

文章编号: 1001-3806(2017)05-0769-06

基于 FPGA 的光纤光斑中心定位算法研究

张 晴, 吴友宇*

(武汉理工大学 信息工程学院, 武汉 430070)

摘要: 为了解决传统数字图像处理算法中数据运算量大、复杂度高、耗时长的问题, 提出一种基于可编程门阵列 (FPGA) 光纤光斑中心定位的方法。采用数字信号处理系统, 利用开发工具 (DSP builder), 设计了光斑图像预处理算法和边缘检测算法, 用最小二乘法拟合光斑边界, 采用流水线设计, 增强了数据处理的并行能力, 提高了处理速度。在 Cyclone V 平台上进行理论分析和实验验证, 取得了光斑图像边界、中心坐标数据。结果表明, 在保证对光斑中心定位的绝对误差小于 0.1 pixel 的条件下, 使用 FPGA 比计算机运算速度能提高 21 倍以上。该研究能够在 FPGA 平台上快速准确定位光斑中心。

关键词: 图像处理; 中心定位; 最小二乘法; 光纤光斑

中图分类号: TN911.73; TN47

文献标志码: A

doi: 10.7510/jgjs.issn.1001-3806.2017.05.030

Locating algorithm of optical fiber spot center based on FPGA

ZHANG Jing, WU Youyu

(School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China)

Abstract: In order to solve the problem of large amount of data, high complexity and long time consuming in traditional digital image processing algorithm, a method for center location of optical fiber spot was presented based on field-programmable gate array (FPGA). By using digital signal processing system and development tools (DSP builder), image preprocessing algorithm and edge detection algorithm were designed. The least square method was used to fit the boundary of light spot. The parallel capability for data processing was enhanced and the processing speed was improved by means of pipeline design. Theoretical analysis and experimental verification were carried out on Cyclone V platform. The image boundary and the central coordinate data were obtained. The results show that under the condition that the absolute error of spot center location is less than 0.1 pixel, the computer speed can be improved by more than 21 times based on FPGA. The spot center can be located quickly and accurately on FPGA platform.

Key words: image processing; spot center detection; least square method; optical fiber spot

引言

随着科学技术的进步, 光纤被广泛应用在各个领域, 光纤光斑中心位置检测是光电测量手段中的关键技术, 其准确性、精度和抗干扰性直接影响测量结果。目前光斑中心定位算法主要有 Hough 变换算法、重心法和拟合法。鉴于 Hough 变换算法需要逐点进行投票、记录, 花费时间较长; 重心法则要求光斑图像光强分布均匀, 且其受光斑形状的影响较大, 会产生误差; 而拟合法时间复杂度较小、运算精度高^[1], 但容易受噪声影响。

随着集成电路的发展, 可编程门阵列 (field-pro-

grammable gate array, FPGA) 器件有着以下特点: 优秀的可编程能力、并行执行能力、硬件资源丰富、低功耗。因此 FPGA 被广泛应用于图像处理算法的领域。现有测量激光光束质量分析的系统都已经相当成熟, 但都是通过计算机对激光光斑图像进行处理来计算光斑的参量。例如, 参考文献[2]中基于 LabVIEW 平台开发了一套快速检测激光发散角的测量系统。而参考文献[3]中将 FPGA 中采集到的激光光斑数据经 USB 线与上位机通信, 并对数据二次处理, 得到光斑的能量分布图, 并解析出光斑的中心。参考文献[4]中对 Hough 变换法、重心法以及高斯拟合算法定位光斑中心相比较, 得到高斯拟合方法效果最好, 误差小于 0.1 pixel。参考文献[5]中对传统最小二乘法拟合圆以及改进的最小二乘法拟合圆进行比较, 改进的圆拟合方法处理得到的中心坐标误差在 0.1 pixel 之内。因此, 为保证误差小于 0.1 pixel 并能够提高计算效率, 研究基于 FPGA 平台处理光斑数据的系统是非常必要的。针对

作者简介: 张 晴 (1991-), 女, 硕士研究生, 研究方向为嵌入式图像处理。

* 通讯联系人。E-mail: wuyouyu1@whut.edu.cn

收稿日期: 2016-10-21; 收到修改稿日期: 2016-12-08

这一问题,相对于传统软件进行光斑图像处理的算法,本文中提出一种基于FPGA的光纤光斑中心定位的算法,提高了运算速率,并且增加了图像预处理部分,可高效地过滤图像中的噪声,采用Sobel算子提取光斑轮廓,获得光斑边界像素坐标,再利用代数距离的最小二乘逼近光斑的轮廓,从而得到光斑的中心位置坐标,为求取光斑质量参量奠定基础。

1 基于FPGA的图像预处理

目前,常见的中心定位方法有Hough变换、重心法和最小二乘拟合^[6]。鉴于重心法和Hough变换法具有复杂度高和精度低的缺陷,因此,本文中选用最小二乘法拟合光斑边界。最小二乘法拟合算法对噪声敏感,所以,问题的根本在于提高该算法的抗干扰能力,以提高光斑中心定位的精度。

1.1 中值滤波

中值滤波是一种非线性平滑技术,通常用于处理椒盐噪声。其实现过程为:首先对图像中的某个采样窗口中的所有数据进行排序,然后用排序后的中值作为该区域的中心位置的像素值。传统的中值滤波算法运算复杂度高,故本文中采用快速中值滤波算法:在3×3的滑动窗口中,先比较每列值中的最大值、最小值、中值;再取出3个最大值中的最小值A₀,3个中值的中值A₁,3个最小值中的最大值A₂;最后取出A₀,A₁,A₂的中值即为最终结果。与线性滤波相比,中值滤波可以保护图像的边缘信息并且有效滤除噪声。经滤波以后的图像如图1所示。

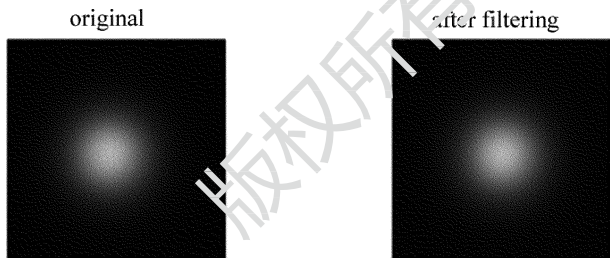


Fig. 1 Spot image before and after filtering

1.2 边缘检测

图像经中值滤波后,需对图像进行阈值分割,即二值化处理——选取适当的阈值^[7],用0或255将灰度图像分成黑与白两种颜色,本文中选取的阈值为图像中灰度最大值的一半,而后采用边缘检测算法提取光斑的轮廓。Sobel算子与Prewitt算子相比,对像素位置的影响做了加权,降低了噪声的影响,用其进行目标光斑的边缘检测。换言之,同时测量水平方向和竖直方向的灰度差,将得到的数据叠加。经Sobel检测后,会得到所需的边缘像素坐标,再对数据进行处理。经

过Sobel算子边缘检测后的图像如图2所示。

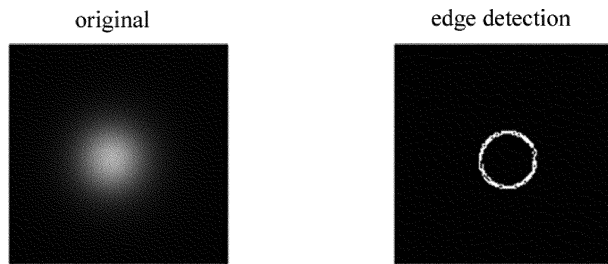


Fig. 2 Spot image before and after edge detection with Sobel operator

2 基于FPGA的圆中心定位

对图像数据进行预处理后,得到平滑的光斑边缘,进而采用曲线拟合对光斑的中心精确定位。圆可以看成是一个特殊的椭圆,本文中研究一般椭圆的光斑中心定位^[8-9]。由图像预处理得到的所有的边缘数据对圆进行最小二乘拟合,利用求解出的方程系数可直接计算出光斑的中心位置。

2.1 最小二乘曲线拟合

在2维坐标系中,平面椭圆方程的隐式方程表示为F(x,y)=0,其中a=(A,B,C,D,E,F)为椭圆方程的参量,即椭圆隐式方程为:

$$f(x,y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F \quad (1)$$

式中,f(x,y)为椭圆方程,其中x和y分别为椭圆上点的横纵坐标值;A,B,C,D,E,F为椭圆方程参量。

比较常见的椭圆拟合方法有基于代数距离的最小二乘法。由最小二乘法可知,椭圆的拟合可以近似看成是求光斑边缘上的i个数据点(x_i,y_i)到曲线f(x,y)=0的残差平方和的最小值,即求目标函数f(B,C,D,E,F) = ∑_{i=1}^m (x_i² + Bx_iy_i + Cy_i² + Dx_i + Ey_i + F)²的最小值。再由极值原理,欲使f(B,C,D,E,F)值为最小,必有∂f/∂B = ∂f/∂C = ∂f/∂D = ∂f/∂E = ∂f/∂F = 0。由此可得一个线性方程组,将其转换成求线性矩阵方程Ax=b的解。这里A为5×5维实矩阵,x为待求5维列向量,b为n维实向量。

$$\begin{bmatrix} \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i y_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m x_i y_i \\ \sum_{i=1}^m x_i y_i^3 & \sum_{i=1}^m y_i^4 & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m y_i^3 & \sum_{i=1}^m y_i^2 \\ \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m y_i^3 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i \\ \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m x_i & \sum_{i=1}^m y_i & \sum_{i=1}^m 1 \end{bmatrix} \begin{bmatrix} B \\ C \\ D \\ E \\ F \end{bmatrix} =$$

$$\begin{bmatrix} -\sum_{i=1}^m x_i^3 y_i \\ -\sum_{i=1}^m x_i^2 y_i^2 \\ -\sum_{i=1}^m x_i^3 \\ -\sum_{i=1}^m x_i^2 y_i \\ -\sum_{i=1}^m x_i^2 \end{bmatrix} \quad (2)$$

式中, (x_i, y_i) 为光斑边界上点的坐标, $(x_i, y_i) \in E$ 表示光斑图像中边缘上所有的点的集合。通过求解该线性方程组, 便可得到椭圆方程参量 B, C, D, E 和 F , 则中心坐标 (x_c, y_c) 为:

$$\begin{cases} x_c = \frac{BE - 2CD}{4C - B^2} \\ y_c = \frac{BD - 2E}{4C - B^2} \end{cases} \quad (3)$$

2.2 线性最小二乘问题求解

在数字信号处理(digital signal processing, DSP)系统设计开发工具中以高斯消元法为基础对线性方程组进行求解, 鉴于 DSP builder 的除法模块只能计算出被除数和除数相除以后的商和余数, 故而采用移位寄存器将被除数移位倍增, 再进行相除, 经验证, 其精度远远小于单精度浮点数, 计算的结果与理论比较误差大于 10%, 因此选用 Quartus II 的单精度浮点运算知识产权(intellectual property, IP)核, 运用 IP 核设计运算处理单元。

一般地, 线性最小二乘问题的求解主要有迭代法和矩阵分解法两类。其中矩阵分解法可以最大限度实现并行计算, 从而更适用于 FPGA 实现^[10]。矩阵分解的方法主要有 QR 分解^[11-12]、LU 分解^[13]、奇异值分解^[14]以及 Cholesky 分解^[15]。QR 分解的计算过程复杂; 而奇异值分解与 QR 分解的计算难度相当; 大规模矩阵一般采用 LU 分解来进行分块分解, 但是 LU 分解过程中需要大量的乘法和除法运算, 会造成一定的延时, 并且消耗过多的硬件资源; LDL^T 分解适用于对称的正定矩阵, 分解过程中避免了开方运算, 其计算的复杂度仅为 LU 分解的一半, 故而本文中采取 LDL^T 分解算法来实现矩阵的分解。

2.2.1 基于 LDL^T 分解的线性方程求解原理 LDL^T 分解适用于对称的正定矩阵, 对任意一个对称正定矩阵 A , 将其分解成 $A = LDL^T$ 的形式, 其中 L 为对角线为 1 的下三角矩阵, D 为对角阵, L^T 为 L 的转置, 即:

$$A = LDL^T = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \dots & \dots & \dots & \dots & \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} d_1 & & & & \\ & d_2 & & & \\ & & d_3 & & \\ & & & \dots & \\ & & & & d_n \end{bmatrix} \begin{bmatrix} 1 & l_{21} & l_{31} & \dots & l_{n1} \\ & 1 & l_{32} & \dots & l_{n2} \\ & & 1 & \dots & \dots \\ & & & 1 & \dots \\ & & & & 1 \end{bmatrix} \quad (4)$$

矩阵 D 及 L 中的元素 d_j 和 l_{ij} 根据下式计算:

$$\begin{cases} d_j = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_k \\ l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}}{d_j} \end{cases} \quad (5)$$

式中, $j = 1, 2, \dots, n; i = j + 1, j + 2, \dots, n$ 。 a_{ij} 和 a_{ji} 分别为矩阵 A 中的第 i 行 j 列和第 j 行 i 列所对应的元素, d_j 和 d_i 为对角阵 D 中对角线上所对应的元素 ($k = j - 1$), l_{ij}, l_{ik} 和 l_{jk} 分别为矩阵 L 中第 i 行 j 列、第 i 行 k 列和第 j 行 k 列所对应的元素。

分解后即将求解的线性方程组 $Ax = b$ 转换为求 $LDL^T x = b$ 的解, 等同于求解 3 个三角线性方程组。

$$\begin{cases} Lz = b \\ Dr = z \\ L^T x = r \end{cases} \quad (6)$$

式中, z 和 r 为引入的 n 维待求列向量。根据以上步骤即可求得方程的解向量 x 。

2.2.2 LDL^T 算法数据依赖关系分析 为了实现在 FPGA 硬件平台上高效的矩阵分解, 必须合理采用流水线技术, 设计并行计算^[16]。算法并行化的根本是数据之间的依赖关系, 下面将分析 LDL^T 分解算法的数据之间的关系。

根据(5)式, 每一个内层循环用于计算矩阵 L 的一列元素和矩阵 D 对应的一个元素, 以第 j 个内层循环中计算的具体数据依赖关系举例说明。将(5)式展开如下所示:

$$\begin{cases} d_j = a_{jj} - [l_{j1}^2 d_1 + l_{j2}^2 d_2 + \dots + l_{j(j-1)}^2 d_{(j-1)}] \\ l_{ij} = [a_{ij} - (l_{i1} d_1 l_{j1} + l_{i2} d_2 l_{j2} + \dots + l_{i(j-1)} d_{(j-1)} l_{j(j-1)})] / d_j \end{cases} \quad (7)$$

由(7)式可知, 矩阵 D 中元素 d_j 的计算依赖于矩阵 A 中的元素 a_{jj} 、矩阵 L 的第 j 行元素以及矩阵 D 的前 $(j - 1)$ 个元素。 L 矩阵的元素 l_{ij} 的计算依赖于矩阵

A 中的元素 a_{ij} 、矩阵 D 中的元素 d_j 以及矩阵 L 的第 i 行元素。而 l_{ij} 与 $l_{(i+1)j}$ 之间的计算没有依赖关系,且 d_j 和 l_{ij} 的计算都要计算中间结果 $l_{jk}d_k$ 。

通过分析数据依赖关系可知, LDL^T 分解算法的外层数据循环计算需要依存已经获得的数据,内层数据循环计算则没有依赖关系,直接并行计算。

2.2.3 LDL^T 分解模块设计 通过对 LDL^T 分解算法及相关数据分析,可设计如图 1 所示的分解结构框图。分解模块主要分为计算对角阵 D 元素的单元 PED、下三角阵 L 元素的单元 PEL 和数据选择器, PED 与 PEL 共享 PED 的计算结果 $l_{jk}d_k$,所以可以将此结果缓存于先入先出缓存器(first in, first out, FIFO)中的直接用于 l_{ij} 的计算,从而避免重复运算,节省资源。PED 的输入 l_{jk} 通过四选一数据选择器,按照一定的顺序将 4 组存

储矩阵 L 中元素的随机存取存储器 (random access memory, RAM) 中的数据读出,用于计算 d_j 。矩阵 A 的数据通过数据分配器从外部传输到 PEL 与 PED。其中,矩阵中 a_{ij} 与 d_j, l_{ij} 各自关联,但是 PED 与 PEL 计算存在延迟,故将数据存于 FIFO_0 ~ FIFO_4 中。当前 l_{ij} 的计算结果与当前 d_j 值关联很大,且 l_{ij} 的计算还会用到计算 d_j 时的中间值 $l_{jk}d_k$,所以 PED 要在 PEL 之前执行。在理想的情况下,为了实现运算速度最快,PEL 的数量越多,效率越高。输入矩阵 A 为 5×5 的矩阵,在每一个内层循环中,需要计算的矩阵 D 的对角线元素 d_j 仅为 1 个,采用一个 PED 单元即可;而计算的矩阵 L 的元素 l_{ij} 的个数最多为 4 个,且互不依赖,故采用 4 个 PEL 并行计算。分解模块具体结构如图 3 所示。其中 RAMD 为存储矩阵 D 中元素的随机存取存储器。

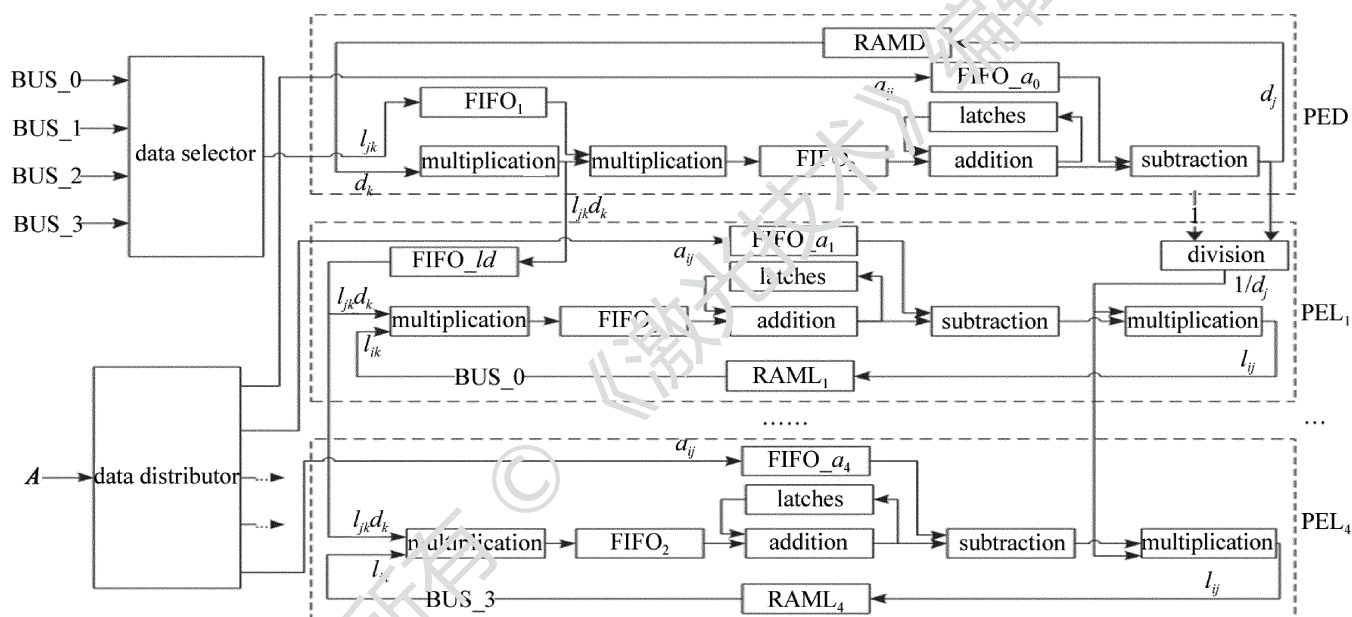


Fig. 3 Structure diagram of decomposition module

图 3 中, PED 和 PEL 模块均采用流水线设计方案,运算单元主要包括乘法器 (multiplication)、除法器 (division)、加法器 (addition) 和减法器 (subtraction),均使用 Quartus II 提供的 IEEE 754 规范的浮点数运算 IP 核实现。

在设计 PED 和 PEL 模块时,鉴于乘法器在计算时存在固有延时,所以将输入的数据 l_{jk} 写入 FIFO_1 中实现缓存,此时 FIFO_1 就作为第 2 个乘法器的输入端,与第 1 个乘法器的计算结果进行第二次乘法运算。由于 $l_{jk}d_k$ 为 PED 和 PEL 计算的共有部分,为减少乘法器的使用,故将计算中间结果 $l_{jk}d_k$ 存于 FIFO_ld 中。累加器是由加法器和锁存器组成,每隔加法器计算的延时时间就将 FIFO_2 中数据读出一个,累加结束后,将结果传递给减法器输入端,并使能减法器。最终的计算结果存入 RAM 中。内层循环结束后,循环计算 5 次,即

可获得对角阵 D 和下三角阵 L 矩阵的元素 d_j 和 l_{ij} 。

2.2.4 求解模块的设计 矩阵分解结束后,获得下三角阵 L 和对角阵 D ,根据 $(LDL^T)x = b$,可以等价成 3 个简单的方程组 $Lz = b, Dr = z$ 和 $L^T x = r$,由此可得求解公式:

$$\begin{cases} z_m = b_m - \sum_{p=1}^{m-1} l_{mp}z_p, (m = 1, 2, \dots, q) \\ r_m = \frac{z_m}{d_m}, (m = 1, 2, \dots, q) \\ x_m = r_m - \sum_{u=m+1}^q l_{um}x_u, (m = q, q-1, \dots, 1) \end{cases} \quad (8)$$

式中, l_{mp} 和 l_{um} 分别表示矩阵 L 中第 m 行 p 列和第 u 行 m 列所对应的元素; d_m 表示矩阵 D 对角线上对应的元素; z_m, r_m, x_m, b_m 分别代表列向量 z, r, x, b 的第 m 个元素; z_p, x_u 分别代表列向量 z 的第 p 个元素和列向量 x

的第 u 个元素; q 为待求向量的维数($q=5$)。

求解模块要用到 PEL 模块中计算的数据 l_{ij} ,故每算出一个 l_{ij} 的值,求解模块中就要有一个计算单元与之相对应。求解模块的结构框图与 PED 和 PEL 的框图基本相同。

3 实验结果及分析

3.1 实验结果及误差分析

本文中基于 Quartus II 14.0 的开发环境,将上述设计在 Altera 公司的 DE1-SOC 开发板上进行了验证。鉴于作者无法准确地得到光斑图像真实的中心,因此通过 MATLAB 程序生成一幅人工光斑图像,并附着一些噪声及干扰,该光斑图像的理论中心坐标为 (150, 166),分别在 FPGA 平台和计算机 Visual Studio (VS) 环境下利用本文中的算法对光斑进行中心定位,根据 (3) 式计算的最终结果对比如表 1 所示。其中,FPGA 的工作时钟设定为 50MHz,并采取 IEEE 754 规范下的单精度浮点制 IP 核来实现设计中的各浮点运算。

Table 1 Deviation between FPGA and PC

	coordinate of center	absolute deviation
visual studio	(150.06907,166.097)	(0.06907,0.097)
FPGA	(150.06881,166.048)	(0.06881,0.048)

由表 1 可知,本方法对圆的中心定位的精度小于 0.1pixel,但 FPGA 平台与计算机 VS 环境下实现相同的算法计算的结果有差异,出现此结果的原因是,在 FPGA 平台使用单精度浮点 IP 核进行计算时,首先要将数据转换为单精度浮点数(32bit),但对于较大的数据,尾数部分会有部分省略,进而对结果产生一定的误差。不过两个平台计算的中心位置的误差都小于 0.1pixel,所以该方法保证了中心定位算法的精度。

3.2 计算性能分析

保证中心定位算法精度的前提下,为测试方法的效率,将实现 LDL^T 解最小二乘问题的方法在 FPGA 平台和计算机 VS 下进行了计算效率比较。具体结果如表 2 所示。

Table 2 Processing time

	FPGA	visual studio	proportion
decomposition/ μ s	36.38	620	1/17.04
solving/ μ s	37.06	995	1/26.85
total/ μ s	73.44	1615	1/21.99

由表 2 可知,在 FPGA 平台下实现解 5 维最小二乘问题时,本文中方法在保证精度的前提下,可达到相对于计算机 21 倍以上的速度。

4 结 论

光斑中心位置检测是光纤光束测量系统中的一个

关键步骤。针对现有的基于计算机平台实现光斑参量测量实时性差,数据吞吐量不高的问题,本文中提出一种基于 FPGA 光纤光斑中心定位的方法:先对光斑图像进行中值滤波、二值化和边缘提取等预处理,提高算法的抗干扰能力,而后在 FPGA 平台上将获得的光斑边界数据进行分析处理。实验结果表明,在单精度浮点制条件下,本文中方法对光斑中心定位的绝对误差小于 0.1pixel,在保证精度的同时可以实现 21 倍以上的速度提升。

参 考 文 献

- [1] FAN Ch L, LIU Ch, DING G, et al. A method of circle curve fitting based on the cumulative error of the radius error[C]// International Conference on Computational Intelligence and Security. New York, USA: IEEE, 2015:211-214.
- [2] WANG Q Q, LIU J, PENG Zh, et al. Measurement system for laser divergence angle based on LabView [J]. Chinese Journal of Lasers, 2012, 39(11):1100-1103 (in Chinese).
- [3] ZHANG L. System of measuring laser spot[D]. Xi'an :Xidian University, 2010:19-15 (in Chinese).
- [4] WANG L, HU Zh W, JI H X. Laser spot center location algorithm based on Gaussian fitting[J]. Journal of Applied Optics, 2012, 33(5):985-990 (in Chinese).
- [5] LIU H L, HOU W, FAN Y L, et al. An improved algorithm of laser spot center location[J]. Computer Measurement & Control, 2014, 22(1):139-141 (in Chinese).
- [6] LAN Zh L, YANG X F. Practical improvement of laser spot center location algorithm[J]. Computer Engineering, 2008, 34(6):7-9 (in Chinese).
- [7] NING Sh N, ZHU M, SUN H H, et al. Realization of improved sobel adaptive edge detection algorithm based on FPGA[J]. Chinese Journal of Liquid Crystals and Displays, 2014, 29(3):395-402 (in Chinese).
- [8] SHI Y, CHENG X. Laser spot center detection based on the geometric feature[C]// International Symposium on Information Science & Engineering. New York, USA: IEEE, 2010:322-325.
- [9] LIU J S, YUAN S C. Subpixel location algorithm for circular targets center based on Zernike moments and curvature[J]. Computer Engineering & Applications, 2010, 46(29):153-156.
- [10] ZHONG X J. Research on FPGA calculation method of typical matrix decomposition[D]. Harbin :Harbin Institute of Technology, 2012:14-66 (in Chinese).
- [11] TAI Y G, LO C T D, PSARRIS K. Applying out-of-core QR decomposition algorithms on FPGA-based systems[C]//International Conference on Field Programmable Logic and Applications. New York, USA: IEEE, 2007:86-91.
- [12] HAN B, YANG Z, ZHENG Y R. FPGA implementation of QR decomposition for MIMO-OFDM using four CORDIC cores[C]//IEEE International Conference on Communications. New York, USA: IEEE, 2013:4556-4560.
- [13] WU G M, DOU Y, PETERSON G D. Blocking LU decomposition for FPGAs[C]// IEEE International Symposium on Field-Programmable Custom Computing Machines. New York, USA: IEEE, 2010:109-112.
- [14] TANG B P, JIANG Y H, ZHANG X Ch. Feature extraction method of rolling bearing fault based on singular value decomposition-mor-

- phology filter and empirical mode decomposition[J]. Chinese Journal of Mechanical Engineering, 2010, 46(5):37-42(in Chinese).
- [15] WU G M, DOU Y, WANG M. A fine-grained parallel algorithm for the Cholesky decomposition [J]. Computer Engineering & Science, 2010, 32(9):102-106(in Chinese).
- [16] GUO L, TANG Y H, ZHOU J, *et al.* Research on parallel architecture for LDL^T decomposition -processor[J]. Computer Engineering, 2011, 37(21):241-243(in Chinese).

版权所有 © 《激光技术》编辑部