对禁区内有代表性的几个点进行比对, 假设禁区是一个矩形, 则只取它的左上角点  $(x_1, y_1)$  和右下角点  $(x_2, y_2)$  这两个点, 当目标进入矩形禁区时, 使目标中心点  $(x, y)$  与  $(x_1, y_1)$ 、 $(x_2, y_2)$  两点满足判断式(9)所示.

$$if(x \geq x_1 \& \& x \leq x_2 \& \& y \geq y_1 \& \& y \leq y_2) \quad (9)$$

同时为了减小误报警率, 当判断出目标进入禁区后要视频帧进行计数, 如果目标连续在禁区中的视

帧数超过了规定的值, 则进行报警.

本文的方法可以监控同一场景中的多个禁区, 只要事先设置好各个禁区的特殊位置即可, 解决了传统的区域入侵异常判断方法所存在的缺陷. 禁区入侵判断方法的流程如图 15 所示.

矩形禁区仿真时序图如图 16 所示, 此处设置预定帧数为 5 帧, 当目标中心点进入禁区并超过 5 帧时, 启动报警信号, 否则继续后期跟踪.

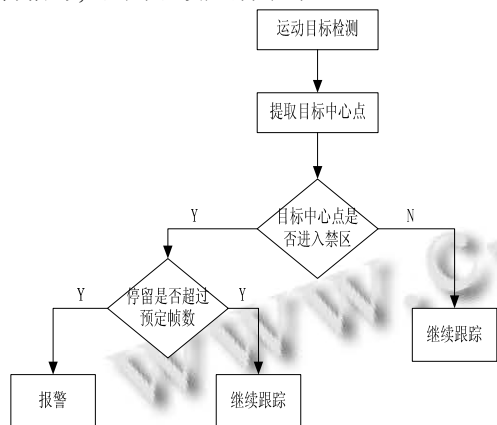


图 15 禁区入侵判断方法流程图

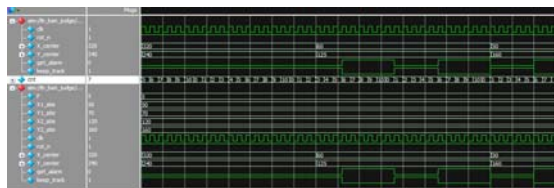


图 16 矩形禁区仿真时序图

## 6 实验结果分析

经过 FPGA 硬件设计后, 在 modelsim 中仿真试验并分析波形, 可以正确检测出运动目标, 其中每个模块的仿真测试数据均来自于 matlab 处理的一个 txt 文件, 这个文件主要是通过 verilog 中的一个系统函数 \$readmemh (“image\_txt”,instr) 将图像数据文件 image\_txt 读入到变量 instr 中, 然后编写 testbench 文件仿真, 仿真时再将相应的处理结果存储到 image\_process.txt 文件中<sup>[14]</sup>, 最后通过 matlab 的 imshow 函数实现图像数据的显示. 图 17 为各个方法的实验检测结果图.

从图 17 中可以清楚的看到背景差分法和 sobel 边缘检测法具有较好的检测结果, 能将运动目标的边缘轮廓及运动方向准确的检测出来.

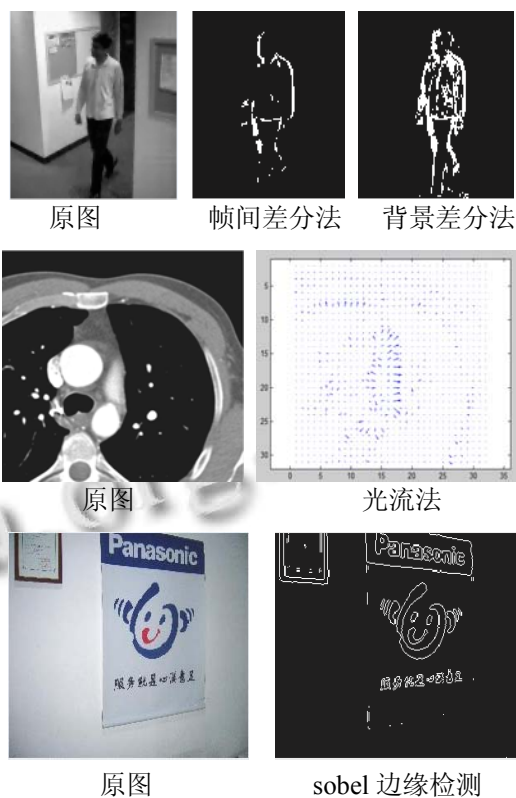


图 17 检测仿真结果对比

## 7 结语

本文针对软件操作处理大量图像数据时, 存在的一些处理速度慢及实时性差等问题, 设计了基于 FPGA 的硬件处理结构, 利用 FPGA 并行处理数据的优势, 能快速准确的检测到目标图像, 并对其进行分析, 为后期目标跟踪奠定了基础. 加入 Sobel 边缘检测算法, 优于传统方法, 可以根据目标形状和运动路径扩展可能的分类对象, 更好地用于监控系统中. 每个模块经过 Modelsim 仿真验证, 最终可以准确提取出运动目标, 对以后的目标跟踪具有实际的应用价值.

### 参考文献

- 1 杜文略,刘建梁,沈三民.基于 FPGA 的运动目标检测系统.电子技术应用,2014,40(11):36-38.
- 2 刘奇峰,曾庆立,章袞袞.基于 FPGA 的视频图像处理系统设计.信息系统工程,2010,10:1001-2362.
- 3 张欢,汪红,王芳芳.基于 FPGA 的实时图像处理实验平台设计.微型机与应用,2014,33(11):26-28.
- 4 武一,程铭,刘建明.边缘检测与帧差分法融合的视频分割算法.电子设计工程,2015,23(12):109-111.
- 5 Nayak S, Pujari SS. Moving object tracking application:

- FPGA and model based implementation using image processing algorithms. 2015 International Conference on Computing Communication Control and Automation (ICCUBEA). 2014. 932-936.
- 6 Gujrathi P, Priya RA, Malathi P. Detecting moving object using background subtraction algorithm in FPGA. 2014 Fourth, International Conference on Advance in Computing and Communication (ICACC). 2014. 117-120.
- 7 赵清华. 基于 FPGA 的目标跟踪算法的设计与实现[硕士学位论文]. 西安: 中国科学院西安光学精密机械研究所, 2014.
- 8 陈全金. 基于视频图像动态目标检测与跟踪算法的硬件实现研究[硕士学位论文]. 成都: 电子科技大学, 2012.
- 9 Mori JY, Llanos CH. Background subtraction algorithm for moving object detection in FPGA. 2012 VIII Southern Conference on Programmable Logic (SPL). 2012. 1-6.
- 10 彭丹阳. 基于 FPGA 的视频跟踪与编码系统设计与实现[硕士学位论文]. 沈阳: 东北大学, 2013.
- 11 马俊. 基于 FPGA 的实时图像采集与预处理系统研究[硕士学位论文]. 成都: 西南交通大学, 2015.
- 12 苟超. ATM 场所嵌入式智能视频监控系统设计[硕士学位论文]. 西安: 西安电子科技大学, 2014.
- 13 张新新, 孔英会. 智能视频监控中的异常检测方法研究. 科技论坛, 2009: 38-39.
- 14 廖传柱. 一种基于 FPGA 的 Sobel 边缘检测算法与仿真研究. 长江大学学报(自然版), 2015, 12(7): 37-40.