

一种快速的二维中值滤波算法及其硬件实现

郑 鹤, 王鲁平, 李 飙

(国防科技大学 ATR 重点实验室, 湖南 长沙 410073)

摘 要: 为了实现实时图像预处理, 介绍了中值滤波器的原理, 详细研究了一种快速的二维中值滤波算法, 给出了Verilog程序流程, 并在现场可编程门阵列 (FPGA) 上用硬件编程语言实现了此中值滤波器。通过对不同算法的仿真结果进行了详细分析和比较, 表明此算法大幅度降低了FPGA的资源占用率, 能有效控制系统成本, 适合用于硬件实现。

关键词: 信息处理技术; 中值滤波器; 二维中值滤波算法; 快速算法; FPGA

中图分类号: TN 911. 73 **文献标识码:** A **文章编号:** 1672-2892 (2005)04-0245-04

Implementation of Fast 2-D Median Filter on FPGA

ZHENG He, WANG Lu-ping, LI Biao

(State Key Lab of ATR, NUDT, Changsha 410073, China)

Abstract: The elements of median filter are introduced, then a fast 2-D median filter algorithm and its implementation are described. Comparing the fast algorithm and the regular one, the former requires much less resource of the FPGA. So its implementation can reduce the cost of the hardware system.

Key words: information processing technology; median filter; 2-D media filter algorithm; fast algorithm; FPGA

1 引言

中值滤波作为一种非线性滤波技术, 能够有效地抑制脉冲噪声(Impulsive Noise)、椒盐噪声(Salt & Pepper Noise)以及解决相关的干扰问题, 而且能够有效地保护边界信息, 因此在图像处理中得到了广泛的应用。但是大部分排序算法具有循环迭代结构和计算次数不确定性(即输入到输出的时延不确定性)的缺陷, 这就决定了中值滤波器只适合于软件的实现而不适合于硬件的实时实现, 基于上述考虑, 研究了一种易于硬件实现的快速二维中值滤波算法。此算法只需进行极少量的循环迭代, 且消除了输入到输出的时延不确定性, 并在现场可编程门阵列(Field Programming Gate Array, FPGA)上用硬件语言编程实现了这种算法, 并对结果进行了详细的分析^[1]。

2 中值滤波原理

对于给定的 n 个数值 $\{a_1, a_2, \dots, a_n\}$, 将它们按照大小有序排列。当 n 为奇数时, 位于中间位置的那个数值称为这 n 个数值的中值; 当 n 为偶数时, 位于中间位置的那两个数值的平均值称为这 n 个数值的中值^[1]。记为 $med(a_1, a_2, \dots, a_n)$ 。

二维中值滤波一般采用一个 $m \times n$ 的滑动窗口, 从左至右, 从上到下逐行移动, 其中 m 为滑动窗口行数, n 为滑动窗口列数。对滑动窗口内像素点灰度值进行排序, 选择排序像素集的中间值作为指定像素点的灰度值^[2]。

中值滤波器的主要运算就是对窗口内的数据进行排序。排序的基本单元是两个数的比较大小后排序, 如图1。

假设滤波窗口为 $m \times n$, 则需要设置一个大小为 $m \times n$ 的寄存器, 将滤波窗内数据存入寄存器, 然后经过排序运算, 求出寄存器中数据的中值。因为排序进行到求出中值时就可以停止, 即进行到求出第 $(m \times n - 1)/2$ 个最大值时就可以停止。这样, 比较运算次数为:

$$(m \times n - 1) + (m \times n - 2) + \dots + \left(m \times n - 1 - \frac{m \times n - 1}{2}\right) = \frac{3}{8}(n^2 \times m^2 - 1) \quad (1)$$

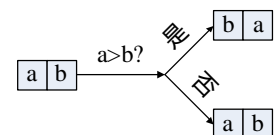


图 1 二数比较

因此,用常规算法求出滤波窗口为 $m \times n$ 的中值,要进行 $\frac{3}{8}(n^2 \times m^2 - 1)$ 次二数比较运算。对于 3×3 窗口,这个比较运算次数为30;对于 7×7 窗口,这个数值增加到900。可见,随着窗口尺寸的增加,运算次数急剧增大,使得处理速度变慢,而且随着需要的寄存器尺寸增加,对资源的占用率急剧增大,使得没有多余的资源留给后续的处理程序,甚至要用更为昂贵的器件才能完成处理。而在文献[3]中所提出的一种二维中值滤波的快速算法很好地解决了这两个问题。

3 快速二维中值滤波器算法

3.1 3×3 中值滤波器

我们先来看一个滤波窗口大小为 3×3 的中值滤波器的快速算法。首先将每一列按照升序排序,然后取第一排的最大值、第二排的中值、第三排的最小值,最后取这三个值的中值,得到了这个 3×3 窗口内像素的中值。下面是此算法的伪代码,图2详细地说明了这个过程^[3]。

```
for c = 0 to 2
  sort column c so that A[r-1,c] <= A[r,c]
  let max0 = max (A[0,0], A[0,1], A[0,2])
  let med1 = med (A[1,0], A[1,1], A[1,2])
  let min2 = min (A[2,0], A[2,1], A[2,2])
  let med3 = med (max0, med1, min2)
```

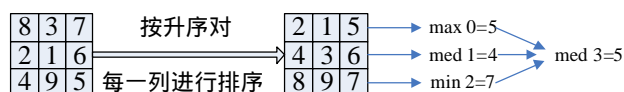


图2 滤波窗口大小为 3×3 的快速中值滤波器算法

按升序对每一列进行排序,所需要比较大小 $3 \times 3 = 9$ 次,第二个步骤所需比较大小 $3 \times 3 = 9$ 次,最后一次求中值需要3次比较运算,因此这种算法只需要进行21次比较大小的运算,而常规的排序算法需要30次比较大小的运算。

3.2 $N \times N$ 中值滤波器快速算法

将 3×3 的中值滤波器快速算法推广至 $N \times N$ 的中值滤波器,其算法如图3所示;图4以一个滤波窗口大小为 5×5 的中值滤波器详细说明了算法的步骤^[3];文献[3]中对此算法作出了详细的证明。

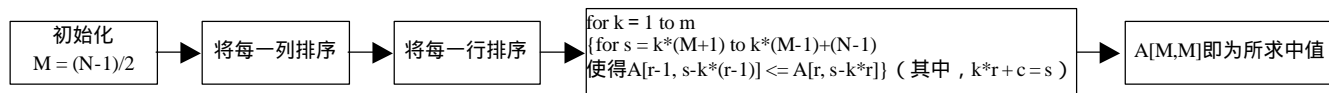


图3 $N \times N$ 窗口的快速中值滤波算法流程图

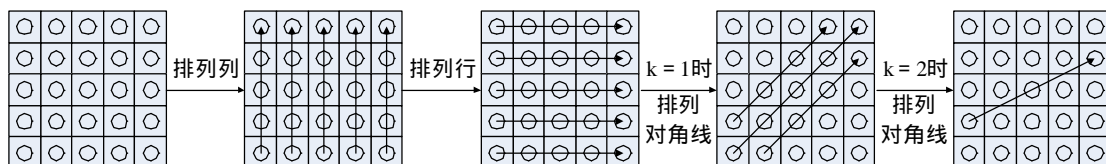


图4 滤波窗口大小为 5×5 的快速中值滤波器

表1详细比较了这种快速算法与常规算法所需要的比较次数,根据表1的数据,做出图5。从图5中可以明显地看出,随着滤波窗口的增大,常规算法所需要的比较次数急剧增加;而快速算法所需要的比较次数增加幅度就平缓得多。在窗口大小为 9×9 时,快速算法所需要的比较次数仅约为常规算法的1/4。

表1 不同窗大小所需要的比较次数

	3×3	5×5	7×7	9×9
常规算法	30	234	900	2460
快速算法	18	125	298	618

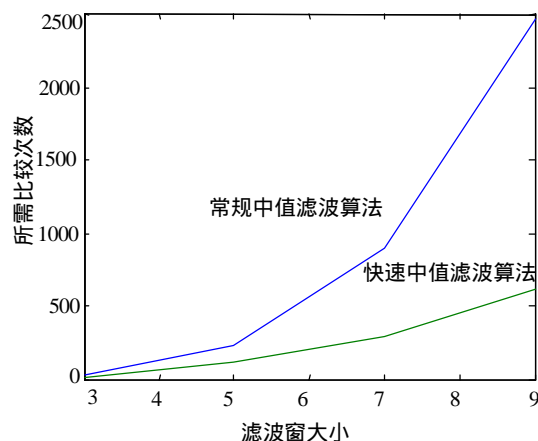


图5 不同窗大小所需要的比较次数

4 硬件实现及结果比较

为了实现图像的实时处理,设计了如图6所示的硬件系统。此系统中,所有的算法都在FPGA中完成。软件结构如图7所示(5×5窗口)。

采用美国Xilinx公司的vertex2 2V2000FG676芯片,在Active-HDL平台上用verilog语言进行编程,用Synplicity pro 7.1进行综合,在ISE6.1平台上实现,所得时序仿真结果如图8和图9所示。

在Xilinx6.1上实现后,软件给出了详细的结果报告。报告中对各个方面的结果都有详细的记录,通过查阅报告,可以看出此快速算法的优点主要表现在报告中的资源占用率上。表2为报告中常规算法和快速算法实现后所占用FPGA资源,表格中所有数据来自软件报告。

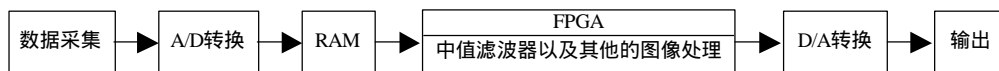


图6 图像处理硬件结构

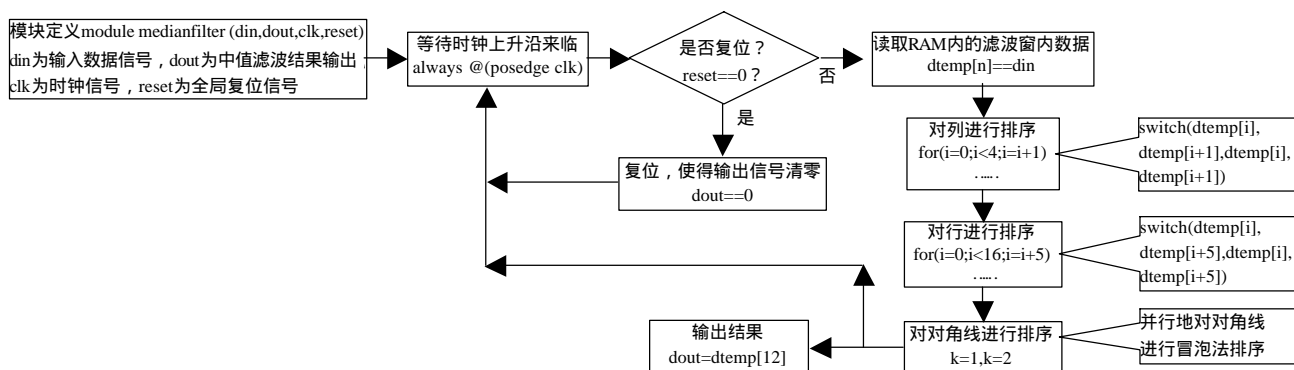


图7 verilog编程^[4]的软件结构

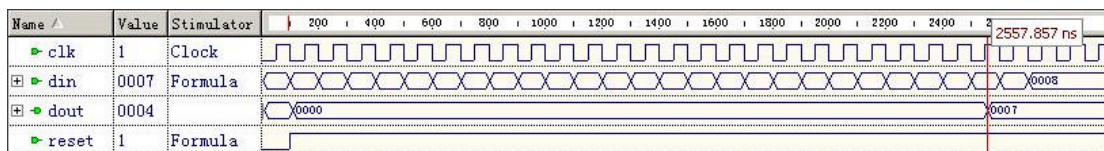


图8 窗口为5×5的常规中值滤波器时序仿真结果

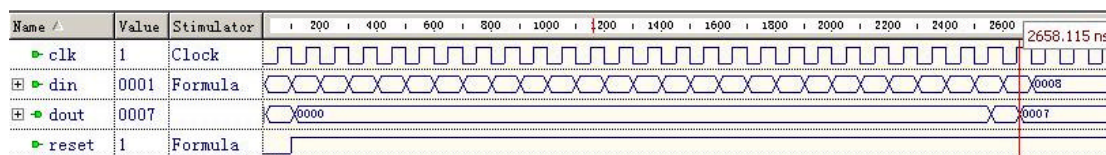


图9 窗口为5×5的快速中值滤波器时序仿真结果

表2 常规算法和快速算法实现后所占用FPGA资源比较

	Slice Flip Flops		4 Input LUTs		Bonded IOBs		Occupied Slices	
	个数	比例(%)	个数	比例(%)	个数	比例(%)	个数	比例(%)
常规3×3	9	1	12	1	19	4	7	1
快速3×3	216	1	1150	5	34	7	579	5
常规5×5	424	1	16042	74	34	7	8078	75
快速5×5	424	1	3357	15	34	7	1690	15

从表2可以看出,随着滤波窗的增大,按常规算法来实现的滤波器,所需的资源急剧增加,仅仅对于5×5的窗口就几乎占用了Vertex2 2V2000FG676的大部分4 Input LUTs和Slices;而此快速算法不仅仅在比较次数上有很大的优势,在资源占用上也有很大的优势。可以看出,从3×3窗口到5×5窗口,需要比较的数据几乎增大三倍,但是所需要的4 Input LUTs却才增加了2倍多一点,Slices也只占用了15%。事实上,用此快速算法进行编程,不需要用Vertex2 2V2000FG676这种高成本的芯片,只需要用SpartanIIE这种成本相对较低的芯片就能胜任。究其原因,因为在常规算法中 $N \times N$ 窗口滤波器需要在FPGA中开辟的寄存器为 $1 \times N^2$ 来放置待处理的数据,且在处理过程中,还会反复开辟 $1 \times N^2$ 大小的寄存器;而在此快速算法中,只需要开辟 $1 \times N$ 的寄

存器,这样就大大降低了资源的占用。

5 结论

本文所研究的快速中值滤波器通过减少比较次数来降低运算量,达到快速运算的目的,并且通过实际仿真得出,这种算法不仅能降低运算次数,而且能大幅降低资源的占用,使得 FPGA 有足够的资源来完成其它的处理任务。这种算法用硬件实现后,能够有效地降低系统的成本,是一种值得推广的快速算法。

参考文献:

- [1] 孙即祥.数字图像处理[M].石家庄:河北教育出版社,1993.
- [2] 罗恒亮,郑 链,王克勇,等.中值滤波在红外成像引信中的应用及硬件实现[J].激光与红外,2004,34(1):43-45.
- [3] Priyadarshan Kolte,Roger Smith,Wen Su. A Fast Median Filter using Altivec[A]. IEEE International Conference on Computer Design (ICCD'99)[C].1999.384-391.
- [4] Donald E,Thomas,Philip R, et al. 硬件描述语言[M].刘明业,蒋敬旗,刁岚松,等译.北京:清华大学出版社,2001.

作者简介:

郑 鹤(1982-),男,江西省婺源人,2003年毕业于南京大学,获学士学位,现为国防科技大学在读硕士研究生,主要从事基于FPGA的硬件设计与开发的研究. Email:stussy1982@msn.com.

王鲁平(1974-),男,山东省泰安人,2003年毕业于国防科技大学,获博士学位,讲师,主要研究方向为图像处理、基于FPGA和DSP的高速信息处理机设计等.

李 飙(1967-),男,浙江省松阳人,1998年毕业于国防科技大学,获博士学位,副教授,主要研究方向为图像处理、视频编码、信号处理等.