

引用格式: 李红军, 郭阳, 贾润. 归一化积相关算法并行加速方法研究及 FPGA 实现[J]. 微电子学与计算机, 2020, 37(2): 80—86. [LI H J, GUO Y, JIA R. Research of parallel acceleration method and implementation of normalized product correlation with FPGA[J]. Microelectronics & Computer, 2020, 37(2): 80—86.]

归一化积相关算法并行加速方法研究及 FPGA 实现

李红军^{1,2}, 郭 阳¹, 贾 润²

(1 国防科技大学, 湖南 长沙 710025; 2 天津航计算技术研究所, 天津 300308)

摘 要: 归一化积相关算法在地形匹配、目标识别等图像处理中有着广泛的应用, 但由于其计算过程复杂, 计算实时性较差, 限制了它的应用范围, 本文结合 FPGA 在并行计算方面的优势, 提出了一种归一化积相关 16 路并行计算方法及硬件实现架构, 具有精度高、速度快的特点, 并在 FPGA 上进行了实现, 与 PC 机 MATLAB 软件和 DSP 芯片 TS201 相比, 运算速度提升了上千甚至上万倍, 计算精度能够达到 10^{-6} , 能够满足地形匹配及目标识别等图像处理的实时性和计算精度要求。

关键词: 归一化积相关算法; 16 路并行计算; 硬件加速

中图分类号: TP391

文献标识码: A

文章编号: 1000—7180(2020)02—0080—07

Research of parallel acceleration method and implementation of normalized product correlation with FPGA

LI Hong-jun^{1,2}, GUO Yang¹, JIA Run²

(1 National University of Defense Technology, Changsha, 710025, China;

2 Tianjin Jinhang Institute of Computing Technology, Tianjin 300308, China)

Abstract: Normalized product correlation method is widely used in image processing filed, such as terrain matching and target recognition, and has limited application for complex computational process and insufficient real-time performance. In this paper, a 16-channel normalized product correlation parallel computing method and hardware architecture are proposed based on the advantages of FPGA in parallel computing. It is implemented on the FPGA and has the characteristics of high precision and fast speed. Compared with the PC MATLAB software and DSP chip TS201, the computing speed is increased by thousands or even tens of thousands of times, and the computing accuracy can reach 10^{-6} , which can meet the real-time and precision and precision requirements of image processing such as terrain matching and target recognition.

Key words: normalized product correlation; 16-channel parallel computing; hardware acceleration

1 引言

积相关算法是地形匹配、目标识别算法的核心, 具有很高的鲁棒性和精准度, 但是由于积相关运算过程复杂, 而且随着图像模板大小的不断提高, 导致积相关算法的计算量大幅增加, 单靠传统串行执

行的 DSP 软件的运算速度无法满足图像处理的实时性要求, 制约了积相关算法的推广应用. 与此同时, FPGA 技术近年来得到了飞速的发展, 运行速度越来越高, 逻辑、数学运算单元及内部 RAM 块等内部资源则成百倍的增加, 为 FPGA 实现实时图像算法提供了硬件基础, 越来越多的学者开始研究如何

利用 FPGA 加速积相关算法。

文献[1]提出了一种用在图像处理中的归一化互相关 FPGA 加速方法,文献[2]提出了基于多路 FIFO 的图像卷积快速算法,文献[3]提出了一种比较大的核卷积算法的 FPGA 快速实现,文献[4]提出一种可变大小的图像卷积算法,文献[5]利用缓冲替换存储器,进行 FPGA 算法加速,文献[6]基于多端口存储器实现快速的归一化互相关算法,文献[7]基于 FPGA 采用 FFT 架构进行相关算法快速实现,文献[8]将相关算法用在光子相关谱检测上,采用了 FPGA 实现的最多 32 通道的相关器,文献[9]则将 FPGA 加速算法用到了 CNN 上。这些研究为图像处理算法的 FPGA 硬件实现提供了一定的借鉴,但同时也存在图像数据的内存管理不够合理,占用过多 RAM 资源和部分设计并行度不够的问题,限制了图像处理算法 FPGA 硬件计算性能的进一步提升。

考虑到实际应用的积相关算法大图、小图模板为可变大小,本文提出了一种大图小图行列可变的归一化积相关算法的 16 路并行计算方法及硬件实现架构,并对图像数据内存管理和核心运算单元并行计算的难点进行了分析说明和工程实现。

2 归一化积相关积相关算法公式化简

归一化积相关(Normalized product correlation, Nprod)算法[10]能够消除积相关算法的伪匹配,计算过程不受图像本身的灰度比例影响,匹配精

度高,并且能够在低信噪比条件下工作,已被广泛应用。根据理论定义,归一化积相关算法如下。

$$C(u, v) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ((A_{ij} - \bar{A}) \times (B_{u+i, v+j} - \bar{B}_{u,v}))}{\sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (A_{ij} - \bar{A})^2 \times \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (B_{u+i, v+j} - \bar{B}_{u,v})^2}} \quad (1)$$

由公式(1)可以看出,积相关运算中存在着定点求平均、定点开方等运算,利用 FPGA 实现时存在精度损失,且需要先求出参与运算的图像数据的均值,不利于采用流水架构进行计算,为了便于积相关算法的 FPGA 实现,对公式(1)进行化简,过程如下。

$$C(u, v) = (-1)^s \times \sqrt{\frac{(\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ((A_{i,j} - \bar{A}) \times (B_{u+i, v+j} - \bar{B}_{u,v})))^2}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (A_{i,j} - \bar{A})^2 \times \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (B_{u+i, v+j} - \bar{B}_{u,v})^2}}$$

式中, s 为 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ((A_{i,j} - \bar{A}) \times (B_{u+i, v+j} - \bar{B}_{u,v}))$ 的符号位, 0 表示正, 1 表示负。因为:

$$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \bar{A} = (m \times n) \times \bar{A}$$

$$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \bar{B}_{u,v} = (m \times n) \times \bar{B}_{u,v}$$

所以:

$$C(u, v) = (-1)^s \sqrt{\frac{((m \times n) \times \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j} \times B_{u+i, v+j} - \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j} \times \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j})^2}{((m \times n) \times \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j}^2 - (\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j})^2) \times ((m \times n) \times \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j}^2 - (\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j})^2)}} \quad (2)$$

由公式(2)可知,积相关运算被拆分为

$$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{ij} \times B_{u+i, v+j}, \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{ij}^2, \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j}, \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{ij}^2, \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j}^2$$

等乘法、累加、平方运算,将定点除法转换为乘法,将定点开方转换为平方运算,将除法和开方运算各减少至一次处理,极大节省 FPGA 资源,在不影响精度的前提下,更适合 FPGA 采用流水架构实现。

3 归一化积相关算法 16 路列并行硬件加速电路的实现

3.1 多路列并行积相关运算方法

以大图像素点 9×9 , 小图像素点 2×2 , 积相关

列并行运算过程如图 1 所示。单路并行运算过程为,从大图中取出 1 个 2×2 图像切片,与小图进行积相关计算,然后右移 1 列再取出一个切片进行计算,直至计算完毕。为提高计算的并行度,以四路列并行计算为例,同时从大图中取出 4 个切片,与小图进行积相关计算,得到 4 个结果,然后右移 4 列处取出 4 个 2×2 图像切片进行第二次积相关计算,得到 4 个结果,直至计算完毕,相比单路运算,速度提升 4 倍。若 FPGA 内部资源足够多,列并行计算的路数可以更多,则积相关运算的速度会显著提高。本文选取 Xinlinx 公司的 V7 系列 FPGA,在单路积相关硬件加速电路实现方案的基础上提出了一种 16 路列并行积相关硬件加速电路实现方案,并进行了工程

实现.

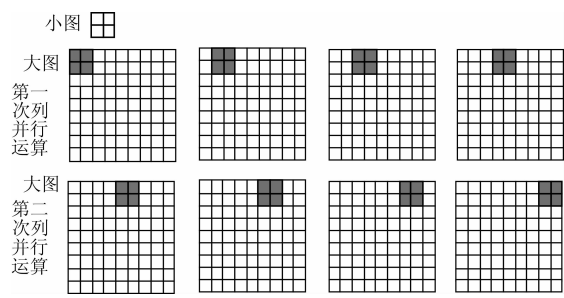


图 1 多路积相关列并行运算示意图

3.2 16 路并行积相关计算硬件实现架构

不同的任务需求需要不同的模板尺寸,因此积相关运算硬件加速电路需要适应不同的模板大小,即大图小图的行列宽度均可配置. 本文根据最大大图的列数确定 FPGA 内部大图列数,根据小图的最大行数确定 FPGA 内部的大图行数,根据小图的最

大行列宽度确定了 FPGA 内部小图的行列数和需要的最大 DSP 乘法器个数. 当实际应用时小图行数小于能实现的最大行数时,需要用数据阵列选择模块将多余行的大图及小图输出切换为 0. 按照公式(2)化简的结果和多路并行计算方法,设计了 16 路列并行积相关硬件加速电路实现架构,如图 2 所示. 要完成 16 路列并行积相关硬件加速电路实现,需要解决两个问题,一个是 16 路列并行积相关运算的 16 路数据源如何产生,另一个是 16 路并行运算模块如何设计. 解决了这两个问题基本上就解决了 16 路列并行积相关运算硬件加速电路的关键. 16 路图像数据管理模块和 16 路积相关并行运算模块分别用来解决上述两个问题,下文中将对它们的实现思路及方法进行描述.

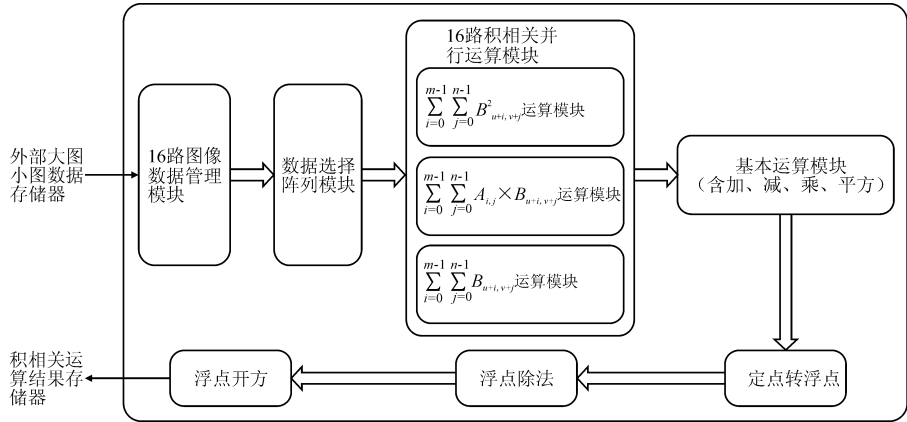


图 2 积相关硬件加速电路单元实现架构图

3.3 16 路图像数据管理模块

16 路图像数据管理模块用于产生归一化积相关 16 路并行硬件加速电路所需要的 16 路大图和小图数据,传统思路上,为了增加多路数据源,只要增加存储器将大图数据多存几份即可,但这样需要更多的 RAM 块资源,在并行路数较少时尚可实现,一旦并行运算路数多到一定程度,调用的 RAM 块超出 FPGA 内部 RAM 资源的上限就不再可行. 为了解决并行路数和 FPGA 内部 RAM 资源之间的矛盾,本文提出了一种 16 路图像数据产生方法,在不增加内部大图存储阵列模块的前提下,提供 16 路并行积相关算法需要的 16 路大图数据源,大大节约了 FPGA 内部有限的 RAM 资源. 16 路图像数据管理模块功能框图如图 3 所示. 从大图数据存储阵列模

块读出的数据直接积相关运算单元 0 到积相关运算单元 15 的大图数据端口相连,从小图数据存储阵列模块读出的小图数据 A_{ij} 以及 A_{ij} 延时 1 拍到 15 拍的数据 $A_{ij_reg1} \sim A_{ij_reg15}$ 依次与积相关运算单元 0~15 的小图数据端口相连. 同样的,将积相关运算控制逻辑直接输出的计算使能信号 en 以及 en 信号延迟一拍到 15 拍的信号 $en_reg1 \sim en_reg15$ 依次与积相关运算单元 0~15 的计算使能信号相连,通过对小图数据存储阵列输出的小图数据以及积相关计算使能控制信号进行 15 级寄存,使对应数据、使能信号同步,达到准备出 16 个数据源的目的.

16 路图像数据管理模块的功能可以划分为小图数据存储阵列模块、大图数据存储阵列模块、小图数据和计算使能信号延时模块组成.

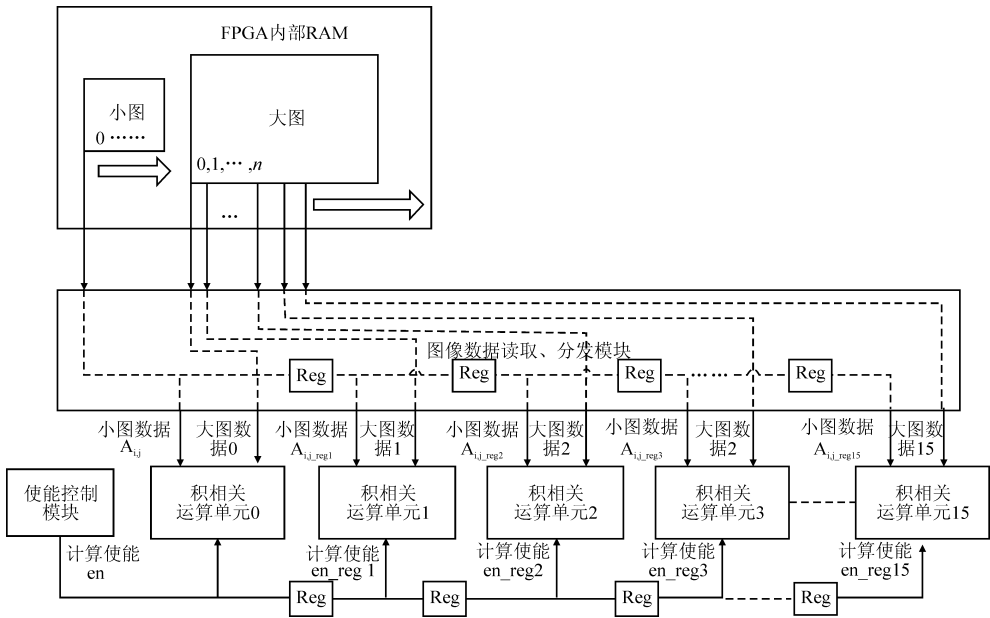


图 3 16 路图像数据管理模块功能示意图

3.3.1 大图和小图数据存储阵列模块

由于本文实现的归一化积相关算法小图最大行列数为 128 行×128 列×8 位,大图最大行列数为 512 行×512 列×8 位,为了实现运算的并行性,本文调用了 128 个 FPGA 内部的 128×8 bit 的双口 RAM 模块,共同组成小图数据存储阵列模块.调用了 128 个 FPGA 内部的 512×8 bit 的双口 RAM 模块,共同组成大图数据内部存储阵列模块.当实际参与积相关运算的大图和小图图像行数 m 小于 128 时,小图数据只存到前 m 行即可,后 128 m 行不使用.大图和小图数据存储阵列模块内部组成框图如图 4 和图 5 所示.

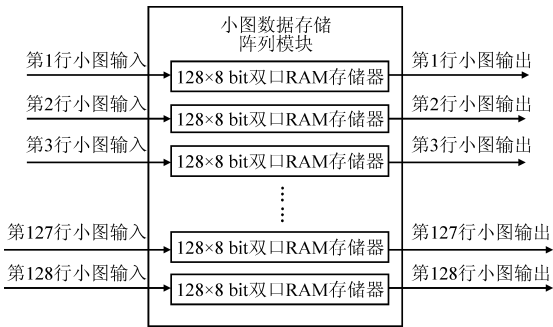


图 4 小图数据存储模块组成框图

3.3.2 小图数据和计算使能信号延时模块

小图数据和计算使能延时信号模块用来实现小图图像数据和计算使能信号 en 的 15 拍延时功能,从而能够为 16 路积相关并行运算单元提供 16 路小图数据源和计算使能信号.由于小图的行数最大为

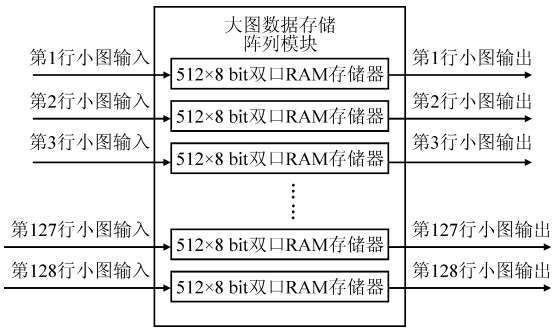


图 5 大图数据存储模块组成框图

128 行,16 路图像数据管理模块输出小图数据最大需要延迟 15 拍,且每个小图图像数据为 8 位,小图数据延时功能共需要 $128 \times 15 \times 8 = 15\,360$ 个寄存器,另外,计算使能信号 en 延时功能需要延时 15 拍,共需要 15 个寄存器,因此,实现小图数据和计算使能信号 15 拍延时模块共需要 15375 个寄存器.小图控制信号模块功能框图如图 6 所示.

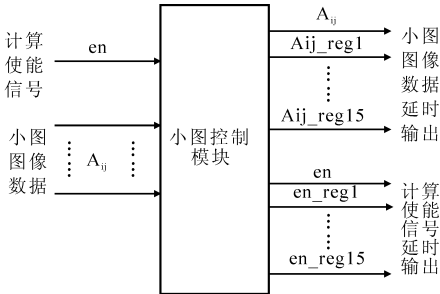


图 6 小图控制信号模块功能框图

因此,16 路图像数据管理模块通过调用一个小图数据存储阵列模块、一个大图数据存储阵列模块和 $128 \times 15 \times 8 = 15360$ 个寄存器实现了 16 路列并行积相关硬件加速电路的 16 路图像数据源的问题. 与重复调用 16 个图像存储阵列相比,极大的节约了 FPGA 内部的 RAM 资源.

3.4 积相关复用计算单元

从归一化积相关运算化简公式(2)中可以看出,式中 $A_{i,j}$ 为小图数据, $B_{u+i,v+j}$ 为大图数据,一旦开始一次积相关运算,小图中所有数据 $A_{i,j}$ 就已经固定,且不会发生变化,因此 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j}$ 、 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j}^2$ 的值就不会发生变化,而 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{ij} \times B_{u+i,v+j}$ 、

$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}$ 和 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}^2$ 的值由于对应不同的大图子图而随着行列变化而发生变化,因此,在进行积相关并行计算时,只需要对 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j} \times B_{u+i,v+j}$ 、 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}$ 和 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}^2$ 的值进行 16

路并行计算即可. 将积相关运算中进行 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j} \times B_{u+i,v+j}$ 、 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}$ 和 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}^2$ 运算的电路模块进行组合,形成积相关复用计算单元,积相关复用计算单元有内部电路功能框图及引脚示意图如图 7、图 8 所示. 由图 7 可以看出积相关复用计算单元由 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}^2$ 运算模块、 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{ij} \times B_{u+i,v+j}$ 运算

模块、 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}$ 运算模块组成.

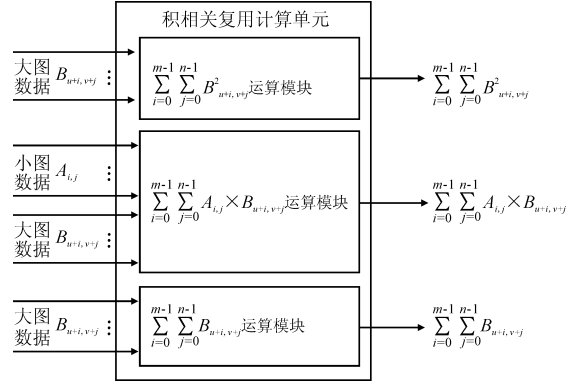


图 7 积相关复用计算单元功能框图

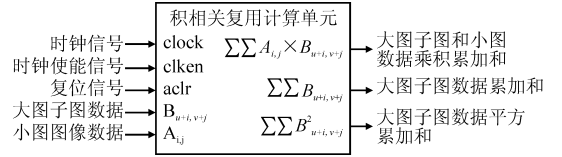


图 8 积相关复用计算单元模块引脚示意图

3.4.1 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}^2$ 运算模块

该运算模块用来实现 128 行大图数据 $B_{u+i,v+j}$ 的平方累加和,该模块由平方阵列运算模块和平方数据累加模块组成. 其中平方阵列模块用来实现 128 行大图数据的平方运算,平方数据累加模块则用来实现 128 个大图数据平方运算结果的累加功能. 功能框图如图 9 所示.

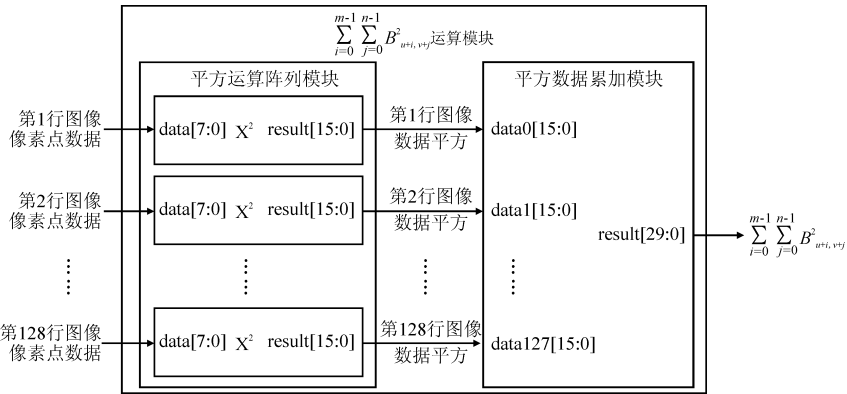


图 9 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i,v+j}^2$ 运算模块功能框图

3.4.2 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{ij} \times B_{u+i,v+j}$ 运算模块

该运算模块用来实现中 128 行大图数据和小图

数据的对应像素点的乘积 $A_{ij} \times B_{u+i,v+j}$ 的累加和,该模块由乘法阵列模块和乘积结果累加模块组成. 其中乘法阵列模块用来实现 128 行大图小图数据的乘

法运算,乘积结果累加模块则用来实现 128 个大图

小图数据乘积的累加功能. 功能框图如图 10 所示.

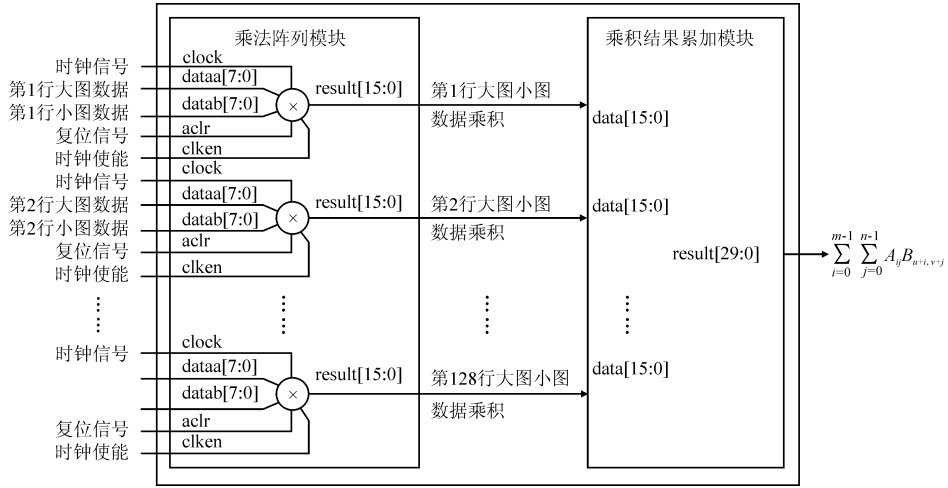


图 10 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{ij} \times B_{u+i, v+j}$ 运算模块功能框图

3.4.3 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j}$ 运算模块

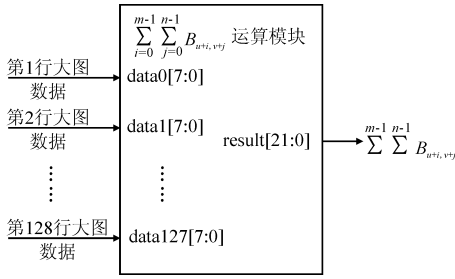


图 11 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j}$ 运算模块功能框图

该运算模块用来实现 128 行大图数据 $B_{u+i, v+j}$ 的累加功能, 功能框图如图 11 所示. data0[7:0] ~ data127 [7:0] 为

$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j}$ 运算模块输入的 128 个大图图像数据, result[21:0] 为输出的 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j}$ 计算结果.

3.5 16 路积相关并行运算模块

重复调用 16 个积相关复用计算单元将构成 16 路积相关并行计算模块, 详见图 12. 将 16 路图像数据管理模块输出的 16 路小图数据、使能信号和大图数据连接到 16 路积相关并行计算模块的端

口, 即可完成 16 个大图子图与小图的 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j} \times$

$B_{u+i, v+j}$ 、 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j}$ 和 $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B_{u+i, v+j}^2$ 列并行计算功能.

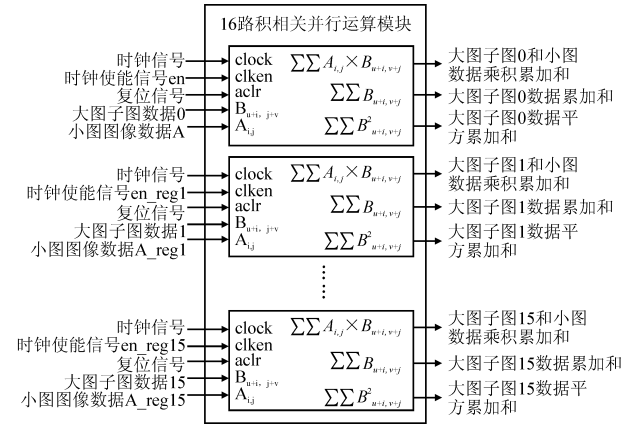


图 12 16 路积相关并行计算模块

4 验结果

4.1 仿真验证

为了说明 16 路并行计算架构的有效性, 进行了功能仿真, 由仿真图可以看出在小图列数 + 15 个时钟周期 (将参与 16 路列并行计算的大图数据读出进行归一化积相关运算所需时间) 内能顺序产生 16 个积相关运算结果, 仿真结果表明该积相关算法 16 路并行计算方案完全可行, 计算速度提升 10 倍以上, 达到了预期的设计目的.

4.2 计算速度及精度试验

为了验证本文实现的归一化积相关并行加速电路的计算性能和计算精度, 本文以实际应用中常用的大图 512×512 、小图 80×80 , 大图 512×512 、小图 128×128 , 大图 320×256 、小图 64×64 三个计算尺寸进行实验, 用 16 路列并行积相关硬件加速电路

进行积相关运算. 记录运算时间进行比对, 如表 1 所示.

表 1 实验时间对比结果

序号	模板大小	TS201 芯片	MATLAB 仿真时间	16 路并行 积相关硬件 加速电路
1	大图 512×512、小图 80×80	546. 136s	17. 4932s	15. 26ms
2	大图 512×512、小图 128×128	689. 822s	29. 6079s	18. 812ms
3	大图 320×256、小图 64×64	114. 586s	3. 2659s	3. 42ms

由表 1 可以看出, 与仅采用哈弗架构的 DSP 实现、以及采用冯诺依曼架构的 PC 机实现相比, FPGA 更适用于并行处理计算, 具有显著的计算效能提升的特点, 实现的 16 路并行积相关算法硬件加速电路, 可以极大的提高积相关运算的速度, 与 PC 机 MATLAB 软件和 DSP 芯片 TS201 相比, 运算速度提升上千甚至上万倍, 完全能够满足实时性的需求.

另外, 通过对归一化积相关公式的化简, 将除法运算和开方运算次数减少为一次, 公式中的其他运算如累加、乘法、平方等定点运算通过扩展数据位宽的方法均可达到精度无损的效果, 极大地减少了精度的损失, 解算结果与 MATLAB 计算结果相比, 精度能够达到 10^{-6} .

5 结束语

本文提出了归一化积相关算法 16 路列并行硬件加速电路的架构以及主要模块的 FPGA 实现方法, 在 Virtex-7 系列 FPGA xc7vx690 上实现了大图 512×512、小图 128×128 模板的 16 路并行处理 FPGA 代码的设计、仿真和调试工作, 模板尺寸根据需求可变. 本文实现的归一化积相关硬件加速电路与 ADI 公司 TS201 软件实现的积相关算法相比, 具有 4 个数量级的速度提升, 与 MATLAB 实现的积相关运算速度相比也提升了近千倍, 且计算精度能够达到 10^{-6} , 能够满足地形匹配及目标识别等图像处理的实时性和计算精度需求, 具有一定的先进性和借鉴意义.

参考文献:

[1] AMILA CABRAL, AJITH PASQUAL. FPGA Implementation of Normalized Cross-Correlation for Real-time Template Matching in Dynamic Search Windows[C]//17th ERU Research Symposium, 2011: 121-124.

[2] ARRIGO BENEDETTI, ANDREA PRATI, NELLO

SCARABOTTOLO. Image Convolution on FPGAs: the Implementation of a Multi- FPGA FIFO Structure [C]//Proceedings of the 24th Euromicro Conference, 1998: 123-130.

[3] JAVIER TOLEDO-MOREO F, JAVIER MARTiNEZ-ALVAREZ J, JAVIER GARRIGóS-GUERRRE F, et al. FPGA-based architecture for the real-time computation of 2-D convolution with large kernel size [J]. Journal of Systems Architecture, 2012, 58(8): 277-285.

[4] VINAY SRIRAM, DAVID KEARNEY. A FPGA implementation of variable kernel convolution[C]//8th International Conference on Parallel and Distributed Computing, Applications and Technologies, 2007: 105-109.

[5] METEB M ALTAF, EBALL H AHMAD, LI W, et al. An ultra- high-speed FPGA based digital correlation processor [C]//IEICE Electronics Express, 2015, 12(8): 1-7.

[6] ALMUDENA LINDOSO, LUIS ENTRENA. High performance FPGA-based image correlation[J]. Journal of Real-Time Image Processing. 2007, 2(4): 223 - 233.

[7] DI CARLO S, GAMBARDELLA G, INDACO M, et al. . An area-efficient 2-D convolution implementation on FPGA for space applications[C]// IEEE 6th International Design and Test Workshop (IDT), 2011: 11-14.

[8] JAKOB C, SCHWARZBACHER A T, HOPPE B, et al. A FPGA Optimised Digital Real-Time Mutichannel Correlator Architecture[C]//10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007), 2007: 35-42.

[9] NNADANA N, DIVYA KIRAN, HARIHARAN RAMASANGU. FPGA Implementation of Efficient Convolution Architecture for CNN[J]. RUAS-SAS Tech Journal, 2017, 17(1): 25-28.

[10] 朱永松, 国澄明. 基于相关系数的相关匹配算法的研究[J]. 信号处理, 2003, 19 (6) : 531-534.

作者简介:

李红军 男, (1980-), 硕士, 高级工程师. 研究方向为计算机架构及嵌入式系统应用.
E-mail: lihongjun8357@163. com.

郭 阳 男, (1971-), 教授, 博士生导师. 研究方向为微处理器设计.

贾 润 男, (1966-), 硕士, 研究员. 研究方向为嵌入式计算机应用及设计.