

## 空域滤波算法硬件化设计

赵东晓, 桑红石

(华中科技大学 自动化学院 多谱信息处理技术国家级重点实验室, 湖北 武汉 430074)

**摘 要:** 分析了目标识别图像底层预处理的几种常用算法, 提出窗口空域滤波计算模型, 并采用 VerilogHDL 语言设计了空域滤波类算法的实时实现硬件结构. 该结构具有良好可配置性, 用户可以根据自己的需求选择相应的滤波处理方法和计算参数. 目前, 已经完成了相关模块的结构设计、RTL 代码编写、功能验证等数字 IC 前端设计流程. 在 SMIC 0.18  $\mu\text{m}$  CMOS 工艺下完成了电路综合、形式验证, 在 120 MHz 时钟约束下得到运算模块面积约为 1.9  $\text{mm}^2$ , 相当于 114.2 k 个等效门, 功耗为 30 mW. 与公开发表的类似设计相比, 我们的设计具有速率快、功耗低、可配置性强的优点.

**关键词:** 图像滤波; 空域滤波; 可配置性; 延迟线

**中图分类号:** TN402

**文献标识码:** A

**文章编号:** 1000-7180(2014)08-0020-05

## Hardware Design of Spatial Filtering Algorithm

ZHAO Dong-xiao, SANG Hong-shi

(National Key Laboratory of Science and Technology on Multi-spectral Information Processing,  
Institute of Automation, Huazhong University of Science and Technology, Wuhan 430074, China)

**Abstract:** In this paper, we analysis several bottom processing algorithms on target recognition and propose spatial filtering calculation model, and implement a set of real-time hardware architecture for spatial filtering algorithms with VerilogHDL. The advantage of this structure is that it has a good configurability which can make users to choose proper filtering methods and calculated parameters according their needs. At present, we have completed digital IC front-end design process of the module, such as the structure design, RTL coding, function verification, etc. Simultaneously, we completed the work of Circuit synthesis and formality verification with Design Compiler Under the SMIC 0.18  $\mu\text{m}$  CMOS process, and obtained the area and power are 1.9  $\text{mm}^2$  equal to 114.2 kGates, 30 mW respectively at the frequency of 120 MHz. The design has fast speed, low-power, highly configurable advantages compared with the published similar designs.

**Key words:** Image filtering processing; configurability; delayline

### 1 引言

在图像生成和采集过程中都会不同程度地引入各种噪声, 使图像质量变差, 影响后续的自动目标识别效果, 因此, 有必要对图像进行预处理, 如对比度增强、图像锐化、图像平滑、边缘检测、膨胀和腐蚀等像素级的运算<sup>[1]</sup>, 我们将空间域滤波的算法统称为空域滤波<sup>[2]</sup>.

空域滤波算法针对每个像素重复执行大量运算,

具有运算量大、运算规则、并行度高等特点, 采用软件实现耗时久、效率低, 难以满足实时性要求, 因此通常采用硬件进行加速<sup>[3]</sup>. 本文提出的空域滤波结构具有很好的可配置性, 具体体现在: (1) 用户可以根据自己的需求选择卷积、均值滤波、最大值滤波、最小值滤波、中值滤波、二值形态学膨胀、二值形态学腐蚀、灰度形态学膨胀、灰度形态学腐蚀以及图像直通等功能; (2) 对于每一种运算可以根据需要对其参数进行配置.

收稿日期: 2013-10-23; 修回日期: 2013-12-19

基金项目: 总装预研教育部支撑计划重点项目“微型 XXXX 光电子系统集成技术”(625010107)

## 2 窗口空域滤波计算模型

基于窗口的处理操作可用如下计算模型表示:

$$s = F_R((F_S(A(x, y), B(x, y)))$$

$$M(x, y) = F_S(A(x, y), B(x, y)) \quad (1)$$

根据式(1),基于窗口的图像操作只需要指定三个参量:

- 窗口模板:大小与系数矩阵一致;
- 作用于每个像素的基本的标量函数  $F_S$ ;
- 缩减函数  $F_R$ :应用于从输入图像中提取出的窗口,从而产生一个输出结果.

标量函数  $F_S$  通常是一组算术逻辑操作,可以采用一个算术逻辑部件进行执行.即在两个图像之间或一幅给定图像上,重复地执行专门的计算,比如加法、乘法、取绝对值或最大(小)值.基于窗口的操作也需要一个缩减函数  $F_R$ ,该函数将生成的中间结果窗口缩减到一个输出结果.

表 1 中给出了本文设计的硬件处理结构所具有的图像滤波处理算法,其中  $A$  表示输入图像像素,  $B$  表示模板参数.

## 3 硬件实现方案

### 3.1 硬件整体结构

图 1 中输入数据  $A$  为被实时处理图像数据.从

表 1 本设计实现的图像滤波算法

| 操作名称    | $F_S$              | $F_R$       |
|---------|--------------------|-------------|
| 卷积      | $(A \times B) / k$ |             |
| 均值滤波    | $A / n^2$          | $\sum M$    |
| 最大值滤波   | $A$                | $Max(M)$    |
| 最小值滤波   | $A$                | $Max(M)$    |
| 中值滤波    | $A$                | $Median(M)$ |
| 二值形态学膨胀 | $A \text{ and } B$ | $OR(M)$     |
| 二值形态学腐蚀 | $A \text{ or } !B$ | $AND(M)$    |
| 灰度形态学膨胀 | $A + B$            | $Max(M)$    |
| 灰度形态学腐蚀 | $A - B$            | $Min(M)$    |
| 直通模式    | —                  | —           |

功能上面讲,图 1 中包括了  $7 \times 7$  窗口运算和  $5 \times 5$  窗口运算两部分,其中  $5 \times 5$  窗口包含在  $7 \times 7$  窗口中,即:以  $7 \times 7$  窗口的中心像素向四边均匀扩展成  $5 \times 5$  窗口.表中最大值滤波、最小值滤波、中值滤波、灰度形态学膨胀、灰度形态学腐蚀运算用到的为  $5 \times 5$  窗口运算,其他各种运算均为  $7 \times 7$  窗口运算.输入数据  $A$  通过延迟线进行缓存,形成二维工作窗口,与同样大小的模板  $B$  执行窗口运算功能.从结构上面讲,图 1 中包括了连接总线接口与数据路径的中心控制逻辑部分和数据路径两部分:数据路径部分包括了用于实现窗口边界扩展的延迟线部分、标量运算单元、缩减运算单元、输出生成电路单元以及分割运算单元;中心控制逻辑部分的功能是产生数据路径各个单元中所需要的控制信号控制数据流动.

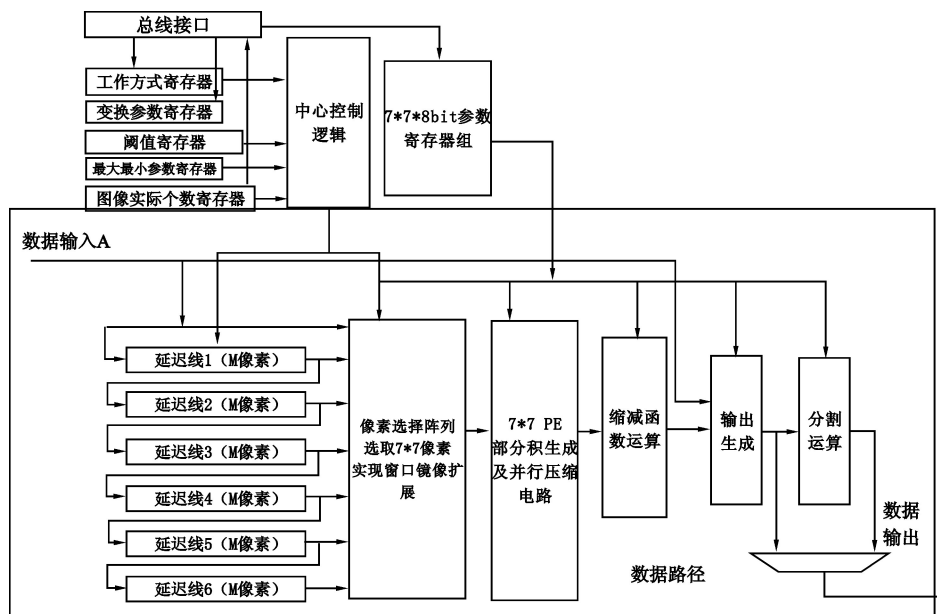


图 1 窗口滤波整体结构图

### 3.2 数据路径各模块设计

空域滤波模块的数据路径由两个任务级流水的运算处理单元:标量运算单元—CSFU 阵列和缩减

运算单元—TN 阵列组成,单个单元结构如图 2 所示.在  $7 \times 7$  窗口中,把标量处理单元得到的 49 个中间结果一起进行  $\&$ 、 $|$ 、卷积运算,在  $5 \times 5$  窗口中,

把标量处理单元得到的 25 个中间结果一起进行求最大值、最小值、中值运算. 其缩减单元内部结构图如下. 其中, OP\_A 为延迟线生成的  $7 \times 7$  大的图像模板, Weights 为对应的  $7 \times 7$  大小的参数模板.

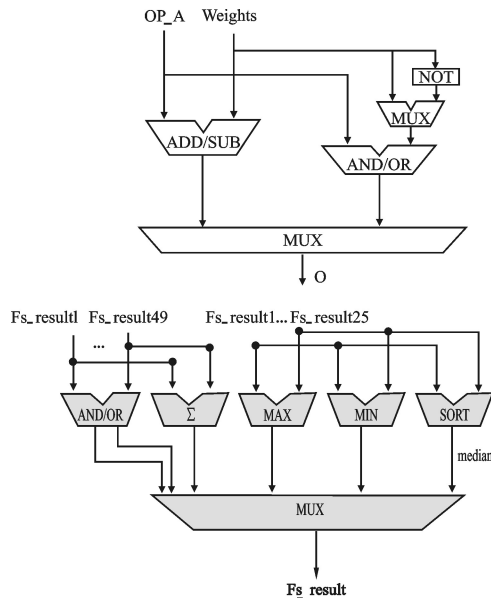


图 2 单个 CFSU 单元和 TN 单元结构结构图

而对于卷积运算, 本设计的处理方式是直接使用 Design Wire 中用于卷积运算的 DW02\_prod\_sum 加法器来实现, 具体实现的运算可用式 (2) 所示.

$$\text{SUM}(k-1:0) = \sum_{j=0}^{N-1} A[(j+1) \times m-1:j \times m] \times B[(j+1) \times n-1:j \times n] \quad (2)$$

式中  $A, B$  为做卷积运算的对象,  $m$  为  $A$  的位宽,  $n$  为  $B$  的位宽,  $N$  为输入数据的个数,  $k$  为得到结果的位宽. 在本设计中, 用于做卷积运算的输入图像数据  $A$  和计算参数  $B$  的位宽分别为 9 bit, 16 bit, 窗口大小为  $7 \times 7$  故输入数据个数  $N=49$ , 结果位宽为 30 bit.

在最大值、最小值、中值滤波运算的  $5 \times 5$  窗口中, 用简单方法求最大值、最小值和中值的排序网络如图 3 所示. 具体处理方式, 对于  $5 \times 5$  窗口, 首先求出窗口中每一行中的最大值、最小值以及中值, 然后在对求出的这 5 个中间结果求最大值、最小值以及中值. 图中, 每一个方框代表一个比较器, 即对于窗口的每一行所要用的比较器个数为 9 个, 而整个  $5 \times 5$  窗口所用到的比较器个数为:  $9 \times 5 + 9 = 54$  个.

对于中值滤波运算, 本设计并不是通过将 25 个元素做大小全排列然后得出中值, 而是先算出每一行的中值, 然后将这 5 个数重新排列得出中值. 我们将这种求中值的近似算法与通过全排列求中值的精确算法通过 matlab 进行了模拟对比实验. 其结果

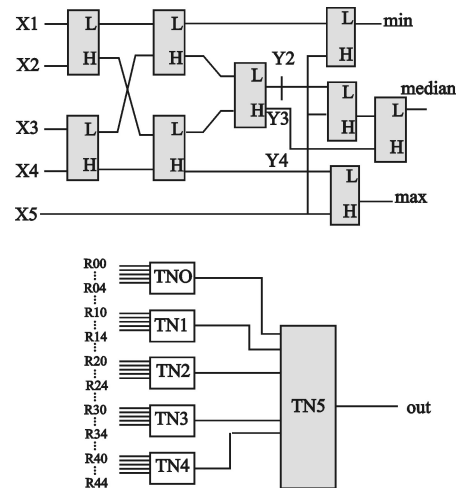
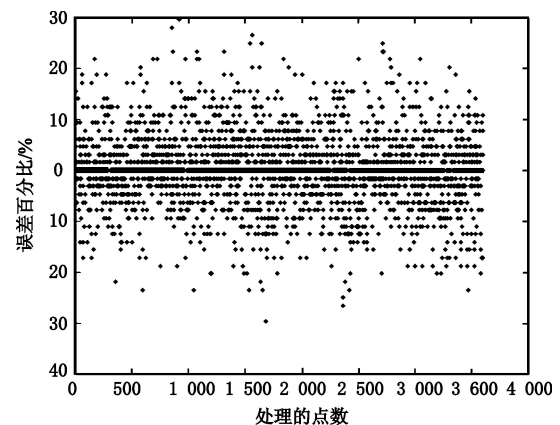


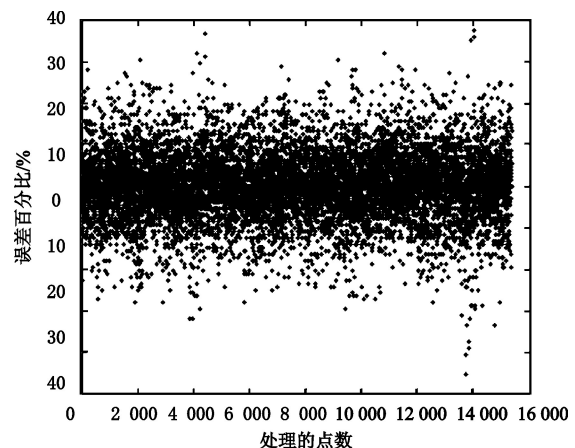
图 3 最大/最小和中值排序网络结构 TN 示意图

表明, 这种近似中值处理方式得到的结果误差主要集中在 20% 之内, 如图 4 所示.

从图 4 分析可知: 这样的近似中值处理方式得出的结果在我们可接受的误差范围之内. 另外, 该处理方式占用的硬件资源为  $9 \times 5 + 9 = 54$  个比较器, 而通过全排列得到中值所需的比较器的个数为  $1 + 2 + 3 + \dots + 24 = 300$  个比较器, 可以计算出, 该近似



(a) 图像大小为  $64 \times 64$



(b) 图像大小为  $128 \times 128$

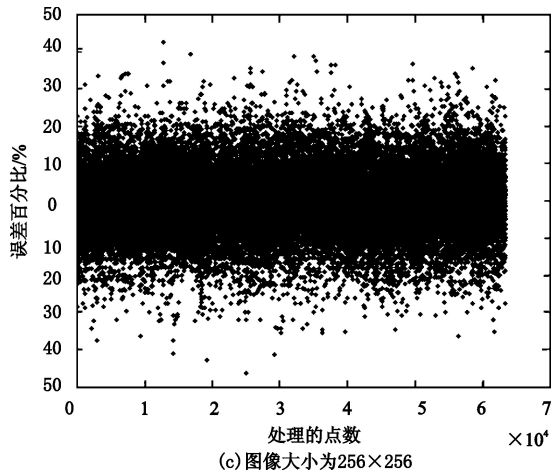
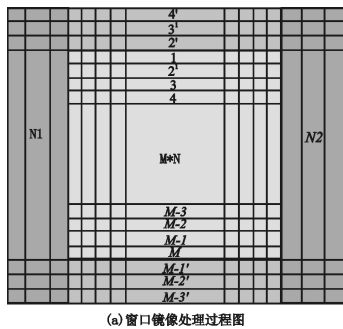


图4 不同图像大小的近似中值算法误差示意图

算法使用的比较器的个数节省了82%,故在硬件资源开销上本算法还是具有很大优势的。

### 3.3 延迟线实现图像边界镜像扩展

空域滤波类操作都是基于模板或窗口的图像处理操作,因此,这类操作也可以被称为基于窗口的处理操作。本设计参照了文献[4]中提到的图像边界镜像扩展方法,具体如下:对一幅  $M \times N$  大小的图像  $C$  进行边界镜像扩展,首先判断窗口的中心像素坐标  $(mid\_row, mid\_col)$  是否处于边界情况;对于  $7 \times 7$  大小的窗口来说边界情况出现在  $mid\_row$  是 1,



2,3,  $M-2$ ,  $M-1$ ,  $M$ ;  $mid\_col$  是 1,2,3,  $N-2$ ,  $N-1$ ,  $N$ . 如果是边界情况,按图 5(a)所示进行扩展. 图中白色区域为对应的原始图像  $C$ , 首先进行行方向上的镜像, 图中所示的  $4'$ ,  $3'$ ,  $2'$  分别是原始图像的第 4,3,2 行镜像得到,  $M-1'$ ,  $M-2'$ ,  $M-3'$  分别是原始图像的  $M-1$ ,  $M-2$ ,  $M-3$  镜像得到, 这样得到了一个  $(M+6) \times N$  大小的图像, 同理, 在对列方向以相同的扩展方式进行镜像扩展, 最终得到一幅  $(M+6) \times (N+6)$  大小的图像  $D$ .

本文通过图 5(b)所示的结构来实现  $7 \times 7$  图像窗口, 其具体设计如下: 对于  $7 \times 7$  的窗口, 需用到 6 个寄存器类型的双口存储器 RFTP1024X9M (存储器的深度为 1 k, 位宽为 9 bit) 作为延迟线来缓存数据. 输入图像像素在进入本行电路的同时也进入下一行的延迟线, 并通过行方向 6 个多路选择器的 10 bit 控制信号以及列方向 7 个多路选择器的 8 bit 控制信号来控制生成  $7 \times 7$  窗口数据. 对于每一行延迟线的存储器, 其读写指针的控制为: 写指针总是提前读指针  $N$  个时钟周期,  $N$  为处理图像的列数. 显然, 就产生了一个数据填充过程, 此期间内, 所有的输出图像数据为无效数据, 有效输出数据的时间延迟大约为  $3N+3$  个时钟周期.

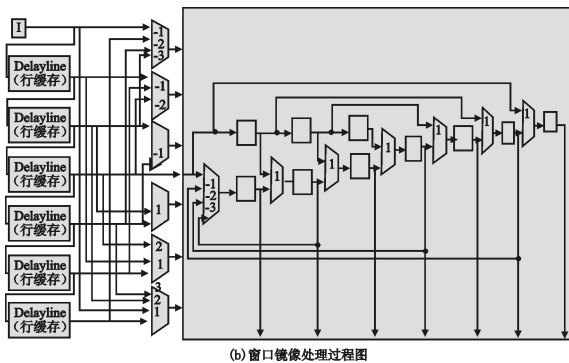


图5 延迟线实现图像边界扩展系统

## 4 仿真和综合结果

本设计 RTL 级的仿真验证方法为搭建 Modelsim 和 Matlab 的联合仿真平台, 在 ModelSim 和 Matlab 里面加载相同的测试激励, 然后将 ModelSim 里面的硬件仿真结果以 Txt 文档的格式反馈到 Matlab 里面做数据自动对比. 具体测试从以下三个方面开展:

(1) 基本运算功能测试: 输入不同大小的图像数据, 硬件电路都能正确实现十三种滤波算法.

(2) 电路非连续工作模式测试: 电路中设置有暂停信号, 当暂停信号有效时, 电路可以暂停工作, 暂停后相应的输出数据有效信号会拉低.

(3) 连续多帧工作模式测试: 电路既能完成单帧运算, 也可以实现连续多帧图像的处理.

在寄存器传输级的仿真验证后, 需要进行逻辑综合, 即在特定的标准单元库、指定的环境和时序约束下将 RTL 级代码转换为门级网表<sup>[5]</sup>. 目前主流的综合工具是 Synopsys Design Compiler, 简称 DC. 本设计运用 DC 工具得到以下关于时序、面积、功耗

三个方面的报告:

时序报告:

```
Timing Path Group 'clk'
Levels of Logic:      21.00
Critical Path Length:  5.31
Critical Path Slack:   0.00
```

在时序的脚本中,将时钟频率设置为 120 MHz 进行综合并得到时序报告,其关键路径 Critical Path Slack 的值为 0,这表明关键路径上没有出现负的 Slack,满足了时序要求。

面积报告:

```
Area
Cell Area:      1 914 183.462 177
Design Area:    1 914 183.462 177
```

功耗报告:

```
Cell Internal Power = 26.200 0 mW (87%)
Net Switching Power = 3.758 8 mW (13%)
Total Dynamic Power = 29.958 8 mW (100%)
```

从面积报告和功耗报告可以看出,空域滤波硬件电路的面积开销大约是  $1.9 \text{ mm}^2$ ,功耗约为 30 mW。

针对本文提出结构的综合结果,我们与文献[6]提出的用于实现空域滤波的三种结构做了相关比较,同时,文献[7]提出的可配置 2D 空域滤波结构也具有卷积、均值滤波、最大值滤波、最小值滤波等滤波功能,我们也进行相应对比,相关比较对象都是在 SMIC  $18\mu\text{m}$  CMOS 工艺下进行电路综合的。需要说明的是,文献[6-7]中并没有对计算参数的格式进行说明,本结构用到的计算参数格式为 16 bit 的有符号数。

通过比较可得出如下结论:(1)对于上述三种不同结构的 FIR 滤波器来说,本文提出的空域滤波硬件结构的主要优势在于速率有了明显提升。在面积开销上,本结构约为 FIR 滤波结构的 5~6 倍,这是由于 FIR 滤波结构的图像窗口大小为  $3 \times 3$ ,本结构图像窗口为  $7 \times 7$ ,后者约为前者的 5 倍大。故在面积上的开销差别不大。(2)对于同样具有可配置性的 2D 空域滤波结构来说,虽然在速率上稍有差距,但本文提出

的硬件结构功耗更低、面积更小。综合来讲,本空域滤波结构在整体性能上是具有优势的。

## 5 结束语

本文提出了一种功能可配置的空域滤波硬件结构,该结构在用户使用层面上具有很好的灵活性,并与其他文献中提出的空域滤波结构对比得出,在速率、面积、功耗三个方面的综合性能权衡下,本结构更具有优势。同时,本结构又可以作为一个独立的 IP 核搭载在 SoC 系统上,在不同的图像处理应用中得到重用。

## 参考文献:

- [1] 方翰华,陈新华,沈国新,等.基于 FPGA 图像滤波算法硬件化设计[J].电子测量与仪器学报,2009,32(12):68-71.
- [2] 何明一,卫保国.数字图像处理[M].北京:科学出版社,2008.
- [3] 邓红梅.基于 SoC 的图像匹配系统设计与实现[D].上海:上海师范大学,2005.
- [4] Donald G Bailey. Image border management for FPGA based filters[J]. IEEE International Symposium on Electronic Design, Test and Application, 2011(6): 144-149.
- [5] 高伟.可配置浮点 FFT 的 ASIC 设计[D].武汉:华中科技大学,2012.
- [6] 勾荣.数字图像空域滤波算法的 FPGA 设计与实现[J].现代电子技术,2007(21):199-201.
- [7] 袁雅婧,桑红石,张天序.可配置 2D 空域滤波操作 VLSI 架构研究[J].微电子学与计算机,2012,29(12): 137-139.

## 作者简介:

赵东晓 男,(1988-),硕士研究生.研究方向为数字集成电路设计.

桑红石 女,(1970-),副教授.研究方向为图像处理和集成电路设计等.