

引用格式:吴皓月,邓军勇,山蕊,等. 可重构阵列处理器 Harris 算法并行化实现[J]. 微电子学与计算机,2019,36(4):67-71.  
[WU H Y, DENG J Y, SHAN R, et al. Reconfigurable array processor Harris algorithm is implemented in parallel [J]. Micro-  
eletronics & Computer,2019,36(4): 67-71. ]

## 可重构阵列处理器 Harris 算法并行化实现

吴皓月<sup>1</sup>, 邓军勇<sup>1</sup>, 山蕊<sup>1</sup>, 张玉婷<sup>2</sup>, 贺飞龙<sup>2</sup>

(1 西安邮电大学 电子工程学院, 陕西 西安 710121; 2 西安邮电大学 计算机学院, 陕西 西安 710121)

**摘 要:** 本文提出可重构阵列处理器 Harris 并行化的算法映射方式, 其中可重构阵列处理器解决了算法在硬件上修改就需要结构重新调整的缺陷同时簇间并行化解决了算法在软件速度和延时的缺陷. 通过 modelsim、Xilinx 公司硬件设计工具 ISE 和 BEE4 开发平台实现 Harris 算法对分辨率为  $512 \times 512$  的图像映射, 实验结果表明, 整个算法映射时间为 0.143 ms, 这个时间相比于相同条件下 CPU、GPU、FPGA 实现 Harris 算法映射的时间都短.

**关键词:** Harris 算法; 可重构阵列处理器; 并行性;

**中图分类号:** TP302

**文献标识码:** A

**文章编号:** 1000-7180(2019)04-0067-05

## Reconfigurable Array Processor Harris Algorithm is Implemented in Parallel

WU Hao-yue<sup>1</sup>, DENG Jun-yong<sup>1</sup>, SHAN Rui<sup>1</sup>, ZHANG Yu-ting<sup>2</sup>, HE Fei-long<sup>2</sup>

(1 College of Electronic Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China;

2 College of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an 710121, China)

**Abstract:** In this paper, the reconfigurable array processor Harris parallelization algorithm mapping methods, the reconfigurable array processor solves the algorithm on the hardware modification requires the defect of structure readjustment between cluster parallel at the same time solve the defect of the algorithm in software speed and time delay. By modelsim, Xilinx ISE company hardware design tools and BEE4 development platform to realize the Harris algorithm to the resolution of  $512 \times 512$  image map, the experiment results show that the algorithm mapping time of 0.143 ms, under the condition of the time compared to the same CPU, GPU, the FPGA implementation Harris algorithm mapping time is short.

**Key words:** Harris algorithm; reconfigurable array processor; parallelism;

### 1 引言

Harris 算法在图像检测领域越来越受欢迎, 目前在各个领域都可以实现, 例如可以在 CPU、GPU、FPGA 都已经能够实现, Harris 角点检测算法的图像角检测的可配置体系结构的一种可编程的方法目前也已经实现<sup>[1]</sup>. 但是软硬件一直存在算法映射的速度慢

和不能实时更新的问题, 要提高速度可采用并行化实现的方式, 同时每次的微小改动不需要结构重新调整的问题可采用本文提到的可重构的方法.

边缘检测是图像的一种基本工具分割和特征提取<sup>[2]</sup>. 本文的加速对象选用两种特征检测算法, 分别为 Sobel 边缘检测和 Harris 角点检测, 在 SDSoc 环境下对分辨率为  $512 \times 512$  的图像采用不同的优化

收稿日期: 2018-08-08; 修回日期: 2018-09-17

基金项目: 国家自然科学基金(61772417, 61602377, 61634004, 61272120); 国家自然科学基金(61802304); 陕西省国际科技合作计划项目(2018KW-006); 陕西省科技统筹创新工程项目(2016KTZDGY02-04-02); 陕西省重点研发计划(2017GY-060)

方法实现了算法的硬件加速,如流水线优化、软硬件并行、循环展开等. Harris 算法使用可重构阵列处理器和并行化使得图像处理在需求方面可以进行实时调整同时在数据处理的时间上性能大大提升.

2 可重构阵列处理器

目前可重构阵列的处理器已经能够实现其功能. 本文用到的可重构阵列处理器的数据流处理采用三级流水线模式<sup>[3-6]</sup>. 三个时钟周期完成每个指令的工作,分为三个步骤“取值-执行-写回”.

(1)配置模块将指令一次性送入,处理器转换为工作模式,根据地址指令存储器输出对应指令.

(2)数据接收时,PE 处理元可接受相邻 PE 处理元送来的数据和其内部寄存器送出的数据,在逻辑运算单元进行相应的逻辑计算.

(3)数据扇出时,是将数据送给相邻 PE 处理元并同时可发送多个数据,即多个接口都可接收到输出的数据. 每个控制单元产生的控制信号可以使得数据传输正确执行. 有一部分数据来自外部存储,这部分数据可以传输至 PE 处理元的内部寄存器中,也可传输到外部存储.

本文映射的算法总共使用 4 个处理元簇. 根据算法模型的映射需要,总计需要调用 00 簇、01 簇、10 簇和 11 簇四个簇的资源. 每个簇内含有 16 个 PE 处理元,划分为 4 个块,四个簇的地址分别为 00 簇、01 簇、10 簇和 11 簇.

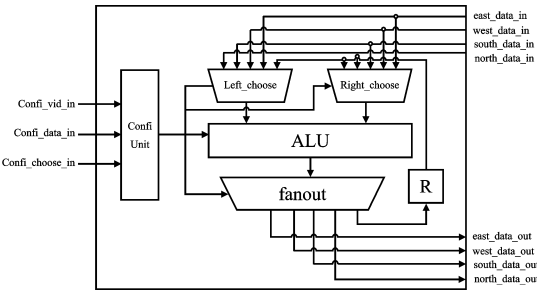


图 1 基于数据流驱动可重构处理单元总体结构图

图 1 是可重构处理单元数据流驱动结构图. 可重构阵列处理器由 ALU、扇出 fanout 单元、右边数据选择单元、左边数据选择单元、内部寄存器单元和配置单元组成. 此阵列处理器单元可以接收内部寄存器邻近的上下左右四个寄存器单元所传输过来的数据,并支持同时传递数据给邻近的四个寄存器单元,这在数据处理速度上有很大优势.

阵列处理器中每个 PE 处理元共有 16 个寄存器,其中有 4 个共享寄存器方向为东西南北寄存器,

也包括 12 个本地寄存器. 当 PE 处理元接受到来自其他处理元的数据时,配置模块将指令会一次性送入,然后根据地址指令存储器输出对应指令,左边数据选择单元和右边数据选择单元就会判断是哪个地址传来的数据,然后把数据传输给 ALU 单元,经过逻辑计算之后传递给扇出 fanout 单元,并同时存至其内部寄存器单元,继续传递给下一个 PE 处理元.

3 Harris 并行化设计

3.1 Sobel 算法

Sobel 图像处理算子是由两个矩阵构成:分别为横向梯度和纵向梯度,算子的图像处理方法是通过对像素位置影响的加权,从而更好的降低图像的边缘模糊度,最终拥有更好的图像处理效果. 本文使用的图像角点检测的 Harris 算法中,Sobel 算子的横向梯度分量  $I_x$  和纵向梯度分量  $I_y$  需要被用到.

|    |   |   |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

$I_x$

|    |    |    |
|----|----|----|
| 1  | 2  | 1  |
| 0  | 0  | 0  |
| -1 | -2 | -1 |

$I_y$

图 2 Sobel 两个方向的算子

公式中  $A$  代表原始图像矩阵,其中  $G_x$  及  $G_y$  分别代表经过横向及纵向边缘检测后的图像灰度值<sup>[7]</sup>,其公式如下:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \tag{1}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \tag{2}$$

可以用以下公式来计算图像中的每点梯度值大小,若得到的某点梯度  $G$  大于给定阈值,那么则认为该点为边缘点.

$$|G| = |G_x| + |G_y| \tag{3}$$

3.2 Harris 算法

Harris 算法是图像处理方面很受欢迎的角点检测算法,它的优点是计算过程简单、稳定性强. 能够实现 Harris 角点检测算法的方法有:软件编程实现和硬件加速实现. 但是软件编程处理图像的速度慢,硬件在对图像处理的加速灵活性不高,实现的结构较复杂. 对于 Harris 角点检测算法,数据密集型应用的这种特征目前应采用并行化或阵列处理器实

现,以解决数据量大的问题.本文实现了在一种可重构阵列处理器上对 Harris 角点检测算法实现并行化映射,通过多个处理簇的并行执行,以实现 Harris 角点检测算法的加速.

计算角点的过程基于如下所示  $2 \times 2$  的矩阵  $\mathbf{B}^{[4]}$ :

$$\mathbf{M} = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (4)$$

公式4中  $I_x$  和  $I_y$  为图像矩阵分别在  $x$  方向和  $y$  方向上的导数,再经过计算得到矩阵  $\mathbf{M}$ . 然后通过  $\mathbf{M}$  的响应函数进一步判断当前点是否为角点. 响应函数中的  $\det(\mathbf{M})$  为矩阵  $\mathbf{M}$  的行列式,  $\text{trace}(\mathbf{M})$  为矩阵  $\mathbf{M}$  的迹,  $k$  是一个可调整的参数,其取值范围在  $0.04 \sim 0.06$ , 本文经过最优映射选择之后,最终确定  $k$  的取值为  $0.04$ .

$$R(\mathbf{M}) = \det(\mathbf{M}) - k * \text{trace}(\mathbf{M})^2 \quad (5)$$

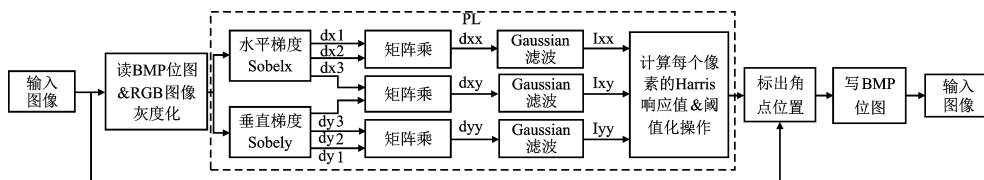


图3 Harris 算法加速器流程

### 3.4 并行化设计

根据 Harris 算法的可重构阵列处理器并行化实现,对图像角点检测的性能,图4是算法阵列处理器映射流程.

共调用四个簇资源:00簇负责进行差分算子滤波和高斯滤波;01簇负责处理 Sobel 算法后的数据、局部非极大值抑制和角点判断;10簇充当外部存储器,11簇起到01簇和10簇之间数据传递作用.下面是各簇工作状态:

(1)00簇负责进行差分算子滤波的计算,共计调用了10个PE处理元.可将10个PE分为两组:载入和运算组;数据写回组与数据发送组.载入和运算组:负责从存储器中读取数据并完成横向梯度分量  $I_x$  和纵向梯度分量  $I_y$  的计算.共调用10个PE:00、01、02、03、10、11、12、13、23、33,00、01、02、03、10、11、12、13、23、九个PE负责计算横向梯度分量和纵向梯度分量;33号PE负责接受结果并存至10簇的外部存储器中.

数据写回组:负责计算  $I_x^2$ 、 $I_y^2$  和  $I_{xy}$  并将计算结果写回到存储器中.共调用9个PE:00、01、02、03、10、11、12、13和23.这九个PE负责计算,将计算结果发送给PE33并写回到10簇的外部存储器中.

### 3.3 Harris 算法加速器

图3是 Harris 角点检测算法的加速过程,可由以下几个步骤来实现:

(1)使用 matlab 工具对  $512 \times 512$  的图像进行处理,使其生成灰度图像的数据矩阵.

(2)对图像使用 Sobel 卷积操作,得到每个像素点分别在  $x$ 、 $y$  方向上的梯度值  $I_x$  和  $I_y$ .

(3)分别对梯度值  $I_x$  和  $I_y$  使用乘法操作,可以得出  $dxx$ 、 $dxy$  和  $dyy$ .

(4)对矩阵  $dxx$ 、 $dxy$  和  $dyy$  分别再使用高斯滤波可以进一步得到  $I_{xx}$ 、 $I_{xy}$  和  $I_{yy}$ .

(5)经过响应函数判断每个点是否为角点,标出角点位置,生成最终数据矩阵,由 matlab 工具对矩阵进行操作,把数据输出为图像.

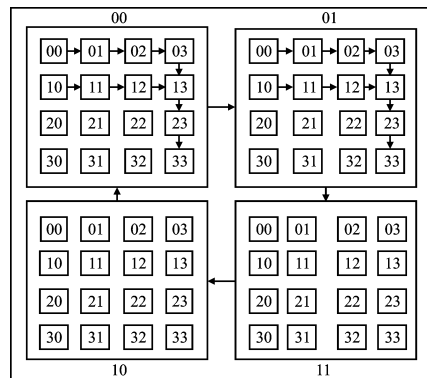


图4 Harris 角点检测算法整体映射结构

数据发送组:负责对发送数据信号进行判定,当信号有效时将当前簇的数据发送到下一个簇进行运算.共调用1个PE33. PE33负责在数据发送信号有效后,将00簇计算出的  $I_x^2$ 、 $I_y^2$  和  $I_{xy}$  发送给01簇.

(2)00簇第二轮负责进行高斯平滑去噪,共调用10个PE,可将10个PE分为两组:数据准备和运算组、数据写回及发送组.

数据准备与运算组:负责从存储器中读取  $I_x^2$ 、 $I_y^2$  和  $I_{xy}$  的数据进行高斯滤波.共调用00簇的10个PE:00、01、02、03、10、11、12、13、23和33.每次从数据矩阵拿取9个数分别放入00、01、02、03、10、

11、12、13 和 23 号 PE 进行高斯滤波生成新矩阵的一个数,计算后的数据放入 PE33,然后由 PE33 存入 10 簇里面。

数据写回及发送组:负责将数据运算组的运算结果写回到存储器并发送给下一簇进行角点量的计算.调用 00 簇的 PE33. PE33 在接收到工作信号后将运算结果发送至下一个簇中。

角点量计算组:负责从存储器中读取高斯平滑去噪后得到的  $I_x^2, I_y^2$  和  $I_{xy}$  并依据角点响应函数  $R = \det(M) - K(\text{trace } M)^2$  计算角点量.调用 01 簇所有 PE. 01 簇需要对高斯滤波后的数据进行角点判断,一次拿取 15 个数分别在除了 33 之外的其他 PE 使用公式  $I_x^2 \times I_y^2 - (I_{xy})^2 - k(I_x^2 + I_y^2)^2$  进行计算判断,在计算,并将计算结果发送至 PE33;完成角点量计算。

局部非极大值抑制组:负责写回计算出的角点量,并进行局部非极大值抑制.调用 01 簇 PE33,该 PE 在接收到来自其他 PE 的数据后,先写回至存储器中,再与已有的极大值进行比较,当新的数据大于当前存储的极大值时,将新数据存储为极大值,否则极大值数据不变.在完成所有判断后,给出角点判断

组工作信号。

角点判断组:负责对计算完毕的角点量进行判断,调用 01 簇的所有 16 个 PE;除了 PE33 的其他 PE 进行与运算后进行条件判断.当运算结果为 1 时,判定该点为角点,向存储器中写入 255;当运算结果为 0 时,判定改点不为角点,向存储器中写入 0.至此,完成对 Harris 角点检测算法的并行化映射方案的描述。

(3)01 簇主要负责进行角点量的计算和局部非极大值抑制,并进行角点判断.为在描述时可与寄存器区别,将角点量矩阵在定义为矩阵 A。

4 仿真与分析

4.1 验证策略

(1)本次设计中仿真使用的 modelsim 工具用于对汇编翻译后的二进制代码在 verilog 中进行基本功能验证.(2)综合工具使用 Xilinx 公司的硬件设计工具 ISE 进行综合生成门级网表.(3)数据生成使用 BEE4FPGA 开发平台.(4)使用 matlab 工具处理实现图像转为数据矩阵和最终数据生成后转化为图像。

4.2 验证仿真结果

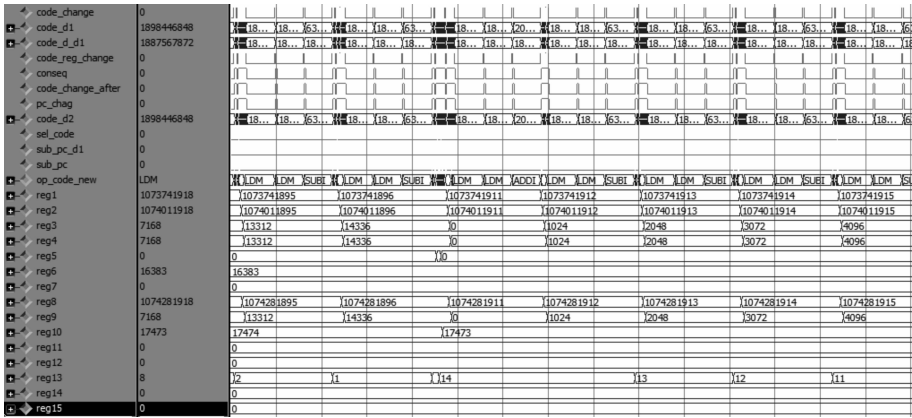


图 5 阈值检测 PE33 的仿真结果

各个 PE 计算的值存储在寄存器里面,16 个 PE,一次检测判断 15 个数是否角点,由 PE33 把最后的值存储到阵列处理器外部存储器的 10 簇中,总共  $512 \times 512$  个数,总共循环 17 477 次,图中 reg10 寄存器就是存的循环轮数,直到 reg10 的数变为 0,所有数判断结束并同时存储到存储器中。

4.3 实验结果

实验结果如图 6 所示,经过检测 k 值为 0.04 时图像角点检测映射结果最好,由 Harris 角点检测映射结果图像可看出,本文提出的可重构阵列处理器并行化映射方法结果正确,同时该结果表示此算法

硬件实现有良好的边缘和角点检测效果。

4.4 性能分析

将文献[8][9]都采用 FPGA 对 harris 算法进行了并行化实现,明显看出相同尺寸下本文速度要快得多;不同尺寸的 FPGA 和本文的四个簇的处理时间比较相差不多.与文献[10]中的专用硬件实现相比,本文进行的处理时间大于专用硬件的时间,但目前采用 4 个处理元簇进行映射,在未来进行存储数据通路优化之后,采用 8 个或者更多个处理元簇进行并行化实现后,处理速度有望接近于专用硬件的速度.而与文献[9]中的 CPU 和 GPU 相比,本文

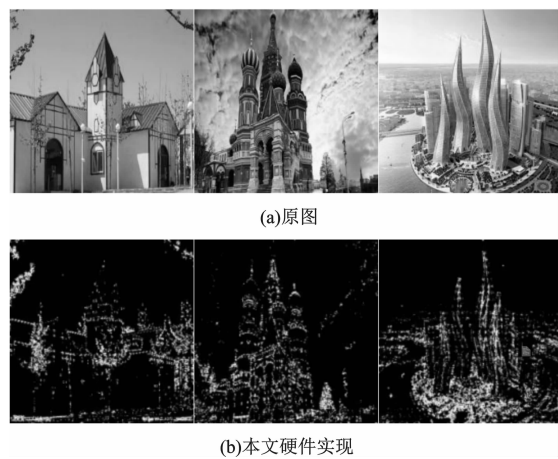


图 6 Harris 角点检测结果

处理速度要比其快得如表 1 所示给出了不同实现下的处理结果对比情况.

表 1 Harris 执行时间对比结果

| Implementation | Size          | Frequency(MHZ) | Cost(ms) |
|----------------|---------------|----------------|----------|
| FPGA[8]        | 1 024 * 1 024 | 225            | 4. 367   |
| FPGA[9]        | 512 * 512     | 232            | 1. 15    |
| GPU[9]         | 512 * 512     | —              | 2. 32    |
| 专用硬件[10]       | 1 024 * 1 024 | 65             | 0. 067   |
| CPU[9]         | 512 * 512     | —              | 20       |
| 本文             | 512 * 512     | 78             | 0. 143   |

5 结束语

针对角点检测 Harris 算法,提出了一种在阵列处理器上实现可重构的图像处理的方法. 首先对分辨率为 512 \* 512 的图像进行 Sobel 算法处理和高斯滤波去噪处理,然后利用四个簇并行性化对数据进行处理,从而提高数据处理速度. 当有需求改变时或者发现不符合功能,不需要重新规划结构,可随时进行修改体现了可重构的优势. 实验结果表明,本文采用 4 个处理元簇进行映射,速度只有比专用硬件速度低,以后采用 8 个或者更多处理元簇的时候,速度有望接近专用硬件. 无论是 CPU、GPU 还是 FPGA 实现 Harris 算法映射速度,在图像角点检测正确的前提下,本文提出的可重构阵列处理器算法映射都相对其较快.

参考文献:

[1] LI J, VIITANEN T, LI L, et al. Design and implementation of a multi-mode harris corner detector architecture[C]// IEEE Nordic Circuits and Systems Conference, Linköping: IEEE, 2017:1-6.

[2] ONAT E. FPGA implementation of real time video signal processing using Sobel, Robert, Prewitt and Laplacian filters[C]// Signal Processing and Communications Applications Conference, Antalya: IEEE, 2017:1-4.

[3] LIU S, LYU C, LIU Y, et al. Real-time implementation of harris corner detection system based on FPGA [C]// IEEE International Conference on Real-Time Computing and Robotics. Okinawa: IEEE, 2018: 339-343.

[4] CHAO T L, WONG K H. An efficient FPGA implementation of the Harris corner feature detector[C]// Iap International Conference on Machine Vision Applications. Tokyo: IEEE, 2015:89-93.

[5] JASANI B A, LAM S K, MEHER P K, et al. Threshold-Guided Design and Optimization for Harris Corner Detector Architecture[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2017, 6 (99):1-1.

[6] 山蕊, 李涛, 蒋林. 动态可重构阵列处理器数据流处理单元的设计与实现[J]. 微电子学与计算机, 2017, 34(1):106-109.

[7] 谭媛, 黄辉先, 徐建闽, 等. 基于改进 Sobel 算子的遥感图像道路边缘检测方法[J]. 国土资源遥感, 2016, 28(3):7-11.

[8] AMARICAI A, GAVRILIU C E, BONCALO O. An FPGA sliding window-based architecture harris corner detector[C]// International Conference on Field Programmable Logic and Applications. Munich: IEEE, 2014:1-4.

[9] POSSA P R, MAHMOUDI S A, HARB N, et al. A Multi-Resolution FPGA-Based Architecture for Real-Time Edge and Corner Detection[J]. IEEE Transactions on Computers, 2014, 63(10):2376-2388.

[10] LAM S K, BIJARNIYA R K, WU M. Lowering dynamic power in stream-based harris corner detection architecture[C]// International Conference on Field Programmable Technology. Melbourne: IEEE, 2017: 176-182.

作者简介:

吴皓月 女,(1995-),硕士研究生. 研究方向为微电子学与固体电子学. E-mail:why1484132896@163. com.

邓军勇 男,(1981-),博士,副教授. 研究方向为集成电路设计.

山蕊 女,(1986-),博士,讲师. 研究方向为集成电路设计.