

改进的中值滤波算法及其 FPGA 快速实现

李飞飞^{1,2}, 刘伟宁¹, 王艳华^{1,2}

(1. 中国科学院长春光学精密机械与物理研究所, 长春 130033; 2. 中国科学院研究生院, 北京 100039)

摘 要: 针对传统中值滤波算法带来的图像模糊问题, 提出一种改进算法, 加入阈值比较环节以便更好地保持图像细节。当用 FPGA 实现中值滤波算法时, 传统方法需要较多的时钟周期, 由此设计一种新的硬件实现电路, 仅用 3 个周期就能快速地取得中值。仿真结果说明, 该改进算法不仅能够取得良好的滤波效果, 而且使所处理的图像更加清晰, 所设计的硬件电路能够快速、高效地对算法进行实现。

关键词: 中值滤波; 现场可编程门阵列; 快速算法; 阈值

Improved Median Filtering Algorithm and Its Fast Implementation in FPGA

LI Fei-fei^{1,2}, LIU Wei-ning¹, WANG Yan-hua^{1,2}

(1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033;

2. Graduate University of Chinese Academy of Sciences, Beijing 100039)

【Abstract】 Aiming at the problem that the traditional median filtering arithmetic blurs the image, this paper introduces an improved algorithm by adding threshold comparison to the traditional median filtering algorithm, which can make the image maintain much more details after processed. As the traditional methods need too many clock periods to implement the median filtering in Field Programmable Gate Array(FPGA), a new circuit is designed, which can quickly get median value of some numbers in three clock periods. The simulation result shows that the algorithm not only can filter the noise better, but also can make the image much clearer, and the designed circuit can implement the arithmetic quickly and efficiently.

【Key words】 median filtering; Field Programmable Gate Array(FPGA); fast algorithm; threshold

1 概述

图像在生成过程中, 往往会受到各种噪声的干扰, 使得图像的质量变差。因此, 在图像处理系统中, 必须对其进行滤波、平滑等预处理来消除噪声。作为一种空域滤波技术, 中值滤波算法与其他滤波算法(如均值滤波)相比, 能有效地消除脉冲噪声和椒盐噪声, 并且能较好地保留图像的边缘信息, 不会使图像变得过于模糊。图像预处理算法处理的数据量非常大, 用一般的软件来实现会比较慢, 但对于一些实时性要求比较高的系统, 如目标识别与跟踪系统, 处理速度往往是要考虑的关键因素, 所以图像预处理算法适宜用硬件来实现。现场可编程门阵列(Field Programmable Gate Array, FPGA)编程灵活、修改方便, 特别适用于流水线方式和并行方式的数据处理。又由于中值滤波算法虽然处理数据量大, 但其在运算过程中不需要图像的统计, 也不需要寄存大量的中间数据, 且运算简单、重复性强, 因此非常适合于映射到 FPGA 架构中由硬件实现。中值滤波在处理图像时或多或少都会使图像变得模糊。本文针对传统的中值滤波算法, 通过加入阈值比较环节, 对算法提出了一些改进, 进一步增强了中值滤波算法保持图像细节的特性。

2 中值滤波算法及其改进

中值滤波是一种非线性滤波方法, 于 1971 年首次提出^[1]。它首先对邻域点的灰度值进行排序, 然后选择中间值作为输出灰度值。中值滤波的公式如下:

$$g(x, y) = \text{med}(f(x-i, y-j)) \quad i, j \in S \quad (1)$$

其中, $g(x, y)$ 和 $f(x, y)$ 为像素灰度值; S 为模板窗口。

中值滤波要实现的是用中值 $g(x, y)$ 来代替中心点值 $f(x, y)$ 。对于椒盐噪声来说, 由于模板的中值很大程度上不会是噪声, 因此中值滤波能很好地消除它。当中心点值为噪声时, 其值和中值数值相差较大, 应该用中值来代替中心点值。但是, 当中心点值不为噪声时, 其值和中值数值相差不大, 如果还用中值来代替中心点值, 那么虽然对于单个像素点来说, 不会产生较大的差异, 但是如果对一整幅图像都进行这样的处理, 这将在一定程度上使整幅图像变得模糊。

据此, 可以对中值滤波算法做一些改进。首先, 排序窗口内的图像数据, 找到中值。然后将中值和中心点值做差, 将差值与事先设定的阈值 T 进行比较, 如果两值之差的绝对值 $|g(x, y) - f(x, y)|$ 大于等于阈值 T , 则认为中心点图像数据是噪声, 用中值代替原数据值; 如果两值之差的绝对值 $|g(x, y) - f(x, y)|$ 小于阈值 T , 则认为中心点图像数据是有效数据, 保持不变。即

$$g'(x, y) = \begin{cases} g(x, y) & |g(x, y) - f(x, y)| \geq T \\ f(x, y) & |g(x, y) - f(x, y)| < T \end{cases} \quad (2)$$

其中, $g(x, y)$ 如式(1)中定义。阈值 T 的选择应该针对不同的图像, 根据经验或者实验进行选取。如果阈值选择过大, 将有可能滤除不掉噪声; 如果阈值选择过小, 将仍会使图像变得模糊。由于噪声和周围像素相差较大, 也和中值相差较

作者简介: 李飞飞(1983-), 男, 硕士研究生, 主研方向: 数字图像处理; 刘伟宁, 研究员; 王艳华, 博士研究生

收稿日期: 2008-03-20 **E-mail:** flyingqiutian@163.com

大, 因此阈值一般选择在几十左右。

这样, 滤波处理之后的图像在除去噪声的同时会更加接近于原图像, 能在更大程度上保留图像细节, 使图像更清晰。

3 FPGA 硬件实现

本设计采用 3×3 模板, 处理的图像数据是 256×256 像素大小, 8 bit。图像数据由 camera link 接口的数字相机提供, 经芯片转化为单端信号后直接送给 FPGA。其中, FPGA 生成的内部电路框图如图 1 所示。

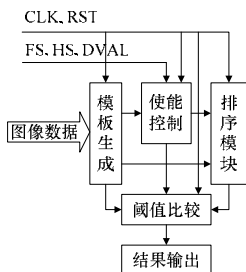


图 1 改进的中值滤波电路结构

由图 1 可知, 本设计分为 4 个模块: 模板生成模块, 使能控制模块, 排序模块和阈值比较模块。其中, HS, FS 和 DVAL 分别是相机的行同步、场同步和数据有效信号。4 个模块分别在统一的时钟信号 CLK 和复位信号 RST 下工作。各模块的功能和设计如下所述。

(1)模板生成模块: 模板生成模块由 3 个移位寄存器组和 2 个 FIFO 组成。因为每行的有效图像数据大小是 256, 滤波模板是 3×3 , 所以由 3 个寄存器和 256×8 bit 的 FIFO 来存储一行图像数据, 其中 FIFO 里应该存储 253 个图像数据。具体的电路结构如图 2 所示。由于图像数据是以数据流的形式依次进入的, 这样, 就可以很容易地取得 3×3 窗口内的图像数据。进行本设计时使用的是 XILINX 的 FPGA 芯片, 考虑到其 FIFO 的稳定性, 在 ISE 里构建 FIFO 时, 没有使用其 IP 核直接生成 FIFO, 而是采用了由双口 RAM 生成 FIFO 的方法, 更有实用性^[2]。

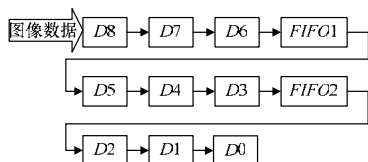


图 2 模板生成模块电路图

(2)使能控制模块: 使能控制模块负责控制模板生成模块、排序模块和阈值比较模块的使能。场同步 FS 到来时, 开始计数随之而来的行同步 HS。在数据有效信号 DVAL 到来时, 开始计数像素数, 同时使能模板生成模块, 将图像数据依次存入移位寄存器和 FIFO。待行同步 HS 计数到 3, 像素数计数到 3 时, 共存储了 2 行零 3 个有效图像数据, 数据恰好存满如图 2 所示的模板生成模块, 此时应该使能排序模块进行数据排序以取得中值。

(3)排序模块: 在对 N 个数 $a[i]$ ($0 \leq i \leq N-1$, N 为奇数) 进行排序时, 如果 N 个数的大小都不相等, 由于排序后的中值处于 N 个数的最中间, 其大于其他数的个数为 $(N-1)/2$, 小于其他数的个数也为 $(N-1)/2$ 。排序时只要分别统计出每个数大于其他数的个数 M , 找到那个 $M=(N-1)/2$ 的数就找到了中值。但是在实际排序时, N 个数难免会出现某 2 个数或某几个数相等的情况, 这将会使数据的排序问题变得比较复杂。

为了避免这种情况的出现, 进行一种假设: 当 2 个数相等时, 认为后出现的数大于先出现的数。即, 如果 $a[i] = a[j]$ ($0 \leq i < j \leq N-1$), 认为 $a[j] > a[i]$ 。这样可以大大简化排序问题。然后, 在分别统计每个数大于其他数的个数 M_i ($0 \leq i \leq N-1$) 时, 设定一个多位寄存器 $Big[k]$ ($0 \leq k \leq C_N^2 - 1$), 每位分别用来标识这 N 个数中某两位数的大小关系, 这个寄存器共需要 C_N^2 位。它的每一位都可以这样设定:

$$Big[k] = \begin{cases} 1 & a[i] > a[j] \\ 0 & \text{其他} \end{cases} \quad (3)$$

其中, i, j, k 有唯一确定的对应关系, 对于不同的 i, j 组合, 只要保证不遗漏不重复, 且 $i < j$ 即可。对具体的某个数统计其大于其他数的个数数 M 时, 只要把相应的寄存器位相加即可。但要注意, 如果此寄存器位寄存的是此数和后面的数的大小比较, 应该直接相加; 如果此寄存器寄存的是它和前面的数的大小比较, 应该将此寄存器位取反。即, 计算 M_i 时, $Big[k]$ 直接取值; 计算 M_j 时, $Big[k]$ 取反。这样就假设实现了: 如果 $a[i] = a[j]$, $Big[k]$ 对于 M_i 是 0, 对于 M_j 是 1。具体到对 9 个数 $D[8:0]$ 进行排序时, 设定寄存器 $Big[35:0]$, 用来标识 9 个数中某 2 个数的大小关系。然后, 对 9 个数分别统计其大于其他 8 个数的个数, 个数为 4 的数即为中值。电路结构如图 3 所示。

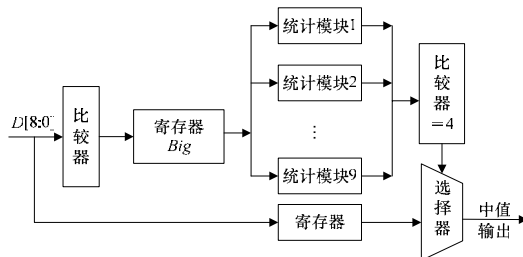


图 3 排序模块电路图

可以看出, 本电路只要 3 个时钟周期就能得到中值结果, 结果输出相对数据输入只有 2 个延迟周期, 这比传统的冒泡排序法和常用的三次排序法都快得多。并且, 在 FPGA 内部利用寄存器暂存处理数据, 实现数据流水, 则处理结果可以顺次输出, 那么每个数据相对自身是延时 2 个周期输出, 相对上个数据输出是没有延时的。

(4)阈值比较: 在使能排序模块后, 应该将 3×3 窗口的中心点数据 $D[4]$ 用 2 个移位寄存器寄存, 延迟 2 个时钟周期, 即排序总共所需时间。之后, 将寄存的 $D[4]$ 和排序所得中值作比较, 根据式(2), 决定是将 $D[4]$ 作为结果输出还是将中值作为结果输出。

4 FPGA 实验结果

本设计用 ISE9.1 在 XILINX 的 XC2VP4 芯片里实现。布局布线后, 结果显示, 占用 Slice register 289 个, 为 4%; 占用 4 input LUT 571 个, 为 9%; 占用 Slice 443 个, 为 14%。可以看出, 本设计占用系统资源是相当少的, 仿真可达最高频率为 171 MHz。

本设计用 Modelsim 6.0 进行仿真, 先用 Matlab 将一幅分辨率为 256×256 的图像转换成灰度值在 0~255 之间的矩阵^[3], 再转化为 Modelsim 6.0 的测试向量文件, 将测试向量文件用本设计测试后, 显示结果如图 4 所示。然后利用 Matlab 把测试结果转化为图像矩阵显示, 结果如图 5 所示。图中的传统中值滤波图像是将本电路的阈值比较部分去掉后所得。

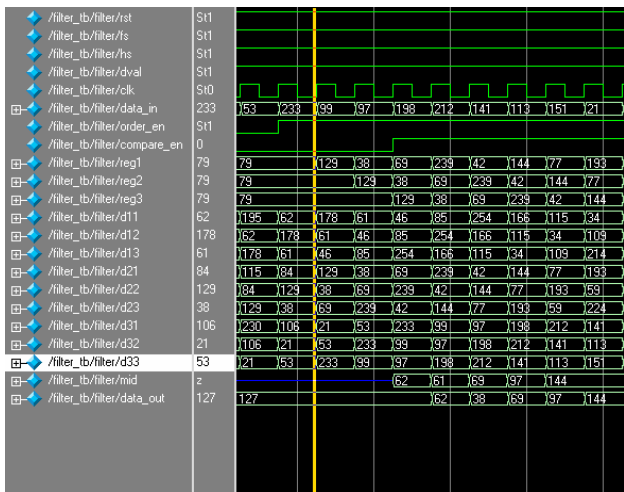


图4 Modelsim 仿真结果

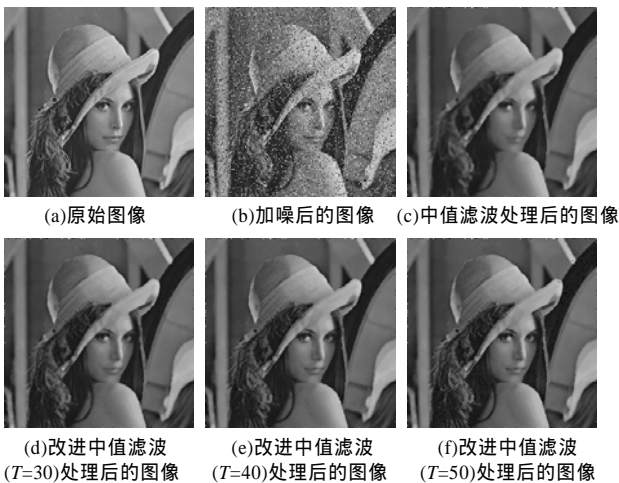


图5 Matlab 仿真结果

可以很明显地看出, 传统的中值滤波算法在一定程度上使图像变得模糊; 本文的改进中值滤波算法更加忠实于原图像, 保留了较多的细节信息。图中改进中值滤波算法的阈值 T 分别取 30, 40, 50, 可以看到, 阈值为 40 时滤波效果最好, 图像也最为清晰; 阈值为 50 时, 有些噪声点没有滤除掉。由于本设计没有对图像的边缘滤波输出, 因此所得滤波图像大小为 254×254 , 这些损失相对于整幅图像来说, 是可以接受的。如果要按 256×256 输出的话, 可以对始末行和始末列分

(上接第 174 页)

4 结束语

本文提出了一种支持 JPEG2000 动态感兴趣编码的算法 RCWARC。算法利用 JPEG2000 编码能精确控制生成码率的特点, 利用率失真函数的凸函数特性, 通过对感兴趣区域增加码率而对背景降低码率的方法, 达到 ROI 优先编码的目的。算法基于块级处理, 生成码流与标准兼容。

参考文献

- [1] Taubman D, Marcellin M W. JPEG2000: Image Compression Fundamentals Standards, and Practice[M]. Boston, USA: Kluwer Academic Publishers, 2002.
- [2] Nguyen A, Chandran V. Important Prioritization Coding in JPEG2000 for Interpretability with Applications to Surveillance

别进行行列复制, 然后利用本电路进行处理。或者将模板分别更改为 1×3 和 3×1 来对始末行和始末列进行滤波处理。

为了客观评价此方法的有效性, 本文采用了最小均方误差 MSE 和信噪比 SNR 2 个性能指标来评价图像质量。

最小均方误差的计算公式为

$$MSE = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (\hat{f}_{i,j} - f_{i,j})^2 \quad (3)$$

其中, $\hat{f}_{i,j}$ 表示滤波后图像像素的灰度值; $f_{i,j}$ 表示原始图像各点的灰度值。

信噪比的计算公式为

$$SNR = 10 \cdot \lg\left(\frac{\sigma_{\hat{f}_{i,j}}^2}{MSE}\right) \quad (4)$$

其中, $\sigma_{\hat{f}_{i,j}}^2$ 表示滤波后图像灰度值的方差。

对本系统处理的图像进行质量评价后, 结果显示如表 1 所示。可以看出, 改进的中值滤波算法能保持较高的图像质量, 并且阈值选取的不同会影响图像质量。

表1 2种滤波方法的图像质量评价

性能指标	传统中值滤波	改进后的中值滤波		
		$T = 30$	$T = 40$	$T = 50$
MSE	469.139	73.126	47.431	77.219
SNR	7.592	15.841	17.766	15.679

5 结束语

本文通过对传统的中值滤波算法进行改进, 能够更好地保证图像质量。把此算法用 FPGA 实现时, 通过消除两数相等情况的出现, 大大简化了排序步骤, 可以在相当小的延迟周期内得到处理结果, 且占用资源也较少。如果增大滤波模板的话, 可以依照此方法重新设计排序部分电路, 但排序周期不会随着模板的增加而增加, 仍将保持 3 个周期不变。

参考文献

- [1] 夏良正. 数字图像处理[M]. 南京: 东南大学出版社, 1999: 154-160.
- [2] Virtex-II Pro and Virtex-II Pro X FPGA User Guide[Z]. Xilinx Inc., 2007.
- [3] Gonzalez R C, Woods R E, Eddins S L. Digital Image Processing Using Matlab[M]. 阮秋琦, 译. 北京: 电子工业出版社, 2006.

编辑 顾逸斐

Image[C]/Proc. of SPIE Visual Communication and Image Processing. [S. l.]: SPIE Society, 2003: 806-817.

- [3] Rosenbaum R, Schumann H. Flexible, Dynamic and Compliant Region of Interest Coding in JPEG2000[C]/Proc. of IEEE International Conference on Image Processing. New York, USA: [s. n.], 2002: 101-104.
- [4] Sanchez V, Basu V, Mandal M K. Prioritized Region of Interest Coding in JPEG2000[J]. IEEE Trans. on Circuits and Systems for Video Technology, 2004, 14(9): 1149-1155.
- [5] 侯俊, 方向忠. 码率 JPEG2000 的感兴趣区域编码[J]. 计算机工程, 2007, 33(5): 202-203.

编辑 顾逸斐