

# 基于中值滤波算法滤波器的 FPGA 实现<sup>①</sup>

李新春, 赵璐

(辽宁工程技术大学 电子与信息工程学院, 葫芦岛 125105)

**摘要:** 通过分析 3\*3 窗口滤波的数学模型, 以 FPGA 为平台, 使用 VHDL 硬件描述语言设计, 实现中值滤波图像处理算法。在设计过程中, 通过改进的中值滤波算法和优化结构, 在合理利用硬件资源的基础上, 有效地运用算法内在的并行性, 同时采用流水线结构优化改进算法, 提高了处理速度。

**关键词:** FPGA; 中值滤波; 流水线

## Implementation of FPGA Based on Median Filtering Algorithms Filter

LI Xin-Chun, ZHAO Lu

(College of Electronic Information, Liaoning Technical University, Huludao 125105, China)

**Abstract:** Through the analysis of the mathematical model of 3\*3 template based on FPGA platform, this paper uses VHDL hardware description language to design and implement median filtering algorithm. In the design, through improved algorithms and optimizing the structure, the rational use of hardware resources is made, with the internal parallelism in the algorithm effectively used. At the same time, the pipelining uses structural optimization algorithm and improves the processing speed.

**Key words:** FPGA; median filtering; pipelining

随着时代进步, 技术的不断提高, 现在图像处理系统对实时性的要求越来越高, 但是图像往往存在噪声干扰。去除图像噪声的过程称为滤波。若使用通用的软件方法, 由于处理过程需要进行大量的简单运算, 因此会消耗大量时间而影响系统的实时性要求, 因此选用硬件实现比较适合。而随着现代 EDA 技术的快速发展, 用 FPGA 来完成图像滤波是最好的选择。

## 1 中值滤波算法

中值滤波的基本原理是把数字图像或数字序列中一点的值用该点一个邻域内各点的中值来代替。把一个特定长度或者特定形状的邻域称作窗口, 一维数字序列  $\{F_i, i \in Z\}$ , 窗口长度为  $n$  (奇数), 对此一维序列的中值滤波公式为  $Y_i = \text{Median}(F_{i-r}, \dots, F_i, \dots, F_{i+r})$ , 式中  $i \in Z$ ,  $r = (n-1)/2$ , 以  $Y_i$  取代窗口中心像素值作为中值滤波输出。二维数字图像为  $\{F_{ij}, (i, j) \in Z^2\}$ ,  $F_{ij}$  为图像上坐标为  $(i, j)$  像素的灰度, 对窗口大小为  $A = m * n$  (含

有奇数个像素) 的中值计算公式为  $Y_{ij} = \text{Median}(F_{i+r, j+s}, (r, s) \in A)$ , 式中  $(i, j) \in Z^2$ , 以  $Y_{ij}$  取代二维窗口中心像素灰度作为中值滤波结果输出<sup>[1]</sup>。中值滤波可以有效去除脉冲型噪声, 同时对图像的边缘有较好的保护。

对于一幅 128\*128 的灰度图像进行 3\*3 方形窗中值滤波处理, 总共需要处理 16384 个像素, 对每个像素点及其邻域取中值需要有 36 次的排序运算。由此可见, 中值滤波算法运算量非常大, 本文提出了改进的中值滤波算法。

## 2 改进中值滤波算法研究

传统中值滤波算法的实现是对窗口内像素点值进行全排序。排序的过程是对像素作比较和交换的过程, 比较次数是影响排序速度的重要原因。对于一个像素数为  $n$  的窗口数列, 第一步遍历整个数列, 记录最大值; 第二步将上一步得到的最大值除去在遍历数列, 记录第二大的值, 如此循环到剩下最后一个值为止,

① 收稿时间:2010-12-18;收到修改稿时间:2011-03-07

这时得到第 $(n+1)/2$ 个数,即为中值输出。在整个比较中,共作 $n*(n-1)/2$ 次比较操作,时间复杂度为 $O(n^2)$ 。算法处理时间长,不利于实现图像处理<sup>[2]</sup>。

根据传统中值滤波算法的这些缺点,本文提出一种窗内像素灰度值排序的改进算法,即改进中值滤波算法。改进中值滤波算法针对传统中值滤波算法的特点采用并行处理和流水线操作,将大问题分解为若干的子问题来并行处理,不仅大大减少了运算量,在 FPGA 硬件上也实现了占用更少的硬件资源。下面以 $3*3$ 窗口数据模型为例来说明中间值的求取过程。

设 $3*3$ 窗口的输入数据如下表 1:

表 1  $3*3$ 窗口像素排列表

	第一列	第二列	第三列
第一行	h11	h12	h13
第二行	h21	h22	h23
第三行	h31	h32	h33

其中,用 min 表示取最小值操作,用 mid 表示取中值操作,用 max 表示取最大值操作。

第 1 步:采用并行处理对每一行的像素数据进行由小到大的排序,能够同一时间得到每行的最小值、中间值和最大值。记录第一行像素的最小值为 $\min\_row1=\min(h11,h12,h13)$ ,中间值为 $\mid\_row1=\mid(h11,h12,h13)$ ,最大值为 $\max\_row1=\max(h11,h12,h13)$ ,同理第二、第三行依次类推。得到结果如下表 2 所示:

表 2  $3*3$ 窗口像素第一次比较列表

	第一列	第二列	第三列
第一行	min_row1	mid_row1	max_row1
第二行	min_row2	mid_row2	max_row2
第三行	min_row3	mid_row3	max_row3

这样每行经过 3 次的比较,总共比较 9 次就得到了三个数据组,分别是第一列数据组,最小值数据组、第二列数据组,中间值数据组和第三列数据组,最大值数据组。

第 2 步:对所得三组数据在分别进行并行处理,最小数据组中经过 3 次比较得到最大值为 $\max\_min=\max(\min\_row1, \min\_row2, \min\_row3)$ ,同理经过 3 次比较取得中间值数据组的中间值为 $\mid\_mid=\mid(\mid\_row1, \mid\_row2, \mid\_row3)$ ,经过 3 次比较最大值数据组的最小值为 $\min\_max=\min(\max\_row1, \max\_row2, \max\_row3)$ ,获得这三个值总共需要 9 次比较。

选取这三个值的原因分析如下:

(1)假设对最小数据组按照由小到大的顺序排列如下: min\_min, mid\_min, max\_min

在这三个数据中,首先 min\_min 是最小数据组中的最小数据,也一定是整个数据组中的最小数据,所以一定不是中值。其次 mid\_min 这个数据,比它小的数据最多有 3 个,即三个数据组的最小数据,这 3 个数据,而比它大的数据有 5 个即三个数据组的最大数据和中间值数据组与最大数据组的中值数据,故 mid\_min 定然也不是中值。而 max\_min 则有可能是整个数据组的中间值。

(2)假设对中间值数据组按照由小到大的顺序排列如下: min\_mid, mid\_mid, max\_mid

这三个数据中,首先 min\_mid 是中间值数据组的最小数据,那么比它小的数值最多有 3 个,即最小值数据组的 3 个数据,而比它大的数据至少有 5 个, min\_mid 的两边一定不存在相等数目的数据,故 min\_mid 不可能是中值;其次, max\_mid 是中间值数据组的最大数据,比它小的数值至少有 5 个,即最小值数据组的 3 个数值和中间值数据组中的其他 2 个数值,而比它大的数据最多有 3 个, max\_mid 数据两端也一定不存在相等数据的可能性,故 max\_mid 不可能是中值。所以在中间值数据组中只有 mid\_mid 有可能是中值。

(3)假设对最大值数据组按照由小到大的顺序排列如下: min\_max, mid\_max, max\_max

这三个数据中,首先 max\_max 是最大值数据组的最大数据,那么它也是所有数据的最大数据,故一定不是中值。其次, mid\_max 数据是最大值数据组中的中值数据,但是比 mid\_max 数值大的数据最多有 3 个,分别是三个数据组的最大数据,而比 mid\_max 小的数据一定有 5 个,即三个数据组的最小数据和中间值数据组的中值数据, mid\_max 的两端数据数目不可能相等,故 mid\_max 数值一定不是中值数值。而 min\_max 则有可能是中值数据。

基于以上原因,选择 max\_min, mid\_mid, min\_max 这 3 个数据来进行下一步分析。

第 3 步,由以上分析可知,窗口中 9 个数据的中值就在选取好的 3 个数据中,对选取好的 3 个可能数据经过 3 次比较就可以找到它们的中值数据,即为所求得的所有数据的中值。

整个算法可以分成以上三个步骤,而且每个步骤在硬件平台上完全可以并行执行,这样可以大大提高排序的效率。而且,算法不是对窗口中的9点数据的完全排序,而是采用分组比较的办法找到中值,简化了排序过程,从而大大提高了处理效率。

针对  $N \times N$  方形窗口数据模型中值求取,按照上述的  $3 \times 3$  窗口数据模型中间值的求取过程来分析计算即可。采用并行处理对每一行的像素数据进行由小到大的排序,能够同一时间得到每行的最小值和最大值,同时也可以求取中一行中的中值;对所得三组数据在分别进行并行处理,取出最小值组中的最大值和最大值组中的最小值,以及中值组中的中间值。对取出的三个值再通过比较得到中间值,即为整个数据模型的中间值。

### 3 算法分析

传统的中值滤波算法要对窗口内的所有像素点的灰度值从小到大依次排列,并以灰度值序列的中间值来代替窗口中心位置像素点的灰度值,即对于  $N \times N$  方形窗的中值滤波器来说,运行一次就要进行  $1/2 \times N^2 \times (N-1)$  次排列比较运算,可以想象要处理一幅  $128 \times 128$  的灰度图像,或者  $256 \times 256$  的灰度图像,就要进行更大量的比较运算,所需的时间是相当长的。

而采用改进的中值滤波算法是将一个  $N \times N$  窗口的数据,先进行行、列的排序运算后,直接对取得的三个像素值再比较即可取得中值。以  $3 \times 3$  窗口为例,行、列排序均需要9次的运算,在对取得的三个像素值比较大小,需要3次运算,则对一组9个像素取中值运算一共需要进行21次的比较运算,而传统算法要进行36次运算。对于  $5 \times 5$  窗口,差别更加的明显,传统中值滤波算法需要300次比较运算,而改进后的算法只需要125次比较运算。由此可见,改进后的中值滤波算法相对传统算法在运算次数上大大减少,从而提高了数据处理速度,更好满足实时性要求;同时,这种算法易于实现,占用更少的芯片资源,提高硬件利用率。

### 4 中值滤波的FPGA实现

FPGA是一种基于查找表结构的可编程逻辑器件,具有大量的灵活的逻辑单元,很好的在线可编程性和高速并行的结构特点。它可以方便地实现组合逻辑和时序逻辑功能<sup>[3]</sup>。FPGA较小的基本逻辑单元(LC),可以通过级联实现较长的数据通路,适合实

现流水线结构。本设计采用 Altera 公司的 ACEX1K 系列的 FPGA 来实现上述中值滤波算法。软件运行环境为 QuartusII6.0,程序采用 VHDL 语言编写,自上而下模块化设计,增加了程序的可读性和可维护性,并易于综合器综合。

基于 FPGA 的中值滤波器的内部工作流程图如图 1 所示。从流程图中我们可以看到,改进的中值滤波算法的硬件设计上,首先考虑构成一个三点比较器。该三点比较器是后续运算处理的基础。程序中 a、b、c 为三点输入数据,clk 为时钟信号,rstn 为复位信号,max、mid、min 分别为最大值、中值、最小值输出。

三点比较器的部分 VHDL 源程序如下:

```
.....
architecture behav of 3dianbijiao is
//实体名称为 3dianbijiao, 结构体名称为 behav//
signal h1,h2,min1,max1,min2,max2:std_logic_vector(7
downto 0); //定义 6 个信号
begin
process(clk,rstn)
begin
if rstn='0' then
h1<=(others=>'0'); h2<=(others=>'0');
min1<=(others=>'0'); max1<=(others=>'0');
min2<=(others=>'0'); max2<=(others=>'0');
//初始化过程
elsif rising_edge(clk) then //上升沿有效
if (a<b) then
min1<=a;max1<=b; else
min1<=b;max1<=a; //比较 a,b 两个数
大小,
endif;h1<=c; // 大者存在 max 中, 小者
存在 min
if(h1<min1) then
min2<=h1;max2<=min1; else
max2<=h1;min2<=min1;
//c 与 a,b 中较小者的比较
end if;h2<=max1;
if (max2<h2) then
max<=h2;med<=max2;min<=min2;
else
max<=max2;med<=h2;min<=mi
```

n2; //输出 a,b,c 的大小  
.....

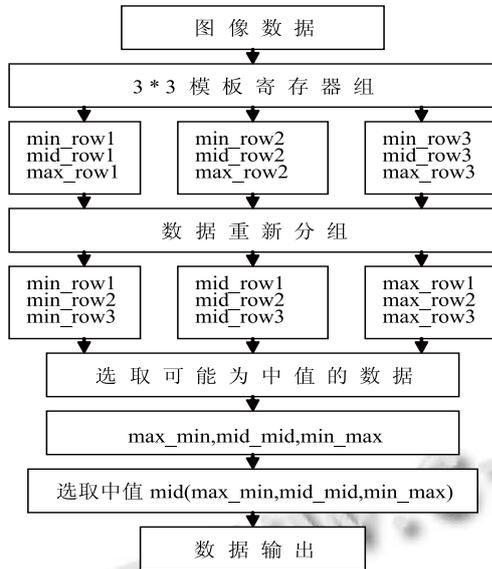


图 1 基于 FPGA 的中值滤波器的内部工作流程

通过综合编译，三点比较器 RTL 实现原理如图 2 所示，三点比较器仿真图形如图 3 所示。

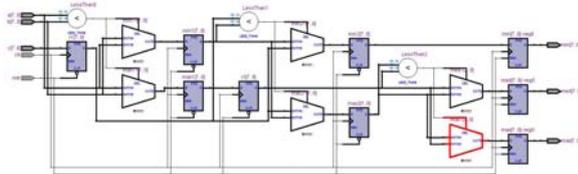


图 2 三点比较器的 RTL 实现原理图

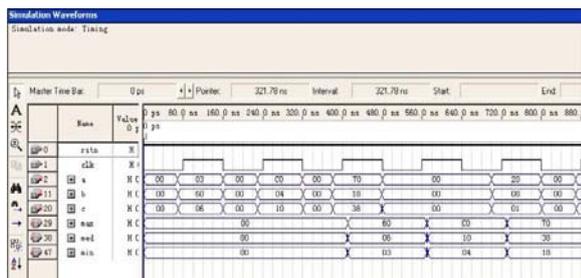


图 3 三点比较器的仿真图形

在三点比较器的基础上实现改进的中值滤波功能就非常方便了，可以使用元件例化语句进行设计，也可以使用包装入库的方法。包装入库的方法是使用 QuartusII6.0 中的“File—Create—Create Symbol Files for Current File”选项，将当前的文件变成了一个包装好的单一元件 (Symbol)，并被放置在工程路径指定的

目录中，即将这个三点比较器包装元件入库，然后在原理图法设计底层结构中直接调用该元件模块<sup>[4]</sup>。

改进中值滤波算法模块顶层硬件原理图由 7 个三点比较器模块组成。共分为三级，第一级 3 个比较器完成对窗中 9 个像素值中三行行排序，第二级的 3 个比较器完成对列排序。第三级的一个比较器完成对选中 3 个像素值的排序，输出中间值。所有的三点比较器模块的时钟输入信号都和全局时钟信号 CLK 相连，复位信号也与全局复位信号 RST 相连，采取统一时钟，统一复位原则<sup>[5]</sup>。第一级和第二级的 3 个比较器均采用并行处理的原则，整体采用流水线结构，保证了数据处理的速度与数据的传输速度同步。电路时序仿真的波形图如图 4 所示。

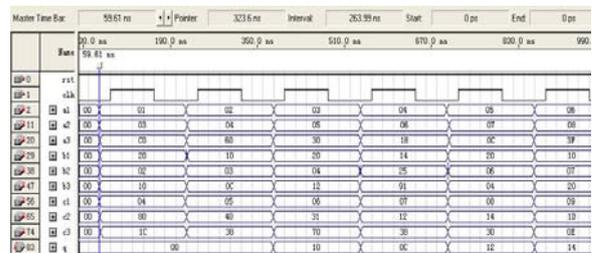


图 4 中值滤波电路时序仿真波形

通过仿真波形，可以看到经过两个时钟周期的处理就可以得到每组数据的中值。由于程序的实现采用了流水线设计，将中间值的求取过程分为了 3 个阶段，造成了有两个时钟周期的延迟。如波形中第三个时钟时间才输出第一组数据中值。虽然带来了 2 个时钟周期的延迟，但是却解决了系统的工作频率的瓶颈问题，提高了系统的带宽。算法中的比较操作是由 FPGA 中的查找表来完成的，查找表实现的逻辑运算延时是固定的，而且可以实现很高的处理速度，所以具有很好的实时性和可靠性。

### 5 结论

本文分析了基于 FPGA 的中值滤波算法的基本原理和并行处理、流水线结构，同时采用模块化设计和流水线设计的思路和方法，进行了实验仿真验证。经过仿真得到的波形与理论算法得到结论吻合，验证了算法及程序的正确性。改进的中值滤波算法较比传统中值滤波在 FPGA 实现上占用的资源少很多。由于使

(下转第 72 页)

如图7和表1所示,随着种群规模的增大,本文算法能够保持较高的计分值,且迭代次数也比较少。而GA,PSO和HPGA算法受种群规模影响比较大,而且易于过早收敛。由此证明本文算法搜索能力的优越性。而由计分值和迭代次数充分说明本文算法既能保持良好的搜索精度,又能快速收敛到最优结构。

## 5 结语

本文提出一种适用于贝叶斯网络学习的改进的遗传算法的。相对于传统遗传算法,该改进算法提出了两个适用于贝叶斯网络结构学习的操作算子:优化变异和修正非法图。仿真证明使用该改进的遗传算法对数据库进行学习,可得到全局最优的贝叶斯网络结构。相对于已有文献传统算法<sup>[4-6]</sup>和智能优化算法(PSO,GA,HPGA),该算法具有更好的学习能力和收敛速度,从而具有更好的搜索能力。但是由于现有评分函数的局限性会导致在不同网络结构上出现相同的计分值,从而导致优化后的结构离最优结构有一定距离,如何设计一个更有效的评分函数将是未来研究的努力方向。

## 参考文献

- 1 Bultan T, Fu X, Hull R, Su J. Conversation specification: A new approach to design and analysis of e-service composition. Proc. of 12th Int'l World Wide Web Conf., May 2003.
- 2 王双成,冷翠平,李小琳.小数据集的贝叶斯网络学习.自动化学报,2009,35(8):1064-1070.

- 3 孙岩,唐一源.新的贝叶斯网络结构学习方法.计算机工程与设计,2008,29(5):1238-1240.
- 4 Cooper GF, Herskovis E. A Bayesian method for the induction of probabilistic networks from data. Machine Learning, 1992, 9(4):309-347.
- 5 Chickering DM. Optimal structure identification with greedy search. Journal of Machine Learning Research, 2002,11(3): 507-554.
- 6 Sahin F, Yavuz M, Amavu Z. et al. Fault diagnosis for airplane engines using Bayesian network and distributed particle swam optimization. Parallel Computing, 2007, 11(33):124-143.
- 7 许丽佳,黄建国,王厚军,龙兵.混合优化的贝叶斯网络结构学习.计算机辅助设计与图形学报,2009,21(5):633-639.
- 8 Friedman N, Goldszmidt M. Building classifiers using Bayesian network. Proc. Nation Conference on Artificial Intelligence. Menlo park, CA: AAAI Pres, 1996:1227-1284.
- 9 尹作海,邱洪泽,周万里.基于改进变异算子的遗传算法求解柔性作业车间调度.计算机系统应用,2009,18(10):156-159.
- 10 Larranaga P, Poza M, Yurramendi Y, et al. Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1996,18(9):912-926.
- 11 邓宏贵,罗安,曹建,丁家峰,王会海.基因多点交叉遗传算法在变压器故障诊断中的应用.电网技术,2004,28(24):467-470.
- 12 Kevin P, Murphy. The Bayes Net Toolbox for Matlab.[2001]. <http://www.cs.ubc.ca/~murphyk/>

(上接第85页)

用资源减少,可为在FPGA上实现其他功能预留资源。并且在FPGA上实现中值滤波算法的硬件结构简单,集成度高,可靠性强,时序固定,延时小等优点。由于FPGA可编程的特点,增加了系统的灵活性,稍作修改就可以适合于不同的系统,有很强的通用性,所以,本文提出的滤波器设计方法具有很强的实用性。

## 参考文献

- 1 Gavin L,Saeid N.FPGA implementation of a median filter.

TENCON' 97 IEEE Region 10 Annual Conference. Australia, 1997:437-440.

- 2 陈加成,徐熙平,吴琼.基于FPGA的中值滤波算法研究与硬件设计.长春:长春理工大学,2008.
- 3 潘松,黄继业.EDA技术实用教程.北京:科学出版社,2005.
- 4 刘皖,何道君,谭明.FPGA设计与应用.北京:清华大学出版社,2006.
- 5 付强.基于FPGA的图像处理算法的研究与硬件设计.南昌:南昌大学,2006.