

DOI:10.19651/j.cnki.emt.2518368

基于 FPGA 的轻量化霍夫变换加速器设计与实现^{*}

蒋晨鑫 肖昊 徐瀚 朱骄阳

(合肥工业大学微电子学院 合肥 230601)

摘要: 霍夫变换是一种检测直线的常用方法,有很好的抗干扰能力及鲁棒性。但由于霍夫变换检测直线计算复杂度高,存储需求大,在硬件上部署困难。本研究提出了一种基于分级霍夫变换思想的改进霍夫变换算法,该算法将一次霍夫变换分解成两次变换运算,第一次运算图像经过下采样处理降低了第一级投票器的存储需求,第二次运算的投票器存储范围受第一次运算出来的参数限制,很好地解决了霍夫变换在硬件部署上存储需求大的问题,并且利用三角函数变换公式改进霍夫变换算法使得每次变换可以采用并行流水线设计,提高了计算效率。实验基于 FPGA 实现了分级流水线霍夫变换硬件架构,结果表明,本研究提出的架构在片上 RAM 资源占用上较经典霍夫变换硬件架构降低了 89.8%,检测精度综合提升了 39.94%,在 100 MHz 时钟频率下检测一张 1024×1024 的图片中的直线花费的时间为 13.11 ms,相比于软件执行霍夫变换直线检测速度上有很大的提升。

关键词: FPGA;霍夫变换;硬件加速器;直线检测;图像处理

中图分类号: TN79 文献标识码: A 国家标准学科分类代码: 510.4050

Design and implementation of a lightweight Hough Transform accelerator based on FPGA

Jiang Chenxin Xiao Hao Xu Han Zhu Jiaoyang

(School of Microelectronics, Hefei University of Technology, Hefei 230601, China)

Abstract: The Hough Transform is a commonly used method for detecting lines and has excellent interference resistance and robustness. However, due to its high computational complexity and large storage requirements, deploying the Hough Transform on hardware is challenging. This study proposes an improved Hough Transform algorithm based on the concept of hierarchical Hough Transform. The algorithm decomposes a single Hough Transform into two transformation operations. The first operation involves downsampling the image, which reduces the storage demand of the first-level voting unit. The storage range of the second operation's voting unit is limited by the parameters obtained from the first operation, effectively addressing the issue of high storage requirements for hardware deployment. Moreover, by improving the Hough Transform algorithm using trigonometric transformation formulas, each transformation can be designed with parallel pipelining, enhancing computational efficiency. A hierarchical pipelined Hough Transform hardware architecture based on FPGA has been implemented. Experimental results show that the proposed architecture reduces on-chip RAM resource usage by 89.8% compared to the classic Hough Transform hardware architecture, and the detection accuracy is improved by 39.94%. At a clock frequency of 100 MHz, it takes 13.11 ms to detect lines in a 1024×1024 image, which is a significant improvement over the speed of software-based Hough Transform line detection.

Keywords: FPGA; Hough Transform; hardware accelerator; line detection; image processing

0 引言

霍夫变换(Hough transform, HT)是一种常用于形状识别的图像处理技术,主要用于检测图中直线、圆等简单的

几何形状^[1]。核心思想是将图像像素点坐标由笛卡尔坐标系转换到参数空间坐标系进行投票操作得到图像中满足条件的形状参数,这使得即使在图像中有噪声或者线条断裂的情况下依然能很好检测出形状。随着自动驾驶技术的快

收稿日期:2025-03-17

*基金项目:国家自然科学基金(62274052)项目资助

速发展以及屏幕缺陷检测需求的提升,霍夫变换因其很好的鲁棒性^[2]常用来检测车道和提取屏幕有效区域,并且随着深度学习技术的快速发展,将深度学习和霍夫变换相结合,通过在深度特征上进行霍夫变换,兼顾了卷积神经网络的特征学习能力和霍夫变换的高效性,能够实现很高精度的形状检测^[3]。但由于霍夫变换需要对图像中满足条件的每个像素点都进行一次坐标参数变换并在投票器中进行投票,这使得算法计算速度慢,基于中央处理器(central processing unit, CPU)的平台因其低并行性而导致很高的延迟,很难满足实时处理的需求。

现场可编程门阵列(field programmable gate array, FPGA)是一种可配置的逻辑器件^[4],目前已广泛应用于车道检测,图像边缘检测等图像处理场景^[5],基于FPGA的霍夫变换算法被大量研究,由于深度学习的高计算资源需求,很难部署到资源有限的嵌入式系统,所以更多的基于FPGA的霍夫变换研究是在传统霍夫变换上进行改进。其中研究方向主要有两个:一是对算法的FPGA实现过程进行优化,二是对霍夫变换算法本身进行优化。对算法的FPGA实现过程进行优化主要是对部署到硬件上的参数进行部分限制以及根据应用场景进行硬件的专项优化。文献[6]中优化了霍夫变换算法在FPGA中的实现,对霍夫参数空间进行限制,能够在大量噪声像素点的实验条件下准确快速的检测直线,但存在FPGA片上存储资源占用过多的问题。文献[7]对特定车道检测场景中的霍夫变换算法进行优化并用FPGA实现,减少FPGA片上资源消耗,但该改进算法未对整张图片进行霍夫变换,而是将感兴趣的区域进行霍夫变换,同时转化到的参数空间并不是经典霍夫变换直线检测中的 (ρ, θ) 而是 (b, θ) 其中 b 为直线截距,应用场景过于局限很难推广。文献[8]针对扫描图像为条形码的应用场景进行算法优化,只对分离出来的中心 40×40 的区域进行霍夫变换,当检测出来的角度接近 45° 时,会进行二次判断,避免因条码过粗导致的误判,但由于里面许多参数设置需要特定情况下的经验值,所以同样存在应用范围受限且参数依赖性较强的问题。对霍夫变换算法进行优化主要是对算法的投票环节以及霍夫变换公式进行优化,文献[9]提出一种改进的霍夫变换算法,该算法使用Sobel边缘检测算法计算出来的梯度角来限制霍夫变换投票器中的参数范围,通过缩小参数范围来降低内存需求,但该算法的检测精度容易受到边缘检测效果的影响。文献[10]对霍夫变换算法进行改进,该算法利用该论文中实验的特定坐标系统即 $X:Y:Z$ 坐标系和物理条件如均匀磁场,将传统的霍夫变换公式简化并优化,以适应高能物理实验中的粒子轨迹跟踪需求,简化后虽然提高了计数速度但也一定程度上降低了精度。随着图像采集系统的发展,图像分辨率越来越大,要求基于FPGA的霍夫变换检测系统满足低延时的同时,尽可能的降低片上资源的占用,方便将其应用于自动驾驶以及屏幕缺陷检测系统,同时随着应用场景越来越复

杂,要求在不同的情况下都能有很好的检测效果。

针对以上问题,本研究提出了一种基于分级霍夫变换思想的改进霍夫变换算法,第一级霍夫变换实现直线的大致定位,将得到的参数传入第二级霍夫变换,第二级霍夫变换实现直线的精确定位,通过对两级霍夫变换参数范围的限制,使得整个算法过程中占用的存储空间减少。同时基于FPGA特性设计了一个并行的流水线霍夫变换结构,提高计算效率并且片上随机存储器(random access memory, RAM)资源占用降低。

1 霍夫变换直线检测算法

1.1 经典霍夫变换算法

霍夫变换最初使用斜截式作为参数空间来检测直线,在量化的参数空间进行投票检测直线,但因为无法检测斜率不存在的直线,难以在实际应用中推广。后续对算法的改进中用 (ρ, θ) 参数空间代替了原来的斜截式参数空间,其中 ρ 为坐标原点到直线上最近点的距离, θ 表示直线的法线与 x 轴正方向之间的夹角,因此可以检测平面内的任意直线,这种修改被认为是经典的HT实现,在实践中被广泛的应用。

经典霍夫变换的计算公式如下:

$$\rho = x \cos \theta + y \sin \theta \quad (1)$$

其中, x 和 y 分别是像素点在笛卡尔坐标系下的坐标,对于直线上的任意一点,其 (ρ, θ) 值保持不变,并且在笛卡尔坐标系上的每个点在经典霍夫变换参数空间中对应一个正弦波。

经典霍夫变换是通过将霍夫空间量化为有限区间来实现的,每个区间称为累加单元,当进行霍夫变换时,每个像素点的坐标被转换成一个离散的 (ρ, θ) 曲线,并且沿着这条曲线的累加单元进行递增。当所有像素点完成参数空间转换后,在累加器空间中寻找局部最大值,提取出最可能的直线参数值。

1.2 改进的分级角度平行霍夫变换算法

针对经典霍夫变换计算公式,如果角度的最小步长为 1° ,角度的范围为 $-90^\circ \sim 90^\circ$,则仅一个像素点就需要进行360次乘法运算和180次加法运算,计算量非常大。同时对于大部分部署在FPGA平台上的HT,为了提高处理频率和减少计算资源的占用,会将 $\sin \theta$ 和 $\cos \theta$ 的计算值直接存放在只读存储器(read only memory, ROM)中,这会占用一部分片上存储资源,为了提高计算速率和减少存储资源消耗,对经典霍夫变换进行改进。

基于FPGA可以并行处理数据的特性,将 ρ 的计算由 θ 和 $\theta + \pi/2$ 两部分同时开始计算,如式(2)所示, ρ_1 和 ρ_2 分别表示当 θ 取 $[0, \pi/2]$ 和 $[-\pi/2, 0]$ 时的计算值。

$$\begin{cases} \rho_1 = x \cos \theta + y \sin \theta, & 0 \leq \theta \leq \frac{\pi}{2} \\ \rho_2 = x \cos \theta + y \sin \theta, & -\frac{\pi}{2} \leq \theta \leq 0 \end{cases} \quad (2)$$

定义 $\theta = \theta' - \pi/2$, 则 θ 取 $[-\pi/2, 0]$ 时, ρ_2 的计算公式可以表示为:

$$\begin{aligned} \rho_2 &= x \cos(\theta' - \frac{\pi}{2}) + y \sin(\theta' - \frac{\pi}{2}) = \\ x \sin\theta' - y \cos\theta', \quad 0 \leqslant \theta' &\leqslant \frac{\pi}{2} \end{aligned} \quad (3)$$

自此, 式(2)推导为如下:

$$\begin{cases} \rho_1 = x \cos\theta + y \sin\theta, \quad 0 \leqslant \theta \leqslant \frac{\pi}{2} \\ \rho_2 = x \sin\theta' - y \cos\theta', \quad 0 \leqslant \theta' \leqslant \frac{\pi}{2} \end{cases} \quad (4)$$

ρ_1 和 ρ_2 的计算可以在 FPGA 中并行开始, 且 θ 的范围大小降为原来的一半, 计算速率提高了一倍。当定义 θ 的步长为 $\Delta\theta$ 时, 按照式(4)中进行计算, θ 的每次取值为 0, $\Delta\theta, 2\Delta\theta, \dots, \pi/2$ 。

当 $\theta = 0$ 时, 代入式(4)可得:

$$\begin{cases} \rho_{11} = x \\ \rho_{21} = -y \end{cases} \quad (5)$$

当 $\theta = \Delta\theta$ 时, 代入式(4)可得:

$$\begin{cases} \rho_{12} = x \cos\Delta\theta + y \sin\Delta\theta = \cos\Delta\theta(\rho_{11} - \rho_{21}\tan\Delta\theta) \\ \rho_{22} = x \sin\Delta\theta - y \cos\Delta\theta = \cos\Delta\theta(\rho_{11}\tan\Delta\theta + \rho_{21}) \end{cases} \quad (6)$$

当 $\theta = (i+1)\Delta\theta$, 代入式(4)可得:

$$\begin{cases} \rho_{1(i+1)} = \cos\Delta\theta(\rho_{1i} - \rho_{2i}\tan\Delta\theta) \\ \rho_{2(i+1)} = \cos\Delta\theta(\rho_{1i}\tan\Delta\theta + \rho_{2i}) \end{cases} \quad (7)$$

由式(7)可以得知 ρ 的计算不再需要 $\sin\theta$ 和 $\cos\theta$ 的值, 只需要设置好 $\Delta\theta$ 值, 通过 ρ_i 可以计算得到 ρ_{i+1} 的值, 节省了用于存放 θ 的正余弦值的 ROM 资源。

根据分级霍夫变换的思想, 将霍夫变换分为两级, 第一级实现形状的粗略定位, 第二级实现直线的精确检测, 以 256×256 分辨率图像为例, 第一级霍夫变换中的 $\Delta\theta_1$ 值取得相对较大, 式(7)中的 $\Delta\theta_1$ 取 1.792° , 则 $\tan\Delta\theta_1 = 1/(2^5)$, $\cos\Delta\theta_1 \approx 1$, 可得:

$$\begin{cases} \rho_{1(i+1)} = \rho_{1i} - \rho_{2i} \cdot \frac{1}{2^5} \\ \rho_{2(i+1)} = \rho_{1i} \cdot \frac{1}{2^5} + \rho_{2i} \end{cases} \quad (8)$$

第二级霍夫变换将第一级霍夫变换得到的粗略直线参数用来限定参数范围, 假设第一级得到的直线参数为 (ρ_a, θ_a) , 则第二级霍夫变换的 ρ 和 θ 的取值范围分别为 $(\rho_a - 20, \rho_a + 20)$ 和 $(\theta_a - 20^\circ, \theta_a + 20^\circ)$, 第二级需要对直线精确定位, 所以 $\Delta\theta_2$ 取 0.4472° , 则 $\tan\Delta\theta_2 = 1/(2^7)$, $\cos\Delta\theta_2 \approx 1$, 代入式(7)可得:

$$\begin{cases} \rho_{1(i+1)} = \rho_{1i} - \rho_{2i} \cdot \frac{1}{2^7} \\ \rho_{2(i+1)} = \rho_{1i} \cdot \frac{1}{2^7} + \rho_{2i} \end{cases} \quad (9)$$

式(8)和(9)将原本复杂的正余弦乘法简化成适合硬件

实现的移位和加法, 节省了计算资源, 同时将图像的坐标原点定位由图像的左上角移至图像的正中心, ρ 的取值范围可以降低到原来的一半, 节省投票器的大小^[11]。由于第一级霍夫变换只是粗略的检测直线的位置, 所以先对原图进行下采样降分辨率得到 128×128 分辨率的处理后图片, 再进行第一级霍夫变换, 则第一级 ρ 和 θ 的取值范围分别为 $(-91, 91)$ 和 $(0^\circ, 180^\circ)$, 而第二级霍夫变换要精确定位直线, 所以第二级霍夫变换输入为未经过下采样的原图, 但由于第一级得到的粗略的直线参数范围限制, 使得第二级霍夫变换投票器占用的内存大小大幅度减小。

2 硬件架构

2.1 分级流水线 HT 硬件架构

图 1 给出的是经典霍夫变换的硬件架构^[7], 由 3 个部分组成, 分别是输入像素坐标缓存单元, HT 计算单元, 投票器。其中输入为经过二值化和边缘检测处理的像素点值, 输出为满足阈值条件的霍夫空间直线参数 (ρ, θ) , 由于每个满足条件的像素点坐标输入后要保持 180 个时钟周期用于计算不同 θ 值对应的 ρ , 所以经典霍夫变换硬件架构没办法实现全流水结构, 必须存在一个输入像素点坐标缓存单元, 且这个缓存单元的大小与图片的尺寸和图片中表示直线的像素点个数相关, 以 256×256 分辨率的图像为例, 假设图片中有一半的像素满足大于像素阈值的条件, 则 buffer 的深度设置成 32 768, 由于图像横纵坐标的取值范围为 $0 \sim 256$, 所以输入数据的位宽为 16 bit, 其中高 8 位表示横坐标, 低 8 位表示纵坐标, 此时 buffer 占用的 RAM 资源为 512 Kbit, 占用了较多的 RAM 资源。

本文依据改进的分级角度平行霍夫变换算法提出的分级 HT 硬件顶层架构如图 2 所示, 主要包括图像数据的前期预处理, HT 计算单元以及外部存储器和数据输入输出控制处理器。其中原图输入后经过二值化和 Canny 边缘检测变成只含有屏幕边缘信息的二值图像来用于第一级霍夫变换。

第一级霍夫变换时, 对二值化处理后的图像进行下采样, 随后选出像素值等于 255 的为参与投票的像素点, 将其坐标信息输入 HT 计算流水线处理单元 (processing element, PE) 中, 每个 PE 负责通过像素点坐标计算出特定角度下的 ρ 存入投票器进行投票, 当投票器中特定位置投票值大于投票阈值的时候, 该位置参数就会被选出进行缓存用于计算第二级霍夫变换。

第二级霍夫变换时, 因为要对图像中的直线进行精确检测, 输入的图像为未经下采样的二值化图片, 同时输入还有第一级霍夫变换得到的直线粗略位置的参数, 在 HT 计算流水线中, PE 计算值只有满足在第一级霍夫变换得到的参数的一定区间范围内这个条件才会放入投票器进行投票, 同样当投票器中特定位置投票值大于投票阈值的时候, 这个位置参数就会被选出作为最终的直线拟合结果输出。

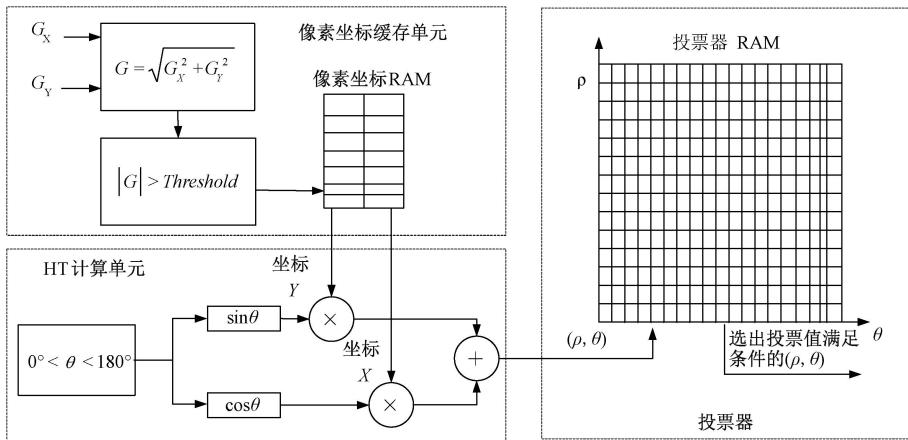


图 1 传统霍夫变换硬件架构

Fig. 1 Conventional Hough transform hardware architecture

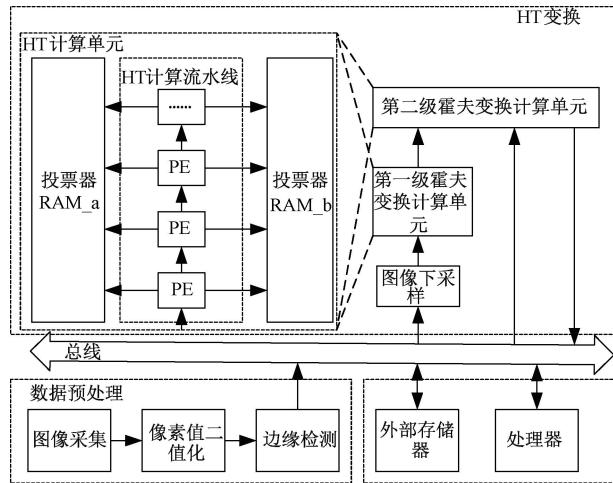


图 2 分级霍夫变换硬件架构图

Fig. 2 Hierarchical Hough transform hardware architecture diagram

其中每一级霍夫变换中 HT 计算流水线中 PE 的数量取决于 $\Delta\theta$ 值的选取,以 256×256 分辨率的二值图像为例,经过下采样后图片分辨率降为 128×128 ,此时第一级霍夫变换 $\Delta\theta_1 = 1.792^\circ$,由于在分级角度平行霍夫变换算法中 θ 的范围为 $0^\circ \sim 90^\circ$,所以此时 PE 的个数为 51 个,在第二级霍夫变换中,假设第一级霍夫变换得到的参数为 (ρ_1, θ_1) , θ 的取值范围为 $(\theta_1 - 20^\circ, \theta_1 + 20^\circ)$,大小为 40° ,由于 $\Delta\theta_2 = 0.4472^\circ$,所以第二级霍夫变换中 PE 个数为 90 个。

由于对第一级霍夫变换输入图像进行下采样处理,对第二级霍夫变换投票器输入 (ρ, θ) 参数进行限制,使得整个霍夫变换架构需要的 RAM 资源大幅度减少,同时每级中 HT 计算采用流水线设计,不需要额外的 RAM 对满足条件的像素点坐标进行缓存,进一步减少了片上存储资源的消耗。

2.2 图像下采样

将输入进来的像素点数据按图 3 所示方法进行下采

样,可以将图片分辨率降低到原来的 $1/4$,下采样的过程是一个信息损失的过程,所以只在第一级霍夫变换时对图像数据进行下采样操作,完成一个直线粗略定位的功能。

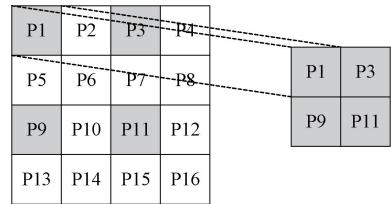


图 3 图像下采样示意图

Fig. 3 Schematic diagram of image downsampling

图像下采样的幅度可以根据图像分辨率的大小作调整,下采样的幅度过大会导致图像信息损失过多,直线粗略定位误差太大,幅度太小会导致后续的计算处理的数据量太大。

2.3 处理单元(PE)

PE 的结构如图 4 所示,对于每 1 个 PE,需要 2 个寄存器,2 个 N 位的移位器,1 个加法器,1 个减法器和 2 个累加器 RAM_a(i) 和 RAM_b(i) 用于参数累加。由于分级角度平行霍夫变换算法将乘法和三角函数运算转换成了移位和加法,可以很容易设计成适应 FPGA 电路的并行结构,其中输入为前 1 个 PE 单元的输出 ρ_1 和 ρ_2 ,将其作为 RAM_a(i) 中对应的内存地址,该地址中的值加 1,对于 RAM_b(i) 也采取同样的操作。在 2 个累加器中,累加地址只需要表示 ρ 的值而不需要 θ 的值,因为 θ 的值可以通过 $\Delta\theta$ 和 PE 单元的序号求得。同时可以通过移位和加法减法操作得到下一个 PE 的输入 $\rho_{1(i+1)}$ 和 $\rho_{2(i+1)}$ 。

如图 5 所示,每个 PE 单元单独完成 1 次运算需要 2 个时钟周期。应用于第一级和第二级霍夫变换的式(8)和(9)的 PE 结构都是一样的,只要将移位操作中的 N 按要求进行设置。以 256×256 分辨率的图像为例,在第一级霍夫变换中先进行下采样降成 128×128 分辨率的图像,里面

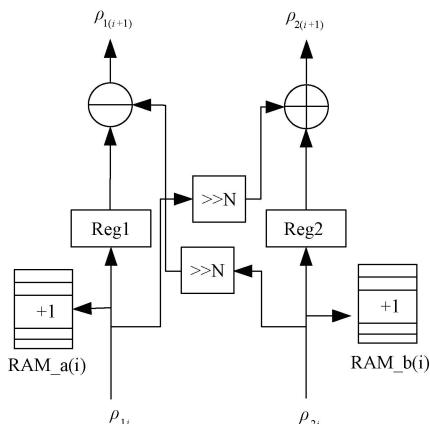


图 4 PE 结构图

Fig. 4 PE structure diagram

的 PE 输入的 ρ 的位宽为 16 bit, 其中 8 bit 表示小数, 投票值地址位宽为 8 bit, 移位寄存器中的 N 的值为 5, 投票器数据位宽为 8 bit。在第二级霍夫变换中处理的数据是未经过下采样的图片, 所以里面的 PE 输入的 ρ 的位宽为 17 bit, 其中 8 bit 表示小数, 投票器地址位宽为 6 bit, 移位寄存器中的 N 的值取 7, 投票值的数据位宽为 8 bit。对于不同角度 PE 中的两个 RAM 大小还可以根据 ρ 的定义优化, 不同角度的 ρ 的取值范围不一样, 可以进一步优化减少片上存储资源占用。

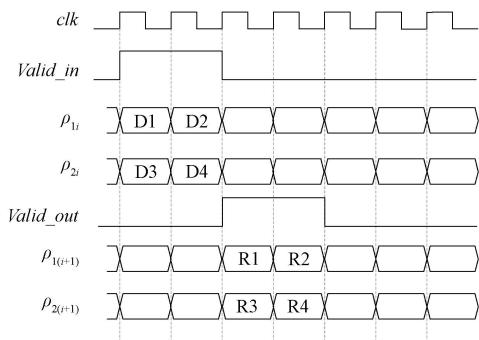


图 5 PE 时序图

Fig. 5 PE timing diagram

每个 PE 单元表示一个角度步进计算, 任一 PE 的输出作为下一个流水线单元的输入参数, 这样的结构优点是在填充完流水线后, 每个边缘像素的投票值计算只需要一个时钟周期就可以完成, 提高了计算速率。

3 实验与评估

3.1 软件仿真测试

实验的目的是对手机屏幕缺陷区域位置进行定位, 先通过工业相机采集包含手机屏幕的原始图片, 再通过霍夫变换直线拟合提取出有效屏幕区域, 最后进行屏幕的缺陷检测, 其中如何提取有效屏幕区域是本次实验的重点。

实验比较了文献[7]中提到的经典霍夫变换算法和分

级角度平行霍夫变换算法来检测直线, 调用开源计算机视觉库(open source computer vision library, OpenCV)实现经典霍夫变换算法和用 C 语言实现分级角度平行霍夫变换算法, 通过比较各自拟合直线框定的区域经过透射变换提取出来的手机屏幕效果来定性判断直线拟合质量。其中竖直方向直线霍夫变换的阈值设置成 156, 水平方向霍夫变换阈值设置成 100, 采集的图像分辨率为 9344×7000 的高分辨率图像。

使用相机采集多张包含手机屏幕各种颜色的图像得到图 6(a), 目标是拟合出手机屏幕的四条直线边并将屏幕通过透射变换提取出来。图 6(b)采用经典霍夫变换拟合屏幕边缘直线, 可以看到四条直线拟合出来有较大的误差, 屏幕的区域无法完全提取出来, 一方面是因为图片的尺寸比较大, 干扰的像素点和噪声较多, 另一方面是因为参数步长受限, ρ 和 θ 无法取比较小的步进, 否则会导致投票器过大, 占用的内存资源过多, 而图 6(c)采用的分级角度平行霍夫变换算法, 由于第一级霍夫变换已经对直线进行了粗略定位, 所以第二级霍夫变换只需要对小范围的区域进行变换运算, 所以第二次的参数步进可以取的很小, 实现很高的检测精度, 同时投票器占用的内存也会远远小于经典霍夫变换。

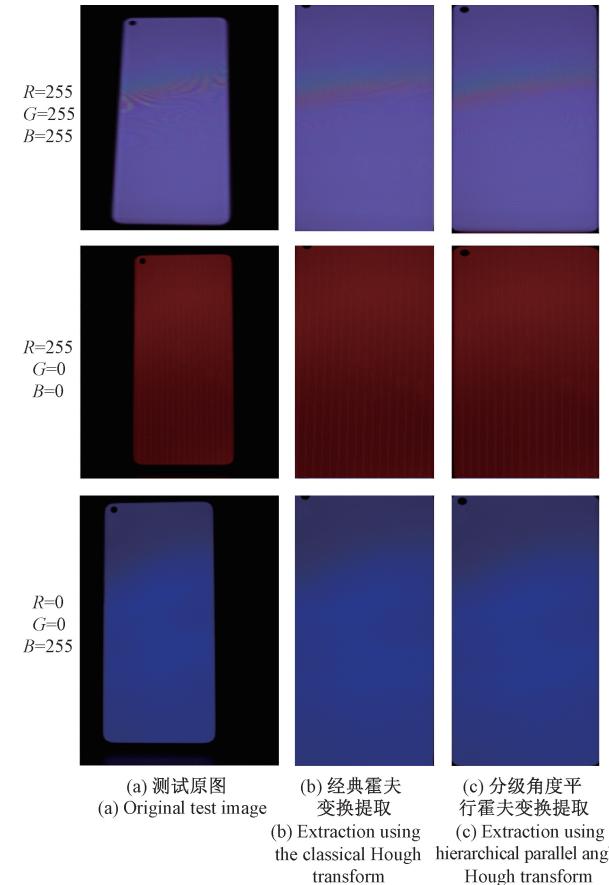


图 6 不同算法提取效果

Fig. 6 The extraction effects of different algorithms

通过平均误差法^[12]可以定量分析直线拟合的效果,通过在样本图像中手动标记来获得需要拟合直线的真值参数(ρ, θ),将算法拟合出来的各条直线的参数与其真值参数进行相减然后求平均值,得到平均误差值。对多张图片进行实验得到两种算法的量化分析如表 1 所示,分析得出分级角度平行霍夫变换算法相比经典霍夫变换算法其拟合出来的直线误差更小。对多张图片进行测试,得到每张图片在两种方法下屏幕各个边缘的拟合误差,计算得到误差的方差如图 7(a)和(b)所示,由于拍摄时将手机竖直拍摄,所以屏幕的左边缘和右边缘分别是手机的长边框,上边缘和下边缘是短边框,根据霍夫变换的投票机制并结合图中信息可以分析得出,由于长边框的有效投票点更多,投票时受到噪点的影响更小,所以拟合的质量更稳定,方差更小,且可以看到本研究提出的分级角度平行霍夫变换的拟合稳定性要稍微优于经典霍夫变换。

表 1 两种算法误差分析

Table 1 Error analysis of two algorithms

霍夫变换算法	ρ /像素	θ (°)
经典霍夫变换算法	24.957	0.329
分级角度平行霍夫变换算法	14.725	0.201

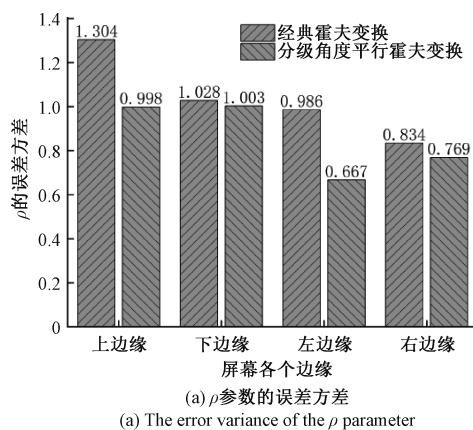
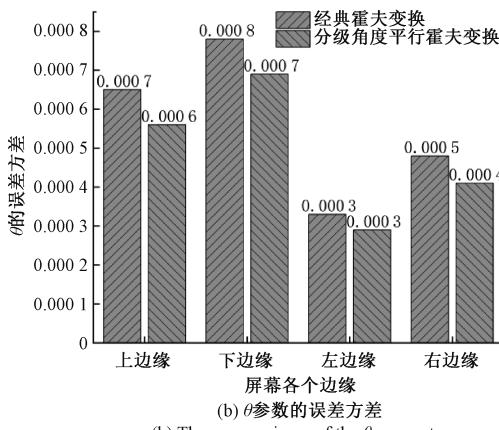
(a) ρ 参数的误差方差(a) The error variance of the ρ parameter(b) θ 参数的误差方差(b) The error variance of the θ parameter

图 7 霍夫参数的误差方差分析

Fig. 7 Variance analysis of Hough parameters

3.2 硬件性能与加速效果分析

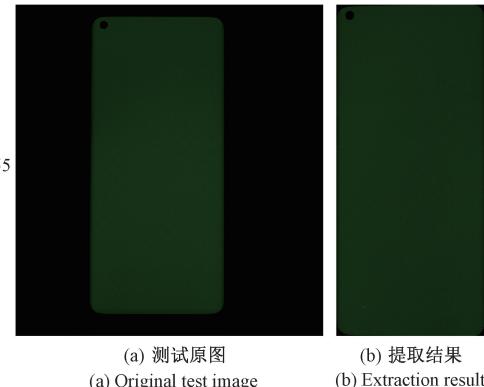
实验选择的硬件开发平台如图 8 所示,选用的是 Xilinx 公司 Genesys II 开发板,芯片的型号为 xc7k325tffg900-2。实验采用的是 100 MHz 的时钟频率。相机选用的是海康威视的工业相机 MV-CH250-90TM/C/N,采用 Gpixel 的 GMAX0505 型号的 CMOS 芯片,像元尺寸小,分辨率高,图像优异,采用万兆以太网接口,快速实时传输非压缩数据,最高帧率可达 41.5 fps。



图 8 硬件开发平台

Fig. 8 Hardware development platform

实验采用 Verilog 硬件描述语言完成硬件编程,在 Xilinx Vivado 2019.1 开发套件上完成了功能仿真和时序仿真,最终部署在 Genesys II 开发版上,将最终检测的结果输出到主机上进行数据分析,结果如图 9 所示,可以看到分级角度平行霍夫变换硬件实现有较好的拟合效果,可以完整的提取出屏幕像素,多张图片拟合出的直线的平均误差 ρ 可以控制在 20 像素以下, θ 误差可以控制在 0.5° 以下。表 2 所示为霍夫变换实现目标区域提取在软件和硬件上实现所花费的时间,花费的时间与输入图片尺寸相关,但相比于软件实现直线拟合目标区域提取,使用硬件实现分级角度平行霍夫变换算法来完成目标区域提取所花费的时间会大幅度降低。



(a) 测试原图

(a) Original test image

(b) 提取结果

(b) Extraction results

图 9 硬件测试结果

Fig. 9 Hardware test results

本次实验采用的硬件平台是 Genesys II 开发板,测试图片为 1 024×1 024 分辨率图片,各资源占用如表 3 所示,该硬件平台资源足以满足系统设计需求。

将多张手机屏幕图像硬件拟合数据导出,分析分级角

表 2 算法软件实现和硬件实现速度比较

Table 2 Comparison of algorithm implementation speed between software and hardware

实现方式	文献	耗时/ms	输入图片尺寸
软件实现	文献[13]	120	512×512
	文献[14]	1 358	3 000×1 600
	本文算法	5 398	9 344×7 000
硬件实现	本文算法	13.11	1 024×1 024

表 3 FPGA 硬件综合结果

Table 3 FPGA hardware synthesis results

资源类型	资源总量/个	资源消耗/个	资源利用率/%
LUT	203 800	18 633	9.14
FF	407 600	25 557	6.27
BRAM	445	64	14.38
IO	500	103	20.6

度平行霍夫变换的 ρ 和 θ 准确率如表 4 所示, 其中规定 ρ 误差在 20 像素以内判定为拟合准确, θ 误差在 0.5° 以内判定为拟合准确。

表 4 测试结果准确率

Table 4 Test result accuracy

霍夫参数	准确率/%
ρ	97.5
θ	95.0

为了验证本研究提出的方法资源占用的优势, 比较了其他文献以及经典霍夫变换与本研究的片上资源消耗情况如表 5。文献[7]中提到的霍夫变换硬件架构各部分的片上 RAM 占比情况如图 10 所示, 当使用 256×256 的测试图片时, 总的块随机存储器资源占用为 2.25 Mbit, 3 个部分资源占比如图 3 所示, 其中主要片上 RAM 占用的是投票器, 占用了总共 80%, 其次是像素缓存单元, 占用了总共的 19%, 还有一小部分是 HT 计算单元, 由此可见主要是投票器占用了大量的 RAM 资源, HT 计算单元中主要是占用了查找表(look up table, LUT)资源和触发器(flip-flop, FF)资源来完成计算, 其中投票器硬件结构如图 11 所示, ρ 的取值范围为 724, 即两倍图像对角线长度, 因此 ρ 的位宽为 10 bit, θ 的取值范围为 $0^\circ \sim 180^\circ$, 所以位宽为 8 bit, 由于 256×256 图像中霍夫空间投票值最大为 363, 所以投票器的数据位宽为 9 bit, 随着输入图片分辨率的提升, ρ 的位宽也会随之增大, 需要的 RAM 地址位宽也会增加, RAM 资源占用会大幅度增加, 由于 HT 计算单元结构并不会随着分辨率的增大而改变, 所以 LUT 和 FF 的资源占用基本不变, 只是计算花费的时间更长。

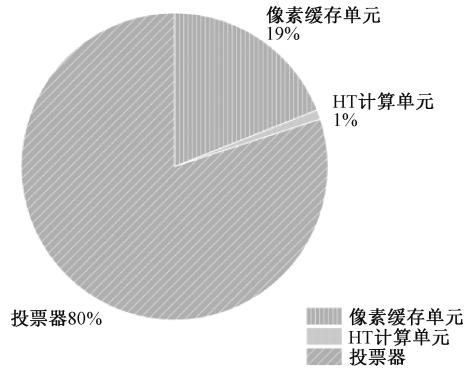


图 10 片上 RAM 占比情况

Fig. 10 The proportion of on-chip RAM usage

表 5 不同架构资源需求比较

Table 5 Comparison of resource requirements for different architectures

资源占用	图像分辨率	片上 RAM 资源/Kbit	LUT	FF
文献[7]	$1 024 \times 1 024$	11 264	23 677	34 789
文献[15]	$1 024 \times 768$	1 566	15 704	23 727
文献[16]	$1 000 \times 1 000$	6 624	12 492	32 673
本文架构	$1 024 \times 1 024$	1 152	18 633	25 557

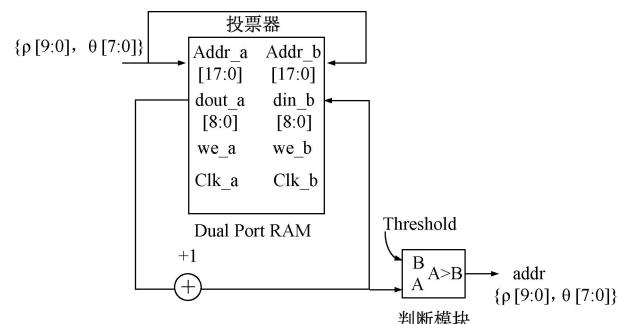


图 11 投票器硬件架构

Fig. 11 Voting device hardware architecture

文献[15]提出了 1 种并行化投票程序来提高霍夫变换计算速度和储存效率, 在一定程度上降低了片上存储资源压力, 但因为只进行一次霍夫变换, 使得占用的片上资源很大程度取决于参数步进的设置, 片上 RAM 资源降低的代价是降低霍夫变换直线检测精度。文献[16]使用 DSP 切片和 RAM 块来实现基于梯度的 HT, 此外, 三角函数值存储在 LUT 中, 同样是通过单次的霍夫变换来实现, 没有对投票器的参数范围进行约束, 所以该论文提出的方法节省的 RAM 资源非常有限, 有很大的优化空间。本研究提出的架构通过两次的霍夫变换, 第一次霍夫变换计算出来的参数用来缩减霍夫空间范围, 第二次霍夫实现直线的拟合, 两次霍夫变换的资源进行复用, 在不大量增加其他资源占用的情况下可以有效的降低片上 RAM 资源占

用,较经典霍夫变换硬件结构片上 RAM 资源占用降低了 89.8%,随着图片分辨率的提高,该架构的 RAM 资源占用并不会大幅度增加,因为在第一次霍夫变换的时候可以通过下采样降低待处理图像的分辨率,第二次霍夫变换通过参数限制处理图像的分辨率。由于没有过多需要人为确定的参数限制,所以可以推广到更多的应用场景,方便部署的同时还可以实现较好直线检测精度。

4 结 论

本研究针对经典霍夫变换算法计算量大及存储需求大的问题,首先提出了一种分级角度平行霍夫变换算法,每一级霍夫变换中的参数步进可以设置成需要的不同值,减少了计算量,且计算由三角函数计算和乘法运算简化成移位运算和加减法运算,降低了计算复杂度,便于硬件实现。其次,设计了 1 种高效的分级流水线 HT 硬件架构,通过并行流水线设计提高了计算速率,同时省去了存储三角函数值的 RAM 资源,通过两级霍夫变换的参数范围限制很好的解决了片上资源占用过多的问题。实验结果表明,本研究提出的霍夫变换硬件架构与其他文献中提到的架构相比可以进一步降低片上 RAM 资源消耗,且拥有较好直线的检测精度,检测速度较软件算法实现霍夫变换直线检测有很大的提高。在后续的研究中,可以针对不同的应用场景进行更进一步的硬件优化,如一些应用场景对检测精度要求不高,可以提高投票器的投票步进,来减少 RAM 资源的占用,通过将 HT 计算中的正余弦参数取近似可以减少计算资源的占用,或者某些场景对速度要求不高,可以不采用流水线设计,插入缓存,将更多的中间资源进行复用的方法来减少计算资源,方便嵌入其他的硬件系统完成图像处理功能。

参考文献

- [1] 沈新平,彭刚,袁志强. 基于霍夫变换和 RANSAC 算法的绝缘子定位方法[J]. 电子测量技术,2017,40(6): 132-137.
SHEN X P, PENG G, YUAN ZH Q. Insulator location method based on Hough transformation and RANSAC algorithm [J]. Electronic Measurement Technology, 2017, 40(6): 132-137.
- [2] 沈烨,王泉,吴昊. FSSD-DETR 自动驾驶场景实时目标检测算法[J]. 电子测量技术,2024,47(14):88-95.
SHEN Y, WANG Q, WU H. FSSD-DETR real-time object detection algorithm for autonomous driving scenarios [J]. Electronic Measurement Technology, 2024, 47(14): 88-95.
- [3] ZHAO K, HAN Q, ZHANG C B, et al. Deep Hough transform for semantic line detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(9): 4793-4806.
- [4] 李宁,肖昊. 基于 FPGA 的稀疏卷积神经网络加速器设计[J]. 电子测量技术,2024,47(5):1-8.
LI N, XIAO H. Accelerator design of sparse convolutional neural network based on FPGA [J]. Electronic Measurement Technology, 2024, 47(5): 1-8.
- [5] 宋倩男,刘光柱,武乐林,等. 基于 Sobel 的 FPGA 图像边缘检测系统设计[J]. 电子测量技术,2024,47(13):68-73.
SONG Q N, LIU G ZH, WU L L, et al. Design of FPGA image edge detection system based on Sobel [J]. Electronic Measurement Technology, 2024, 47(13): 68-73.
- [6] GABRIELLI A, ALFONSI F, DELCORSO F. Simulated Hough transform model optimized for straight-line recognition using frontier FPGA devices [J]. Electronics, 2022, 11(4): 517.
- [7] EIHAJJOUJI I, MARS S, ASRIH Z, et al. A novel FPGA implementation of Hough transform for straight lane detection [J]. Engineering Science and Technology an International Journal, 2020, 23 (3): 274-280.
- [8] 陈荣林. 条形码定位与识别的算法设计和硬件实现[D]. 福州:福州大学,2022.
CHEN R L. Algorithm design and hardware implementation of bar code localization and recognition[D]. Fuzhou: Fuzhou University, 2022.
- [9] LAM D K, DINH P T L, NGUYEN T N D. Hardware-based lane detection system architecture for autonomous vehicles [J]. Lecture Notes in Electrical Engineering, 2023, 188: 34-44.
- [10] ALFONSI F. Flexible Hough transform FPGA implementation for the ATLAS event filter [C]. Proceedings of the CTD 2023. Geneva, Switzerland, 2023.
- [11] BAILEY D. Considerations for hardware Hough transforms [J]. IEEE Transactions on Image Processing, 2011, 20(120): 120-125.
- [12] LIU W F, ZHANG ZH Q, LI S, et al. Road detection by using a generalized Hough transform[J]. Remote Sensing, 2017, 9(6): 590.
- [13] 刘强. 新型阵列屏体的显示缺陷视觉检测技术研究[D]. 哈尔滨:哈尔滨工业大学,2020.
LIU Q. Research on visual inspection technology for display defects of new array panel [D]. Harbin: Harbin Institute of Technology, 2020.
- [14] ZHANG Y C, ZHANG Y, GONG J. A LCD screen mura

- defect detection method based on machine vision[C]. 2020 Chinese Control And Decision Conference (CCDC 2020). Hefei: Hefei University of Technology Press, 2020: 4618-4623.
- [15] GUAN J G, AN F W, ZHANG X Y, et al. Real-time straight-line detection for XGA-Size videos by Hough transform with parallelized voting procedures [J]. Sensors, 2017, 17(2): 270.
- [16] ZHOU X, ITO Y, NAKANO K. An efficient implementation of the gradient-based Hough transform using DSP slices and block RAMs on the FPGA [J]. IEEE Transactions on Industrial Informatics, 2014, 10(3): 762-770.

作者简介

蒋晨鑫,硕士研究生,主要研究方向为图像处理硬件加速器技术、FPGA 开发。

E-mail:2834497746@qq.com

肖昊(通信作者),博士,教授,主要研究方向为人工智能与安全芯片,硬件加速器等。

E-mail:xiaohao@hfut.edu.cn

徐瀚,硕士研究生,主要研究方向为图像压缩处理硬件加速器技术。

E-mail:2023171206@mail.hfut.edu.cn

朱骄阳,硕士研究生,主要研究方向为数据安全与硬件加速。

E-mail:2023171218@mail.hfut.edu.cn