文章编号：2095-2163(2021)08-0006-06 中图分类号：TP391.4;TS107 文献标志码：A

基于 FPGA 的模板匹配加速器的设计与实现

李 锋，周仕杰
（东华大学 计算机科学与技术学院，上海 201620）

摘 要：本文针对当前基于计算机视觉的织物瑕疵检测系统在实时性及经济性上无法满足实际生产需求这一问题，对瑕疵检测领域的模板匹配算法进行了改进、设计并实现了一种基于 FPGA 的模板匹配算法加速器。为了提升加速器的工作效率，深入分析了该加速器架构的处理时延，并从访存时延、传输时延、计算时延 3 个方面对加速器进行了优化，提升了总线带宽的利用率。实验结果表明，该加速器使得传统的模板匹配算法在时钟频率为 150 HMz 的 Zynq-7000 平台上获得了 33 MHz 的像素处理速率，即每秒可处理分辨率约为8 192×4 096大小的织物图片，与通用 CPU i7-8750H 相比，性能是其 10.5 倍，满足了实时性需求。同时该解决方案采用 SoC 技术取代了传统的 PC 级板卡式结构，降低了系统成本，应用前景广阔。

关键词：织物瑕疵检测；模板匹配；FPGA；算法加速器；SoC

Design and implementation of template matching accelerator based on FPGA

LI Feng, ZHOU Shijie
(College of Computer Science and Technology, Donghua University, Shanghai 201620, China)

【Abstract】 In order to solve the problem that the current fabric defect detection system based on computer vision fail to meet the demands of manufacturing process in terms of real-time and economic performance, the template matching algorithm in defect detection field was improved, and a template matching algorithm accelerator based on FPGA was designed and implemented. In order to improve the working efficiency of the accelerator, the processing delay of the accelerator architecture is deeply analyzed, and the accelerator is optimized from three aspects: access delay, transmission delay and computation delay, which improves the utilization of the bus bandwidth. Experimental result shows that the accelerator enables traditional template matching algorithms to achieve a pixel processing rate of 33 MHz on the Zynq-7000 platform with a clock rate of 150 MHz, that is, the accelerator can process a fabric image with the resolution of 8 192×4 096 per second. Compared with i7-8750H, the performance is 10.5 times higher than that of i7-8750H, which meets the real-time performance requirements. At the same time, the solution uses SoC technology to replace the traditional PC board structure, which reduces the cost of the system and has a broad application prospect.

【Key words】 fabric defect detection; template matching; FPGA; algorithm accelerator; SoC

0 引 言

织物的瑕疵检测是织物质量控制中的重要一环，传统的人工验布方式存在着检测效率低下、漏检率高等缺陷，还会对工人本身的视力造成伤害，是提升织物生产效率的一大瓶颈[1]。为了改变这一现状，织物的瑕疵检测成为计算视觉领域的一大研究热点[2]。国内外已有不少基于计算机视觉技术的解决方案，但是这些方案在实时性、准确性、经济性及适用性上还不足以满足实际的生产需求，还需要进一步探索。

基于深度学习的方法可以很好地拟合大量非线性数据，在复杂的织物纹理背景下有着更好的检测效果，但是其面临数据集获取困难、难以增量学习、不可解释性等难题[3-4]。纯色织物的纹理是具有周期性的，传统的模板匹配的方式可以很好地增强瑕疵区域的显著性[5]。但是模板匹配算法本身计算量庞大，当前的通用 CPU 架构无法满足实时计算需求。而 FPGA 作为一种高性能、低功耗的可编程芯片，可以通过编程直接生成专用电路。与 CPU 分时同步的并行方式不同，FPGA 利用电路的并行特性可实现真正的多核并行[6]。现有解决方案大多是将织物瑕疵检测算法部署在高性能服务器上，这些服务器大多采用通用 PC 外接图像采集卡及图形加速卡的结构，成本较高[7]。近年来，随着 SoC 技术不断成熟，FPGA+ARM 的异构平台（Zynq-7000 系列）的推出，使得高性能且低成本的软硬协同定制计算，逐渐成为织物瑕疵检测系统的首选解决方案[8]。

本文首先分析了运用于瑕疵检测领域的模板匹

作者简介：李 锋（1969-），男，博士，教授，硕士生导师，主要研究方向：人工智能、嵌入式系统；周仕杰（1996-），男，硕士研究生，主要研究方向：图像处理、嵌入式系统、FPGA。

收稿日期：2021-05-06

配算法，根据该算法的特点设计了相应的基于 FPGA 的硬件加速器，并对加速器的访存时延、传输时延、计算时延等环节进行了优化。本方案在 Zynq -7020 平台上实现，对该加速器进行了性能评估，并与通用 CPU 的运行效果进行了对比。

1　瑕疵检测中的模板匹配算法

目前基于计算机视觉的瑕疵检测算法可大致分为 4 类：基于结构的方法、基于统计的方法、基于频谱的方法和基于机器学习的方法[9]。其中，基于结构的方法通常是将织物的纹理视为纹理基元的组合，织物图案的纹理即为纹理基元的周期性排列。由于瑕疵会破坏织物纹理的周期性，可以使用无瑕疵的织物图像作为模板，通过模板匹配算法将织物图片与纹理模板相减，可以提取出拥有较好一致性与完整性的目标瑕疵，该方法可用于织物瑕疵图像的显著性检测，织物瑕疵图像的显著性检测就是将瑕疵区域定义为显著区域，其余纹理图案部分定义为低显著性区域[10-11]。

比较常见的基于灰度的图像匹配算法有平均绝对差算法（MAD）、绝对误差和算法（SAD）、误差平方和算法（SSD）、平均误差平方和算法（MSD）、归一化积相关算法（NCC）、序贯相似性检测算法（SSDA）等[12]。由于 SAD 方法计算较为简单，适合 FPGA 的实现，本文采用 SAD 方法，基本原理如式（1）和（2）：

$$d(i,j)=\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}|t(i+x,j+y)-s(x,y)|, \quad (1)$$

$$I=\mathrm{t}_{ij}(x,y), d(i,j)=\min\{d_1,d_2,d_3,\cdots,d_k\}\ k=(M-m)\times(N-n). \quad (2)$$

其中，$s(x,y)$ 为待检测图像，其分辨率为 $m\times n$，$t(x,y)$ 为模板图像，其分辨率为 $M\times N$，$M>m$，$N>n$。

将待检测图像在模板图像上滑动，如图 1 所示。$t_{ij}(x,y)$ 为待检测图片覆盖到的模板图像区域，即子模板图，i、j 为待检测图左上角在模板图像中的坐标位置，其中 i、j 的范围为：$0\leqslant i<M-m$，$0\leqslant j<N-n$。

绝对误差和算法（SAD）将子模板图与待检测图像之间像素灰度的差的绝对值之和 $d(i,j)$ 作为子模板图与待检测图之间的相似度，绝对差之和越小，表示待检测图与该位置的子模板图越相似，取使得 $d(i,j)$ 最小的子模板图 $t_{ij}(x,y)$ 作为模板匹配的结果图像 I，计算公式如式（1）和（2）所示。然后将待检测图像与模板匹配的结果图像相减，定义为该待检测织物图像的显著性图像。

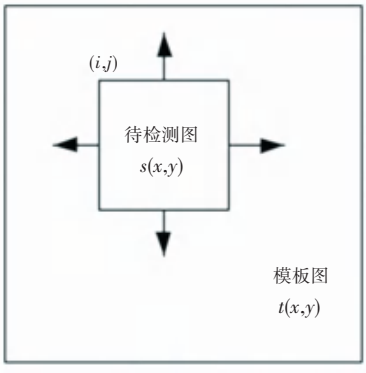

图 1　模板匹配过程

Fig. 1　Template matching procedure

值得注意的是 $M-m$ 与 $N-n$ 的取值，即待检测图像与模板图像之间的大小关系。由于织物纹理存在周期性，将织物纹理抽象为一个正弦函数：

$$y=\sin(x), x\in[0,+\infty), \quad (3)$$

假设待检测图像所包含的纹理周期为：

$$y=\sin(x+\varphi_1), x\in[0,\theta_1], \quad (4)$$

模板图像所包含的纹理周期为：

$$y=\sin(x+\varphi_0), x\in[0,\theta_0]. \quad (5)$$

若 $\varphi_0\neq\varphi_1$，则待检测图与模板图之间存在相位差，为了保证待检测图像是模板图像的子图，则须满足 $\theta_0\geqslant\theta_1+2\pi$，即模板图像须在横向及纵向上均比待检测图像大一个纹理周期。若织物纹理一个周期的像素大小为 $p\times q$，则为一张分辨率为 $m\times n$ 的待检测图像进行模板匹配运算，需进行 $m\times n\times p\times q$ 次减法运算，该方法的计算量庞大，当前的通用 CPU 无法满足其在实时场景中的算力需求，需要设计加速器。

2　基于 FPGA 的模板匹配加速器设计

2.1　从模板匹配算法到 FPGA 加速器

模板匹配是提取显著性图像的重要步骤，其速度直接影响整个瑕疵检测的实时性，为此设计了一个模板匹配加速器，作为一个外设挂载在操作系统之下，CPU 和加速器之间通过内部数据总线相连。此外加速器访问内存需要消耗大量时钟周期，因此被读入片上缓存的数据要尽可能得到复用，但片上缓存的大小有限，一帧 4k 的完整织物图片无法直接存入片上缓存，因此每次将包含若干纹理周期的子图存入片上缓存。本文的论述主要基于像素大小为 16×16 的子图，以及像素大小为 24×24 的模板图，该方案适用于纹理周期小于 8×8 的纯色织物图片。

针对纹理周期较大的情况,可以考虑增大模板图,但这也意味着更大的计算量、更多的资源消耗。本文设计的加速器在每次启动的时候先将模板图像存入片上缓存,然后在完整的织物图像中依次为每一个子图执行模板匹配运算,并将结果不断写回内存,其工作流程如图 2 所示。

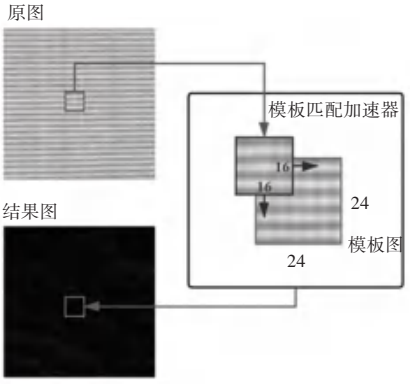

图 2 加速器工作流程

Fig. 2 Accelerator workflow

2.2 加速器结构

加速器采用了 3 层存储架构,分别为片外缓存(DDR),片上缓存(BRAM),以及运算单元内部的寄存器,如图 3 所示。加速器通过 AXI4(Advanced eXtensible Interface 4)总线与 PS(Processing System)的内存通信,两个 AXI4 接口均工作在 master 模式下,可对内存进行随机访问,其中 input 接口负责将内存中的数据写入片上缓存,而 output 接口负责将计算后的结果写回内存。此外该加速器的多个控制寄存器通过 AXI4-Lite 总线实现与内存统一编址,该接口工作在 slave 模式下,PS 端可通过 AXI4-Lite 总线来获取并控制 PL(Programmable Logic)部分的加速器工作状态。

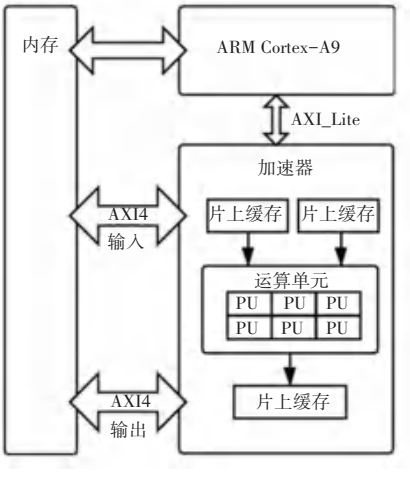

图 3 模板匹配加速器结构

Fig. 3 The structure of template matching accelerator

该加速器的主要运行时延由 3 部分组成,分别为访存时延,即加速器通过 AXI 总线在内存中随机寻址所消耗的时间;传输时延,即数据在 AXI4 总线上传输所消耗的时间;以及计算时延,即模板匹配运算本身所消耗的时间。因此,可以从这 3 个方面对加速器进行优化设计,提高资源利用率,提升加速器的算力。

2.3 加速器时延优化策略

2.3.1 访存时延的优化

访存时延的优化主要通过 AXI4 总线的突发传输机制来实现。AXI4 总线中的突发传输是指在地址总线上进行一次地址传输后,可连续进行多次数据传输,即第一次地址传输中的地址作为起始地址,后续数据的存储地址在起始地址的基础上递增,AXI4 总线的最大突发传输长度为 256[13]。得益于这一机制,在加速器顺序访问大量连续内存地址的过程中,其速率可近似为每个时钟周期访问一个内存数据。对于本文的模板匹配算法来说,每一次访存即是从一张完整的图片中选取一个 16×16 的子图,也意味着每次只有 16 个数据在内存中是顺序排列的,突发传输的长度被限制为 16。实验结果显示,在突发传输长度为 16,时钟频率为 150 Mhz 的情况下,传输一张分辨率为1 024×1 024 的灰度图耗时约 31 ms,如果选取的子图扩大为 32×32,则突发传输的长度扩大为 32,同样传输一张1 024×1 024的灰度图,则耗时约为 19 ms。理论上子图宽度越大,访存时延越短,但是过大的子图纹理难以与模板图中的纹理对齐,获得的检测效果也越差;在不丢失图片几何特征的情况下,子图越小,越容易与模板图中的纹理对齐,获得的检测效果也越好。经实验,在大多数应用场景中分辨率为 16×16 的子图在获得相对较好的检测效果的同时,也获得了相对较低的传输时延。

2.3.2 传输时延的优化

传输时延的优化主要通过时延折叠的方式来实现。该加速器的执行过程可抽象为 3 个步骤:数据输入、数据计算、结果输出。由于三者是对同一组片上缓存中的数据进行操作,因此在未经优化的情况下这 3 个步骤是一个串行的过程。常见的乒乓操作是例化两组片上缓存,在每组缓存上交替执行输入与输出的步骤,以实现输入与输出的时延折叠[14]。本文则进一步采用了三重缓冲的思想,例化了 3 组片上缓存,将输入、计算、输出三者的时延折叠,在传统乒乓操作的基础上进一步优化传输时延,其时序

图如图 4 所示。

图 4　采用三重缓冲后的时序图

Fig. 4　Time sequence with triple buffering

其中，同一种颜色代表同一组片上缓存，在某一时刻往第一组片上缓存中输入数据时，计算单元开始处理第二组片上缓存，同时将第三组片上缓存中的结果输出到内存中。3 组片上缓存交替执行各个步骤，构成一个三级流水线。

2.3.3　计算时延的优化

上文中提到完成一次模板匹配运算需进行 $m \times n \times p \times q$ 次减法运算，因此对于 16×16 的待检测子图及 8×8 的纹理周期，完成一次模板匹配运算需进行 16 384 次减法运算。如果加速器内只有一组运算单元，即使对整个运算过程进行了流水线优化，在运算单元的起始间隔为 1 个时钟周期的情况下，仍旧需要 16 384 个时钟周期才能完成一轮计算，完成一张1 024×1 024 图片的模板匹配运算，则至少需要 447 ms。以 16 的突发传输长度实现分辨率为 1 024×1 024 图片的传输仅需 31 ms，时延折叠方式，则会产生如图 5 所示的时序图，此时的运算过程将成为性能瓶颈，因此需要对计算时延进行优化。

图 5　计算时延优化前的时序

Fig. 5　Time sequence before computation delay optimization

虽然可以通过例化多组计算单元并行计算来达到优化的目的，但是模板图与待检测图存储在片上缓存即 BRAM 中，而 BRAM 本身的读写端口数是有限的，一般为两个，即一个时钟周期内只能读取两个数据。可将 BRAM 进行分块，以此来增加接口数量，使得在一个周期内读取更多的数据，再通过例化多个计算单元实现并行计算[15]。

如图 6 所示，将 24×24 的模板图沿着 x 方向划分成 24 组数据，并存入 24 块 BRAM 中，将 16×16 的待检测图同样沿着 x 方向划分成 16 组数据，并存入 16 块 BRAM 中。每块 BRAM 拥有两个端口，因此一个时钟周期内可读出 32 组数据，同时再例化 32 组计算单元实现并行计算，将原本16 384个时钟周期的运算时延缩减为 512 个时钟周期。实测该方案为分辨率为1 024×1 024的图片进行模板匹配运算耗时约 24 ms，此时的数据传输与计算过程的时延相差不大，时延得到充分折叠，时序如图 7 所示，且消耗的资源也不多。

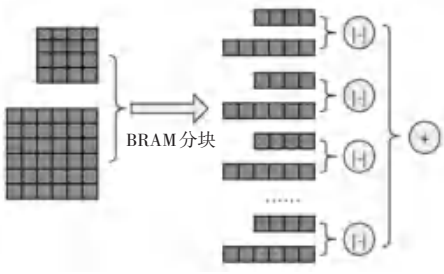

图 6　BRAM 分块过程

Fig. 6　BRAM partition process

图 7　计算时延优化后的时序

Fig. 7　Time sequence after computation delay optimization

如果将所有 BRAM 彻底分块，即把模板图与待检测图的每一个数据都存储在一个单独的寄存器中，例化 256 组计算单元，以实现在 64 个周期内完成一次模板匹配运算，该方案虽然可行但是访存时延就会成为瓶颈，同时这种方案会消耗大量不必要的逻辑资源。

3　实验评估

3.1　实验环境

使用搭载 Zynq-7020 芯片的 Pynq-Z2 开发板，Zynq-7020 异构平台由双核 ARM Cortex-A9 与 Artix-7 FPGA 组成。其中 FPGA 部分的资源包括可编程逻辑单元 85 K、片上缓存 BRAM 4.9 Mb、DSP 切片 220 个，双核 A9 的时钟频率为 667 MHz，板载 512 MB 内存。采用工厂中常见的的 4 K 工业线扫相机作为数据输入源，像素频率为 24 MHz，该类相机每分钟可采集宽度约 1 m、长度约 85 m 的织物图像。开发板通过转接板扩展出 Camera Link 接口与线扫相机连接。FPGA 部分除了包含模板匹配算法加速器，还集成了用于将 Camera Link 差分信号解析为 RGB 数据，并将 RGB 数据转为 AXI4-Stream 总线数据传入内存的一系列相关 IP 核。其中模板匹

配加速器的资源消耗见表 1。

表 1 模板匹配加速器资源耗费

Tab. 1 Template matching accelerator resource consumption

LUT	FF	*BRAMs*	DSP	*f*/ Mhz
6325(11.89%)	5588(5.25%)	1(0.7%)	6(2.73%)	150

其中,LUT(Look-Up-Table)为查找表;FF(Flip Flop)为触发器; *BRAMs* 为大小为 36 Kb 的片上缓存数量;DSP(Digital Signal Processing)为数字信号处理器,以上均为 FPGA 内部资源; *f* 代表该加速器的工作时钟频率。

与之对比的通用 PC 机,采用 CPU i7-8750H,6 核12 线程,默认主频 2.2 GHz,搭载 16 Gb 内存,通过 PCIE(peripheral component interconnect express)外接图像采集卡的方式来连接线扫相机。

在成本的方面,传统的 PC 级板卡式结构中仅一张专业的 Camera Link 图像采集卡价格就千元以上,一台高性能主机的价格也普遍在 5 000 元以上。相比之下,一块搭载 Zynq-7020 的开发板价格仅为 1 000 元左右,成本降低了 6 倍以上。

3.2 性能对比

本文设计的模板匹配加速器的处理效果在观感上与通用 CPU 的处理效果一致,无论是通用 CPU 还是本文设计的加速器,都很好地消除了织物的纹理背景,凸显了瑕疵区域。由于本文中的模板匹配算法是以子图为单位进行处理的,因此处理结果存在轻微网格效应,通过 OTSU 或 TRIANGLE 自适应二值化及一些基本形态学操作后可以轻松消除这一现象,处理效果如图 8 所示。此外,对织物原图进行诸如保边滤波、纹理提取等预处理步骤之后,再对纹理特征进行模板匹配,可以更好地消除噪声及光照不均所带来的干扰,获得更好的检测效果。

在检测速度上,通用 PC 与本文设计的加速器对比见表 2。处理一张分辨率为 1 024×1 024 的图片,使用同样的算法,基础频率为 2.2 GHz 的 i7-8750H 耗时 0.324 s,无法跟上线扫相机 24 MHz 的像素频率,时常出现漏帧现象。相比之下,本文设计的模板匹配加速器,在 PL 部分时钟频率为 150 MHz 的 Zynq-7020 平台上耗时仅为 0.031 s,速度是 CPU 的 10.5 倍,相当于 33 MHz 的像素处理频率,该速率大于工业线扫相机 24 MHz 的像素频率。工厂中普遍采用 4 个像素覆盖 1 mm 的织物长度,以此来估算,对于 4k 分辨率的线扫相机,本系统每分钟可处理的织物面积约为 120 m^2,大于每分钟 85 m^2 的速率要求,满足了实时性需求。

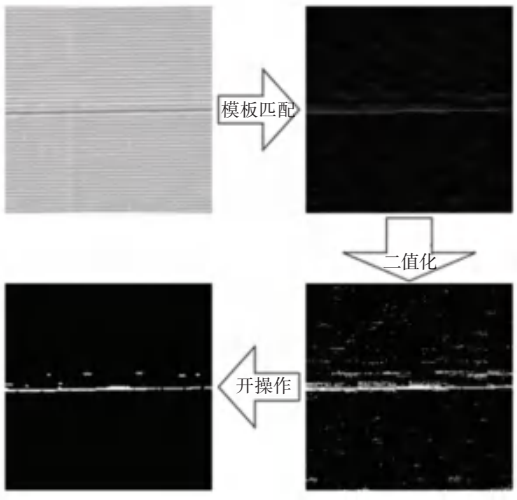

图 8 加速器处理结果

Fig. 8 Accelerator processing result

表 2 本文加速器与通用 PC 对比

Tab. 2 The accelerator compared with general PC

参数	PC(x86) i7-8750H	ARM+FPGA Zynq-7020
时钟频率/MHz	2 200	650+150
内存大小/Mb	16 384	512
内存频率/MHz	2 666	1 066
峰值功率/W	56	2.101(1.256+0.845)
处理时延/ms	324	31

4 结束语

为了将织物瑕疵检测算法更有效地部署到实际生产环境中,本文对瑕疵检测领域的模板匹配算法进行了改进,为该算法设计了一种基于 FPGA 的算法加速器,并对该加速器的各项处理时延进行了优化。最终该加速器在 Zynq-7000 系列异构平台上获得了 33 MHz 的像素处理频率,相当于每分钟可处理面积约 120 m^2 的织物图片,满足了工业领域的实时性需求。设计过程中采用 SoC 技术取代了传统的 PC 级板卡式结构,使织物瑕疵检测系统的成本降低了 6 倍以上。

参考文献

[1] 王钢,周建,汪军,卜佳仙,等. 采用奇异值分解的机织物瑕疵检测算法[J]. 纺织学报, 2014, 35(7): 61-66.

[2] KUMAR A. Computer-Vision-Based Fabric Defect Detection: A Survey [J], IEEE Transactions on Industrial Electronics, 2008, 55(1): 348-363.

[3] LI F, LI F. Bag of tricks for fabric defect detection based on Cascade R-CNN [J]. Textile Research Journal. September, 2021, 91(5-6):599-612.

(下转第 14 页)