

基于 FPGA 的图像锐化实现

刘 智, 夏春蕾, 戴曙光

(上海理工大学 光电信息与计算机工程学院, 上海 200093)

摘 要: 简要介绍了图像锐化中的 Laplacian 算法和 FPGA 实现图像处理的优点, 然后给出了基于 FPGA 的 Laplacian 算法实现方案, 并且利用 Altera 公司提供的 IP 核实现了算法。对关键部分进行了仿真分析, 将程序下载到开发板上进行验证, 从结果可以看出, 本算法很好地满足了实时性和增强图像的要求。

关键词: FPGA; 图像锐化; Laplacian 算法

中图分类号: TP274

文献标识码: A

文章编号: 1674-7720(2012)03-0040-02

Image sharpening based on FPGA

Liu Zhi, Xia Chunlei, Dai Shuguang

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: The paper briefly introduces the Laplacian algorithms in image sharpening and the advantages of using FPGA to process image at first. Then it gives the program based on Laplacian algorithm and realizes the algorithm using IP core provided by Altera Company. This paper also analyses the key parts of the design. At last, the paper programmes the design and downloads to the development-board. The results prove that the design satisfies the real-time requirements well.

Key words: FPGA; image sharpening; Laplacian algorithm

随着计算机技术的快速发展, 图像处理也出现了前所未有的高速发展。目前常用的图像处理工具是 MATLAB 提供的图像处理工具箱, 它提供了一套全方位的参照标准算法和图像工具, 用于对图像进行处理、分析、可视化和算法开发。一方面, 用它增强了图像的效果, 获得了清晰度更高的图像; 但另一方面, 伴随着信息复杂程度的提高以及信息量的增大, 这种方法由于实时性不高往往会产生滞后和失真现象。

采用 FPGA 来进行图像处理的优势主要是速度快、应用灵活。速度快的优势来源于 FPGA 的硬件逻辑, 特别是 FPGA 的并行执行和流水线操作。FPGA 的逻辑功能全部用硬件电路实现, 因此所有的延迟只来源于门电路, 而一般门电路的延迟都在纳秒级别, 因此用 FPGA 来对图像实施处理, 可以弥补上面所提到的缺憾, 很好地达到实时性和增加图像的双重目的。

1 图像锐化

图像锐化是图像增强的主要应用方面, 图像锐化处理的目的是增强图像边缘, 使目标物体的边缘鲜明, 以

便于提取目标物体的边界、对图像进行分割、目标区域识别以及区域性状提取等, 为图像理解和分析打下基础。而图像边缘模糊的实质是平均或积分运算造成的, 因此可以对图像进行逆运算来使图像清晰化, 达到锐化的目的。

图像锐化有梯度锐化和 Laplacian 锐化两种方法, 常用的图像锐化处理方法是基于 Laplacian 算子的模板运算。模板操作实现了一种邻域运算, 即某个像素点的结果不仅和本像素的灰度值有关, 而且和其邻域点的值有关。模板运算在数学中的描述是卷积(或互相关)运算。

设 ∇^2 为 Laplacian 算子, 则有:

$$\nabla^2 = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(i+1, j) + f(i-1, j) + f(i+1, j) + f(i-1, j) - 4f(i, j) = M1 \times X1 + M2 \times X2 + M3 \times X3 + M4 \times X4 + M5 \times X5 + M6 \times X6 + M7 \times X7 + M8 \times X8 + M9 \times X9 \quad (1)$$

$$\text{其中, } \begin{bmatrix} X1 & X2 & X3 \\ X4 & X5 & X6 \\ X7 & X8 & X9 \end{bmatrix} = \begin{bmatrix} f(i-1, j+1) & f(i, j+1) & f(i+1, j+1) \\ f(i-1, j) & f(i, j) & f(i+1, j) \\ f(i-1, j-1) & f(i, j-1) & f(i+1, j-1) \end{bmatrix} \quad (2)$$

《微型机与应用》2012 年第 31 卷第 3 期

$$\begin{pmatrix} M1 & M2 & M3 \\ M4 & M5 & M6 \\ M7 & M8 & M9 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad (3)$$

式(2)为输入图像矩阵,其值为灰度值大小。式(3)即为 Laplacian 模板,此外还有下面两个模板:

$$\begin{pmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{pmatrix} \quad (4)$$

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (5)$$

比较 3 个模板可以得出:式(3)模板只能检测水平和垂直方向的边缘,式(4)模板只能检测两对角方向的边缘,而式(5)模板检测水平、垂直和两对角方向的边缘。

2 图像锐化的 FPGA 实现

根据整体方案的设计思路,系统的硬件构架框图如图 1 所示。图像数据在 FPGA 的控制下通过 PC 总线由 CMOS 图像传感器获得,得到的图像数据在 SDRAM controller 的控制下存储在外部双端口 SDRAM 中,然后在 Image Sharpening 中对得到的数据进行锐化处理,最后按照 VGA 时序标准送入 VGA 显示。

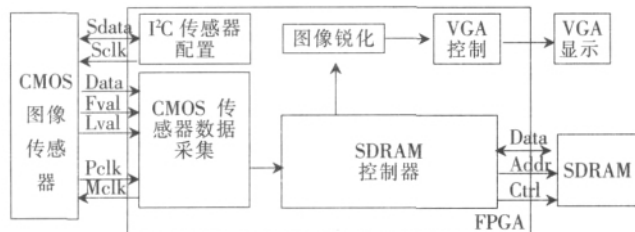


图 1 图像处理的构架

其中图像锐化是本文研究的重点,它主要由数据缓冲和模板计算两部分构成,如图 2 所示。



图 2 图像锐化的实现

2.1 数据缓冲设计

由前面的分析可以看出,设计的关键在于矩阵(2)的硬件实现。其在 MATLAB 中很容易实现,但是在硬件语言中,由于没有矩阵的概念,处理起来比较麻烦。因为所做的图像增强中用到的基本为 3×3 矩阵,所以本文讨论了在 FPGA 中怎样按行来实现和处理 3×3 的矩阵。

本设计主要采用了 Altera 公司的移位寄存器(RAM-based) IP 核,其示意图如图 3 所示。前期采集到的图像数据通过 shiftin 通道按行进入移位寄存器(RAM-based),经 IP 核后有 3 个输出端口:taps0x、taps1x、taps2x。通过图 4 所示的仿真图可以看出,当最后一个像素 X_9 输入后,在下一个时钟周期内得到的是像素 X_1 、 X_4 、 X_7 ,再下一个时钟周期是 X_2 、 X_5 、 X_8 ,最后一个时钟周期是 X_3 、 X_6 、 X_9 ,实现了期望的结果。

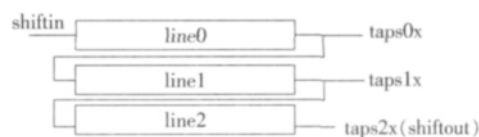


图 3 移位寄存器(RAM-based)

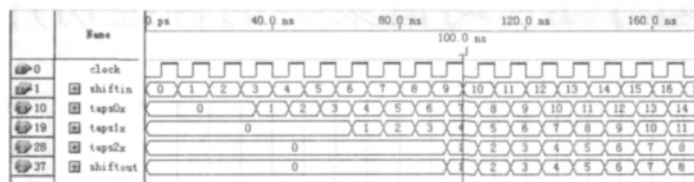


图 4 移位寄存器(RAM-based)仿真图

2.2 模板计算

模板计算主要完成式(1)的计算,该模块主要由 ALTMULT_ADD 和 PAPALLEL_ADD 及 LPM_ABS 3 个 IP 核来实现。ALTMULT_ADD 实现的是乘积累加和功能,其示意图如图 5 所示,计算公式为 $mac_2 = dataa_0 \times datab_0 + dataa_1 \times datab_1 + dataa_2 \times datab_2 = M9 \times X9 + M8 \times X8 + M7 \times X7$,其中 dataa_1 与 dataa_2 分别为 dataa_0 延时 1 个时钟周期和 2 个时钟周期得到的。PAPALLEL_ADD 主要实现并行加法的功能,完成了流水线操作,缩短了处理时间。

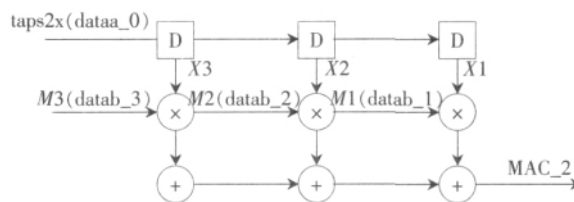


图 5 ALTMULT_ADD

为了防止出现灰度值为负和溢出的情况,在后期数据处理中加入了求绝对值的设计,用 LPM_ABS 这个 IP 核来实现。当得到的数据大于 255 时,最后得到的结果取 255;小于 255 时,即为输出结果。

3 FPGA 芯片选择

Altera 公司的 FPGA 开发工具 Quartus II 开发环境友好、易操作且自带有大量免费的 IP 核供设计者使用,从而减少了用户的大量工作,便于设计者模块化开发,特别是在完成复杂系统设计的时候,这些宏功能模块无疑将大大减少设计风险及缩短开发周期。

一般地,输入图像的格式大于 1280×1024 (60 Hz),这样的分辨率要求图像的数据传送带宽在 100 Mpix/s 以上,但是一般晶振提供的时钟频率为 50 MHz,满足不了上述要求,因此需要 FPGA 内部带有锁相环。另外,考虑到系统占用的面积和图像处理需要乘法器的特点,要求 FPGA 的内部资源尽量丰富。为此,该系统最终选用了 Altera 公司 Cyclone II 系列的 EP2C8Q208C8。该芯片最多用户 I/O 管脚数为 182 个,锁相环为 2 个,逻辑资源为 8 256 个,存储器资源为 165 888 bit,嵌入式乘法器

(下转第 45 页)

通过多次运行 K-均值算法发现, 最佳聚类数为 3, 图像背景作为一类, 运动型功能束作为一类, 感觉型功能束作为一类, 这与图 10 所示的人工识别的神经切片图像结果一致。聚类完成后得到 3 个聚类中心的灰度值, 基于这 3 个灰度值划分 3 个灰度范围并把这 3 个灰度范围的像素点的灰度值归一化。以聚类后灰度直方图(如图 7 所示)对比原始图像灰度直方图(如图 5 所示)为例, 聚类完成后 3 个聚类中心点的灰度值分别为 95.4、143.9、175.6, 把灰度值在 (0, 95.4) 范围内的像素点的灰度值统一设置为 10, 同时, 这一类像素点将被设置为一类; 范围在 (95.4, 143.9) 内的像素点的灰度值设置为 138, 同时这些像素点被设置为另一类; 范围在 (143.9, 255] 内的像素点的灰度值设置为 245, 这一类是背景区域。原则上设置的灰度数据对比应该越强烈越好。

造成部分图像不能成功自动进行聚类识别的原因是: (1) 神经切片染色图像过于复杂, 难以提取到更多作为识别功能束类型的有效特征; (2) 部分图片受染效果不好; (3) 图片制备过程中形成的不可避免的人工操作误差, 造成计算机难以自动识别功能束类型, 只有通过医生的经验进行人工判断。

因为 K-均值聚类算法是一种贪心算法, 贪心算法

并不一定能得到全局最优解, 往往会陷入局部最优, 而且 K-均值聚类算法对初始化的类别数 K 和初始聚类中心具有很大的敏感性, 所以本文采用多次运行 K 均值得到多个聚类结果, 选择方差最小的那个结果的方法进行改进。如何改进算法中存在的以上问题以提高算法的效率, 以及如何从医学图像中提取到更多能描绘图像信息的特征从而得到最佳识别效果, 将是下一步亟待解决的问题。

参考文献

- [1] 阮秋琦. 数字图像处理学[M]. 北京: 电子工业出版社, 2000.
- [2] 王虎, 李学东, 杜世新, 等. 微区喇曼光谱法区分周围神经束[J]. 中国神经再生研究, 2006(8): 104-106.
- [3] CASPI O, LESMAN A, BASEVITCH Y, et al. Tissue engineering of vascularized cardiac muscle from human embryonic stem cells [J]. Circulation research, 2007, 100(2): 263-272.
- [4] JOVANOVIĆ -TALISMAN T, TETENBAUM -NOVATT J, MCKENNEY A S, et al. Artificial nanopores that mimic the transport selectivity of the nuclear pore complex [J]. Biophysical Journal, 2009, 96(3S1): 545-545.

(下转第 49 页)

(上接第 41 页)

9 个, 完全满足系统对 FPGA 的要求。

4 结果及分析

按照上述思路和要求选定芯片, 建立工程, 用 Verilog HDL 语言编写程序。将灰度图像读入 MATLAB 中, 生成文本激励, 然后作为输入在 ModelSim 中进行仿真, 再将仿真得到的输出数据送入 MATLAB 中, 生成经锐化处理后的图像。实验前后图像如图 6、图 7、图 8 所示。从实验得到的结果可以看出, 原本边缘模糊的图像经过处理后图像边缘变得非常清晰, 较好地达到了锐化处理的目的。通过对比两种模板得到的图像可知, 选择式(5)模板能取得更好的效果。

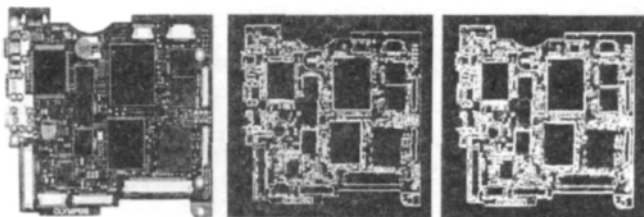


图 6 原图像

图 7 经(3)式模板
处理后的图像

图 8 经(5)式模
板处理后的图像

本文采用 FPGA 来实现图像的锐化处理, 以 Laplacian 算法为核心算法, 确立系统的总体设计, 完成

《微型机与应用》2012 年第 31 卷第 3 期

了主要部分的设计。该系统采用全硬件并行算法实现, 提高了图像处理的整体速度, 仿真结果表明, 本设计符合预期设想。

参考文献

- [1] 冈萨雷斯. 数字图像处理(第二版)[M]. 阮秋琦, 译. 北京: 电子工业出版社, 2007.
- [2] 许录平. 数字图像处理[M]. 北京: 科学出版社, 2007.
- [3] 夏宇闻. Verilog 数字系统设计教程[M]. 北京: 北京航空航天大学出版社, 2003.
- [4] 王冠. Verilog HDL 与数字电路设计[M]. 北京: 机械工业出版社, 2006.
- [5] 叶敏, 周文晖, 顾伟康. 基于 FPGA 的实时图像滤波及边缘检测方法[J]. 传感技术学报, 2007, 20(3): 623-627.
- [6] Altera Corporation. Altera product data sheets[Z]. 2008.

(收稿日期: 2011-11-24)

作者简介:

刘智, 男, 1984 年生, 硕士研究生, 主要研究方向: FPGA 嵌入式及图像处理。

夏春蕾, 女, 1974 年生, 讲师, 主要研究方向: 信息处理与数字图像处理。

戴曙光, 男, 1957 年生, 教授, 主要研究方向: 信息处理, 测试测量及仪器科学。

欢迎网上投稿 www.pcachina.com 45