

自适应阈值图像二值化及形态学处理的 FPGA 实现*

陈鑫元¹ 李筠¹ 杨海马^{1,2} 宋佳¹ 邵璐先¹

(1. 上海理工大学光电信息与计算机工程学院 上海 200093; 2. 中国科学院上海技术物理研究所 上海 200083)

摘要: 针对固定阈值法图像二值化不足以及二值化后图像存在严重干扰,传统 DSP 等处理器在高速图像处理中实时性差,提出了以 FPGA/Nios II 为处理器,运用改进的自适应阈值 Sobel 算法实现对图像的二值化,再进行形态学的处理。整个系统实现包括原始图像的采集、色彩格式转换、灰度处理、边缘检测以及运用中值滤波和加权平均法实现自适应阈值图像二值化,最后再运用形态学方法对二值化后图像做进一步处理,实现轮廓的更加有效的提取,并通过 VGA 接口实现显示。实验结果表明该系统检测效率高,速度快,稳定可靠。

关键词: 二值化;FPGA/Nios II;自适应阈值;形态学;边缘检测;中值滤波和加权平均

中图分类号: TP391.4 **文献标识码:** A **国家标准学科分类代码:** 510.4050

Adaptive threshold binarization and morphological image processing based on FPGA

Chen Xinyuan¹ Li Jun¹ Yang Haima^{1,2} Song Jia¹ Shao Luxian¹

(1. School of Optical-Electrical and computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; 2. Shanghai Institute of Technical Physics of Chinese Academy of Sciences, Shanghai 200083, China)

Abstract: For reducing a fixed threshold binarization method's drawback and the serious Inadequate and Interference after binarization handling, meanwhile the traditional DSP and other processors in high-speed image processing in real-time poor is not good enough, proposing that using FPGA / Nios II processor and the improved adaptive threshold Sobel algorithm for image binarization, morphological processing again. The whole system implementation includes the original image capture, color format conversion, gradation processing, edge detection and median filtering using the weighted average method and adaptive threshold image binarization, and finally by using morphological methods to further deal with the binary image, to achieve a more efficient extraction of contour, and through VGA interface display. Experimental results show that the system is high efficient, fast, stable and reliable.

Keywords: binarization; FPGA/Nios II; adaptive threshold; morphology; edge detection; median filtering and weighted average

1 引言

图像边缘检测及其二值化是图像特征提取的重要研究内容之一,如何很好地对图像边缘进行有效提取,一直是国内外学者研究的课题。虽然目前针对自适应阈值图像二值化及其后续处理的算法很多,但是受到算法复杂度、实时性和处理器性能的限制,很多还只是理论上的实现而已。可编程逻辑器件的快速发展使得运用逻辑性操作极强的门阵列来实现数字图像的高速处理成为了可能^[1],由于没有指

令集系统,与 DSP、ARM 等传统图像处理器相比处理速度快了很多。基于此,提出了以 FPGA 为主控芯片,结合 Sopc 片上系统 Nios II 软核实现数字图像处理功能。CMOS 摄像头采集到的原始图像经过 YUV 到 RGB 格式转换后进行灰度处理^[2],接着通过 Sobel 算子对图像边缘进行提取^[3],对提取后的边缘运用中值滤波和加权平均法实现自适应阈值二值化处理,代替传统的固定阈值法^[4]。由于二值化后的图像中可能存在一些空洞和断点,采用数学形态学进行进一步处理^[5],提取更理想的图像轮廓。

收稿日期:2015-12

* 基金项目:国家自然科学基金青年基金(61302181)、上海市闵行区产学研合作计划(3714302006)、上海市教委创新项目(13YZ111)资助

2 系统结构及工作原理

系统采用 Altera 公司的 Cyclone III 系列 FPGA 芯片, 型号为 EP3C40F484C6, 速度等级为 6, 使用 27 M 和 50 M 外部晶振提供时钟源, 其中 27 M 晶振提供给 CMOS 摄像头和 VGA 接口, 50 M 晶振经过 PLL 锁相环后输出 100 M 时钟控制 SDRAM 的读写, 采用双 SDRAM 结构, 分别用来存储读取 CMOS 摄像头采集的图像数据和供 Nios II 软核使用, Nios II 软核用来配置一些底层驱动程序(如摄像头寄存器配置, I²C 总线协议编写, 细化算法的实现)。摄像头采用 MT9V034, 分辨率 752×480。VGA 显示采用 640×480 的分辨率。系统总体结构如图 1 所示。

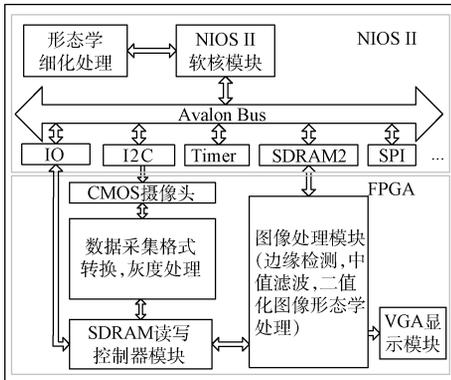


图 1 系统总体结构

3 算法原理

3.1 自适应阈值 Sobel 边缘检测算法原理

图像边缘是指在图像周围像素灰度急剧变化的那些像素的集合, 它是图像最基本的特征。Sobel 算子是运用一阶微分算子检测图像边缘, 一阶微分边缘算子(梯度边缘算子)是利用图像在边缘处阶跃性进行边缘检测的。在实际算法实现过程中, 通过 3×3 模板作为核与图像中被选中的相同大小区域的每个像素点做卷积运算, 计算出选中图像在水平方向和垂直方向上的梯度值, 通过相应的公式将图像每个像素点的横向和纵向的灰度值结合得到该点的在一个方向上的梯度值。由于经典 Sobel 算法对噪声比较敏感, 且需要人为指定阈值, 常常会造成边缘点的误判。为了解决这一问题, 有很多自适应阈值选取方法, 比如直方图法。不过直方图法需要图像经过 Sobel 算法边缘检测后的灰度直方图呈现明显的双峰值, 通过选取两峰之间的谷底对应的灰度级作为阈值来实现二值化, 因此极大地限制了直方图法实现自适应阈值的选取。基于此, 在图像经过卷积梯度值计算后, 采用中值滤波和加权平均法实现自适应阈值的计算实现图像的二值化。取 a_1, a_2, \dots, a_9 为灰度图像中所选取的 3×3 区域, 利用 Sobel 算子对其作梯度计算, 可以得到中心像素点 P_5 在水平、垂直方向的梯

度 G_x 和 G_y 计算后结果如式(1)和式(2)所示。

$$G_x = (a_3 + 2a_6 + a_9) - (a_1 + 2a_4 + a_7) \quad (1)$$

$$G_y = (a_1 + 2a_2 + a_3) - (a_7 + 2a_8 + a_9) \quad (2)$$

像素点 P_5 处的梯度值 $|G|$ 可根据式(3)计算得出。

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (3)$$

中值滤波按维数可分为一维中值滤波和二维中值滤波, 其本质是指运用一个奇数点的移动窗口, 将窗口中心点的值用窗口内各点的中值代替。一维中值滤波数学表示如式(4), 二维中值滤波数学表示如式(5)所示。式中 A 为所选窗口区域, $\{f_{ij}\}$ 为二维数据序列。加权平均法是指根据每个元素在总体中的重要性乘上相应的权值, 结果相加作为总体的平均值, 权值之和为 1, 数学表达式如式(6)所示, p_1, p_2, \dots, p_n 为总体 n 中各元素对应的权值。整个系统自适应阈值提取过程如图 2 所示。

$$Y_i = Med\{f_{i-v}, \dots, f_i, \dots, f_{i+v}\}, i \in N, v = \frac{m-1}{2} \quad (4)$$

$$Y_{ij} = Med\{f_{ij}\} \quad (5)$$

$$\bar{X} = x_1 p_1 + x_2 p_2 + \dots + x_n p_n, p_1 + p_2 + \dots + p_n = 1 \quad (6)$$

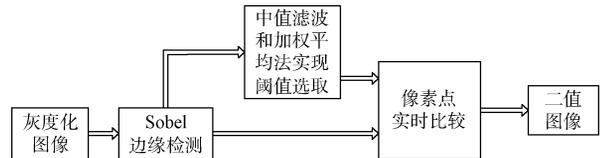


图 2 自适应二值化过程

由于经过 Sobel 算子梯度化后对应的是加权后的中心像元的灰度, 边缘处的和非边缘处的像素点灰度值有着比较大的差异, 传统的办法是通过尝试设定一个固定阈值实现二值化, 这样处理虽然简单却会严重受到外在因素影响, 为了实现阈值自适应调整, 在经过卷积运算后的图像中选取 5×5 区域, 运用中值滤波求出区域中间值、次最小值、最小值, 由于中值滤波的目的是将区域中心像素点用所选取窗口像素点的中值代替, 故对于二值化阈值的选取更关注的是区域中值、次最小值和最小值像素点。将这 3 个值进行加权平均, 并将此结果作为阈值, 权值的选择随着离中心像素点的距离的变大而减小。

3.2 数学形态学原理

数学形态学是一种采用集合论方法定量描述目标几何结构的学科, 这种思想对图像处理产生了很大的影响, 可以很好地对图像中存在的一些空洞或断点进行处理, 得到更加本质的图像形态^[6]。形态学主要包括膨胀、腐蚀、细化、粗化、开运算、闭运算。

膨胀是将目标区域的背景点合并到该目标物中, 使目标物边界向外部扩张的处理^[7]。选取结构元素 B , 将图像 X 中的每一个像素点扩大为 $B+x$, 其数学表达式如式(7)

所示。膨胀可以使图像断点连合,成为一个整体。

$$X \oplus B = \{x \mid B + x \cup x \neq \varphi\} \quad (7)$$

腐蚀是将目标区域的边界收缩,直观上就是使得目标区域变小,与膨胀是互为对偶的过程,但是并非经过两种运算后图像可以还原。选取结构元素 B ,将图像 X 的每一与结构元素 B 全等的子集 $B + x$ 收缩为点 x ,其数学模型如式(8)所示。腐蚀可以消除边界处的干扰点。

$$X \ominus B = \{x \mid B + x \subset X\} \quad (8)$$

开运算和闭运算是为了解决膨胀和腐蚀运算改变图像大小而提出来的。开运算是先腐蚀后膨胀的过程,可以消除图像中细小噪声,平滑物体边界。闭运算是先膨胀后腐蚀的过程,可以消除图像中存在的空洞,平滑物体边界。

细化是指在不破坏图像连通性的前提下,细化图像,使得在经过二值化后的轮廓更加细致,便于进一步做检测。细化效果的评定标准主要有以下几个方面。

- 1)不破坏图像的连通性;
- 2)算法复杂度低;
- 3)保持图像的基本形状;
- 4)收敛性;
- 5)细化后的线段尽量为原始线段的中心线。

二值图像的细化有很多的算法可以实现,比较经典的有 Hildith, Rosenfeld, Pavlidis, Zhang 并行快速细化算法等^[8]。这几种算法各有特点,Hildith 是一种比较普遍的算法,适用性比较强;Rosenfeld 采用的是并行和串行混合处理的细化算法,通过位运算实现特定模式的匹配;Pavlidis 采用并行细化算法,串行处理和并行处理均是迭代算法的常用方法,串行方式不仅取决于前次迭代结果还取决于本次迭代处理中已经处理过的像素点分布,而并行方式仅取决于前次迭代结果,显然这两种方式算法复杂度较大,图像的处理效果要比 Hildith 算法要好,但适用范围不如 Hildith 算法^[9]。Zhang 并行快速细化算法相比前 3 种算法,算法的检测效率更高,是一种常用的方法,其原理是选取 3×3 区域,通过约束条件,对其中存在的需要删除的前景点进行标定,待一次扫描结束后同时删除。系统采用 Zhang 并行快速细化算法实现细化。

粗化与细化是一对逆运算,在本文中只涉及二值图像的细化,不进行粗化过程。

4 模块的实现

系统的实现包含多个模块的实现,主要有色彩格式转换、灰度转换、I²C 驱动模块实现、VGA 显示模块实现、改进的自适应阈值 Sobel 边缘检测模块实现以及形态学处理模块实现^[10-11],限于篇幅,此处只介绍后两种模块实现方式。

4.1 自适应阈值 Sobel 边缘检测算法硬件实现

采用 FPGA 的 FIFO 模块实现对外部 SDRAM 存储芯片的读写控制,FIFO 模块读写采用异步时钟控制方式

和乒乓操作。乒乓操作的结构如图 3 所示。在第 1 个缓存时间周期内,将输入的第 1 个数据流缓存到 FIFO1 模块中,在第 2 个缓存周期中,输入的第 2 个数据流缓存到 FIFO2 中的同时 FIFO1 的数据作为输出,在第 3 个缓存时间内,输入的第 3 个数据流缓存到 FIFO1 的同时 FIFO2 的数据作为输出。如此不断循环。其优点可以实现数据的无缝缓存与处理,节约了缓存区空间,达到高速数据流处理效果。乒乓操作仿真结果如图 4 所示。

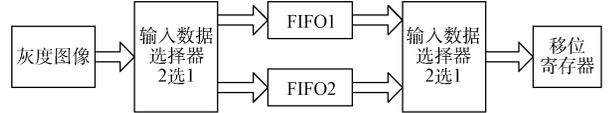


图 3 数据流乒乓操作

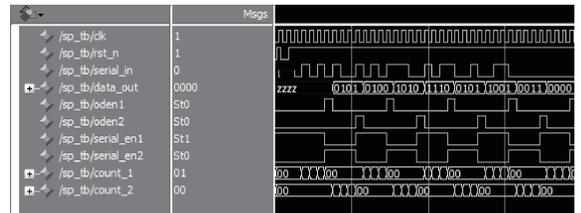


图 4 Pingpong 操作仿真波形

灰度化后的图像经过 FIFO 写操作存于 SDRAM 芯片中,通过 FIFO 读操作实现对像素点的读取,这样就可以实现对图像的缓存,解决前后级之间的速度不匹配问题。对读取的像素点使用移位寄存器,硬件加法和乘法器实现 Sobel 算法边缘检测。配置移位寄存器输入输出的数据位宽为 8,taps 位宽 640,总的 taps 数目为 3。通过移位寄存器可以实现对像素点邻近 3 行的并行处理,其工作原理如图 5 所示。

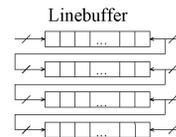


图 5 移位寄存器结构

每个 clk 输入 1 个待移位数据,同时输出 1 个数据,而在存储内部每个 clk 周期移位 1 次,每个 tap 的输出直接移位到下一级 tap 的输入。移位寄存器作用很大,下面模块算法的实现均运用到了此寄存器。仿真波形如图 6 所示。

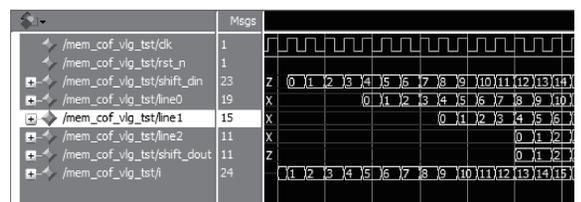


图 6 移位寄存器仿真波形

从仿真结果可以看出,根据摄像头分辨率配置合适的移位寄存器,使得在经过移位寄存器移位操作后,实现 3 行像素点的同时读取,通过硬件乘法器,加法器和整数平方根实现 Sobel 边缘检测算法,由于硬件实现是并行处理,且延时很短,极大提高了卷积运算的实现。其实现过程如图 7 所示。

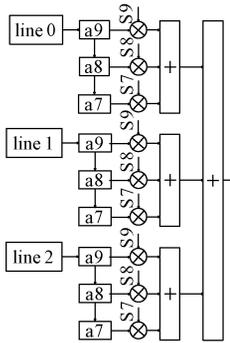


图 7 Sobel 算子硬件实现

中值滤波依然采用移位寄存器实现,选取总的 taps 数量为 5,实现 5×5 像素区域的选择,相比于 3×3 区域,首先对图像的边缘平滑效果会更好,其次由于区域选择越大,可供选择的范围也会更大,这样得到的阈值会更加合适。对于加权平均权值的选择由中心像素点向外选为 0.4,0.3,0.3,调试中可以根据实际情况进行选取。中值滤波算法的实现采用分段法求解,先对第一行进行冒泡排序,求出中间值,其它行均采用同样方式,由于这 5 行数据是并行处理的,5 行数据的读取和处理是同时进行的,加快了算法的实现,体现出了 FPGA 在并行处理中所占的优势。对求出的 5 个中间值,次最小值,最小值再进行一次冒泡排序,求出最终区域内的中间值,次最小值和最小值,以这 3 个值的加权平均值的微调结果作为自适应阈值。算法实现流程如图 8 所示。采用该算法与传统的大津法获取自适应阈值相比算法的复杂度有了明显的提高,同时由于算法中处理的点均是实时变化的点,其动态适应性性能也有了很大提高。

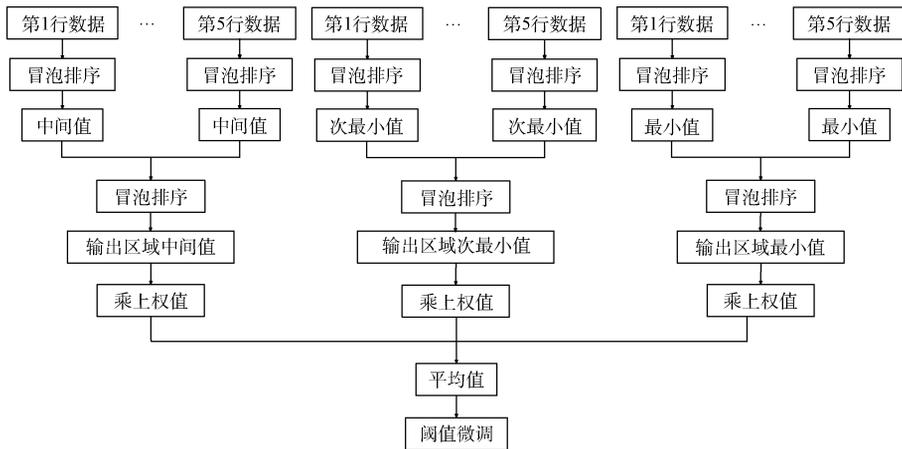


图 8 自适应阈值算法流程

4.2 形态学处理实现

形态学的实现在于结构元素的选择,系统首先采用 3×3 的结构元素对像素点进行逐点膨胀,接着对像素点再进行一次腐蚀操作。像素点的获取依然采用移位寄存器得到,通过非阻塞赋值从相邻的 3 行像素点中分别提取 3 个元素,形成 3×3 的区域,对这 9 个元素通过或运算实现膨胀,与运算实现腐蚀,如果效果不明显可以进行多次闭运算,达到消除图像中存在的空洞和断点的目的。

系统以 Zhang 并行快速细化算法实现对经过反复闭运算后的二值化图像细化,闭运算后的二值化图像经过 FPGA 中外扩 SDRAM 芯片读写 FIFO 控制模块实现与 FPGA 的 Sopc 片上系统 Nios II 软核中的寄存器接口对接。通过调用 IO 口读函数实现对此寄存器中的数据进行实时读取并存入二维数组中,对二维数组同时进行 Zhang

并行快速细化算法实现。采用 Nios II 软核的目的在于 Nios II 与传统的处理器相比,具有强大的定制性,用户可以根据自己的需求在 Sopc builder 中选取合适的组件和进行相关配置组成高性能的系统,避免了选取其它处理器作为外扩,同时由于采用 C 语言编程,方便了算法的实现。

设开辟的二维数组空间为 `imagebuf[640][480]`,每点对应的坐标用 $H_x[640], H_y[480]$ 表示。每次按顺序选取其中 3 行 3 列像元 P_1, P_2, \dots, P_9 构成 3×3 区域,作为算法实现模板,如图 9 所示。

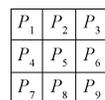


图 9 Zhang 并行快速细化算法模板

以 P_5 为中心像素点,按两步约束法实现,前景点值为 0,背景点值为 1。第一步约束条件为:

- 1) P_5 为前景色(值为 0);
- 2) 除去中心元素,以 $P_2, P_1, P_4, P_7, P_8, P_9, P_6, P_3$ 为序时,这些点的值由 1→0 变化的次数为 1;
- 3) $P_2 \parallel P_6 \parallel P_8 = 1$;
- 4) $P_4 \parallel P_6 \parallel P_8 = 1$;
- 5) P_5 的 8 邻域点中非 1 个数 ≥ 2 且 ≤ 6 。

标记所要删除的点,待所有边界点扫描结束进行删除。第二步约束条件为:

- 1) P_5 为前景色;
- 2) 除去中心元素,以 $P_2, P_1, P_4, P_7, P_8, P_9, P_6, P_3$ 为序时,这些点的值由 1→0 变化的次数为 1;
- 3) $P_2 \parallel P_4 \parallel P_8 = 1$;
- 4) $P_2 \parallel P_4 \parallel P_6 = 1$;
- 5) P_5 的 8 邻域点中非 1 个数 ≥ 2 且 ≤ 6 。

为了提高算法的执行效率,通过采用 Altera 公司提供的 C-to-Hardware(C2H)加速编译器,其专为 Nios II 设计,目的就是使得用户可以直接将关键性能的 C 语言转换为 FPGA 中的快速硬件加速器并使其链接到 Sopc builder,并自动重新生成系统,并调用 Quartus II 对系统进行重新编译、硬件设计、生成 FPGA 的配置文件。采用 C2H 可使系统整体性能提高一个数量级以上。

5 实验结果

根据系统的整体结构搭建实验平台,设置相关模块配置,编写图像采集,色彩格式转换,图像灰度化,Sobel 算子实现,形态学处理,VGA 显示等程序,通过不同的图像处理方式得到图 10(a)灰度化图像,图 10(b)没有经过多次闭运算形态学处理的二值化结果,图 10(c)经过多次闭运算操作未经过细化的二值化图像,图 10(d)经过细化后的最终二值化图像结果。从图 10(b)图像处理结果来看,运用中值滤波和加权平均算法实现的自适应阈值 Sobel 边缘检

测存在许多空洞和断点,不过相比与固定阈值对环境因素适应性有了很大提高,在经过多次闭运算后可以有效地解决这个问题,得到更为真实的图像轮廓,再经过细化处理可以得到如图 10(d)所示的效果图。

6 结 论

本文介绍了以 FPGA/Nios II 为处理器,运用 Verilog HDL 硬件描述语言和可编程宏功能块结合 C 语言对图像边缘进行检测,并通过改进的自适应阈值算法实现图像的二值化,增强了对环境的适应性,最终运用形态学对二值化后的图像做进一步处理,消除存在的空洞,断点,细化轮廓骨架,对噪声起到了很好地抑制,系统总共消耗逻辑资源 16%,运用 Sopc 片上系统方便了开发,同时避免了处理器的外扩。实验结果表明,本系统可以实现对图像的轮廓有效提取,同时检测效率与 DSP 相比也提高了许多。

参考文献

- [1] 许芳,席毅,陈虹,等.基于 FPGA/Nios-II 的矩阵运算硬件加速器设计[J].电子测量与仪器学报,2011,25(4):377-383.
- [2] 孙辉. Bayer 彩色序列图像的电子稳像[J].光学精密工程,2014,22(2):481-488.
- [3] 杨新华,寇为刚.基于 FPGA 的 Sobel 算子图像边缘检测算法[J].仪表技术与传感器,2013(1):102-104.
- [4] 曹杨,苏丽娜,沈琪,等.一种改进的 Sobel 边缘检测算法的设计及其 FPGA 实现[J].微电子学与计算机,2012,29(10):124-127,132.
- [5] 武英.形态学图像处理的应用[J].计算机与现代化,2013(5):90-94.
- [6] 刘紫燕,祁佳.实时图像边缘检测形态学优化设计及 FPGA 实现[J].计算机技术与应用,2013,39(9):132-134,138.
- [7] 师文,朱学芳,朱光.基于形态学的 MRI 图像自适应边缘检测算法[J].仪器仪表学报,2013,34(2):408-414.
- [8] 李杰,彭月英,元昌安,等.基于数学形态学细化算法的图像边缘细化[J].计算机应用,2010(5):514-516,520.
- [9] CHATBRI H, KAMEYAMA K. Using scale space filtering to make thinning algorithms robust against noise in sketch images [J]. Pattern Recognition Letters,2014,42(1):1-10.
- [10] 李易难,牛燕雄,杨露.基于 DSP+FPGA 视频图像采集处理系统的设计[J].电子测量技术,2014,37(1):58-61.
- [11] 潘明,陈元枝,李强.基于 FPGA 的图像采集系统的设计[J].国外电子测量技术,2012,31(3):58-61.

作者简介

陈鑫元,1991 年出生,硕士研究生,主要研究方向为数字图像处理、嵌入式开发设计。

E-mail:709416504@qq.com

李筠,1975 年出生,副教授,主要研究方向信号分析与处理,自动化控制与数字图像处理等。

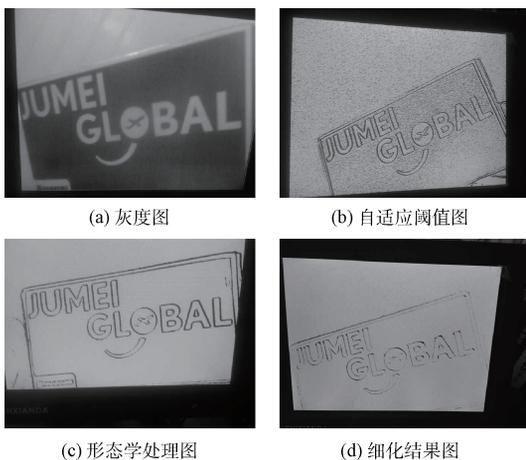


图 10 系统测试结果