

文章编号 1004-924X(2015)10-2909-10

大尺寸动态视觉测量系统的并行加速

董明利^{1*}, 许晓臣¹, 王 君¹, 孙 鹏^{1,2}, 燕必希¹

(1. 北京信息科技大学 光电测试技术北京市重点实验室, 北京 100192;

2. 北京邮电大学 信息光子学与光通讯研究院, 北京 100876)

摘要: 由于现有以大数据量和计算量为基础的大尺寸动态视觉测量系统处理速度较慢, 本文建立了一个高速大尺寸动态视觉测量系统, 并对该系统涉及的特征点中心定位、编码点识别、相机定向等算法进行了并行化研究。首先, 分析了在不同测量条件下各个主要算法的时间消耗情况及每个主要算法的并行性; 然后, 对常规的特征点中心定位和编码点识别算法做了介绍, 分别提出了特征点中心并行快速定位和编码点并行快速识别算法, 并详细说明了这两种并行快速算法的实现原理。最后, 针对大量原子操作的问题, 提出了线程束集体原子操作的优化方法。实验结果表明: 在不损失定位精度和识别率的前提下, 图像中包含 300 个点时的并行方案比串行方案的时间开销减少了 42%, 当点数达到 20 000 时, 时间开销减少 91% 以上。实验显示提出的并行设计方案有效地提高了处理速度, 解决了大尺寸动态视觉测量系统实时性差的问题。

关键词: 动态视觉测量; 并行计算; 编码点识别; 特征点定位; 大尺寸测量

中图分类号: TP242.6; TP391 **文献标识码:** A **doi:** 10.3788/OPE.20152310.2909

Parallel acceleration of large scale dynamic vision measurement system

DONG Ming-li^{1*}, XU Xiao-chen¹, WANG Jun¹, SUN Peng^{1,2}, YAN Bi-xi¹

(1. Beijing Key Laboratory of Optoelectronic Test Technology, Beijing Information Science & Technology University, Beijing 100192, China;

2. Institute of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China)

* Corresponding author, E-mail: dongml@sina.com

Abstract: As existing large scale dynamic vision measurement systems based on large amount of data processing and computing have a lower data processing speed, a high speed and large scale dynamic vision measurement system was proposed. The parallel optimization techniques involving in the system such as target locating, code detecting, camera orientation and other systematic algorithms were re-searched. Firstly, time consumptions of corresponding key algorithms under different considerations were analyzed. Then, the traditional target centroid and code detecting algorithms were introduced, their parallelisms were analysis, and the fast target center location algorithm and code recognition algorithm under the general parallel architecture were put forward. Moreover, the implement methods of the two parallel algorithms were explained in detail. Finally, the Warp Atom Operation Optimiza-

收稿日期: 2015-02-10; 修订日期: 2015-04-27.

基金项目: 国家自然科学基金资助项目 (No. 51175047, No. 51475046)

tion (WAOO) method for massive atomic operations was proposed. The experimental results under the same location precision and recognition rate show that the processing time is reduced by 42% and 91% for 300 and 20 000 targets respectively when the parallel algorithm is used to replace the traditional serial algorithm. The algorithms proposed in this paper are verified effective in accelerating the processing speed and improving the real-time problem in large scale vision measurement systems.

Key words: dynamic vision measurement; parallel computing; code recognition; target location; large scale measurement

1 引言

作为一种非接触测量手段,动态视觉测量技术具有操作简单、精度高及测量不受被测物表面差异影响等优点。随着工业制造技术的快速发展,对大尺寸被测物实现高精度、高速度的动态视觉测量的需求越来越迫切,如大尺寸雷达天线面形变化的动态测量、飞机及其结构部件在风洞实验中的振动分析等^[1-3]。不过,当使用千万级像素的高分辨率相机采集密集点云图像时,目前的大尺寸动态视觉测量系统在满足高精度的前提下,处理速度一般小于 3 Hz,很难实现对实时性要求很高的动态测量。大尺寸动态视觉测量系统的主要算法包括特征点中心定位、编码点识别、相机定向、对应点匹配和束调整等,其中特征点中心定位和编码点识别消耗的时间占整个测量过程的 66%以上,随着图像中特征点数量的增加,其所占比例还会明显增多,因此提高这两部分的处理速度成为提高测量速度的关键。

国外在视觉测量方面的研究起步较早并推出了动态视觉测量系统,如美国的 V-STARS Dynamo 系统采用 500 万像素相机时标称相对精度可达 1/80 000,最高工作频率为 10 Hz;德国 Gom 公司的 ARAMIS 系统采用 200 万像素相机时标称相对精度可达 1/20 000,最高工作频率为 15 Hz。但是当采用千万级像素高分辨率相机测量密集点云时,现有测量系统的处理速度一般都很难满足动态测量的需求。也就是说,要想在高精度的前提下提高处理速度,必须寻求一种新技术。由于 CPU 频率的限制,短时间内无法实现串行处理速度的大幅度提升,因此采用并行处理的方式来提高处理速度是一个很好的选择。统一计算架构(Compute Unified Device Architecture, CUDA)是美国 NVIDIA 公司 2007 年正式推出的一

种通用并行计算架构,它通过 CPU 直接调用 GPU 的计算资源,使数据密集型和计算密集型的部分在 GPU 上开展并行处理,可有效提升程序的处理速度。目前国内外的很多学者已经在很多领域进行了并行计算的研究^[4-6],国外研究人员已经在语言模型训练、霍夫变换和流场仿真等方面实现了并行计算,并获得了非常理想的效果。虽然国内很少有针对视觉测量中并行计算加速问题的相关研究,但已经有学者针对其他领域的并行计算进行了研究^[7-10],如分别将液晶波面数值分析、图像恢复和自适应耦合分辨模型等问题在 CUDA 架构上实现,且都取得了很好的效果。这些研究都证实了将并行计算应用到具体并行性问题上能够有效提高程序的处理速度。

CUDA 并行架构为研究人员提供了一种新的并行计算理念,本文基于 CUDA 架构提出了大尺寸动态视觉测量系统的并行加速算法,分别对图像处理 and 图像识别两个最耗时部分实现了并行处理。

2 大尺寸动态视觉测量系统并行方案

大尺寸动态视觉测量系统是一套高精度、高可靠性的测量系统。近年来相机分辨率的升高和算法复杂度的提升,导致测量过程中的数据量和计算量都有了巨大的增加,最终严重影响了测量系统的处理速度。针对此,本文提出了特征点中心并行快速定位算法和编码点并行快速识别算法。

2.1 问题提出

大尺寸动态视觉测量系统由高质量的 CCD、高精度的基准尺及回光反射点等附件和系统软件组成。在系统硬件确定的情况下,测量系统的精度和速度主要取决于软件算法,其主要包括特征点中心定位、编码点识别、相机定向、同名点匹配

和束调整 5 个主要算法。在 i7-3930K CPU 平台上使用 1600 万像素的相机时,在保证高精度的前提下,测量方案中各串行算法消耗的时间占系统总时间的情况如表 1 所示。

表 1 串行方案各部分的时间消耗情况

Tab. 1 Time-consuming of each step in serial program

特征点 点数	点中心定 位(%)	编码点 识别(%)	相机定 向(%)	同名点 匹配(%)	束调 整(%)	总时间 /ms
270	63.3	3.3	13.3	13.4	6.7	448.5
5000	23.4	69.2	2.9	3.0	1.5	5119.4
10000	11.2	86.1	1.0	1.1	0.6	14971.7
20000	5.0	94.2	0.3	0.3	0.2	52860.9
30000	3.2	96.4	0.1	0.2	0.1	112638.0

由表 1 分析可知:在串行方案中,点中心定位和编码点识别算法消耗时间占总时间的 66% 以上。随着特征点数的增加,编码点识别消耗的时间呈指数级增加,点中心定位消耗的时间也有明显的增加。不过,即使点中心定位消耗时间增多,由于总时间急剧增加,其相对于总时间所占比例反而有所下降。而相机定向、同名点匹配和束调整等部分由于仅依赖特征点坐标或编码点坐标的计算结果,故它们消耗的时间只占整个测量系统的很少一部分,即使特征点数增多,但由于计算量一定,其消耗的时间几乎不变,对测量系统的实时性也不会造成影响^[11]。因此只要提升点中心定位和编码点识别算法的处理速度,就可以有效提高系统的实时性。

2.2 问题分析

高分辨率或者包含密集点云的图像,点中心定位和编码点识别算法处理的数据量和计算量巨大,严重影响了测量系统的处理速度,下面详细分析这两种串行算法的特性。

2.2.1 特征点中心定位算法

特征点中心坐标是整个测量系统的数据来源,定位精度直接关系到测量的最终精度。特征点中心定位算法一般分为基于边缘拟合的方法和基于灰度信息的方法两类。相关学者研究发现,采用基于灰度信息的方法求取的精度一般要优于基于边缘拟合的方法,本文采用变权重灰度质心法来定位特征点中心坐标^[12]。

变权重灰度质心法由传统的灰度质心法改进而来,算法最终的计算公式如式(1)所示:

$$x_0 = \frac{\sum_{i=h_1}^{h_2} \sum_{j=w_1}^{w_2} x_i (g(x_i, y_j) - \beta)^\alpha}{\sum_{i=h_1}^{h_2} \sum_{j=w_1}^{w_2} (g(x_i, y_j) - \beta)^\alpha}$$

$$y_0 = \frac{\sum_{i=h_1}^{h_2} \sum_{j=w_1}^{w_2} y_i (g(x_i, y_j) - \beta)^\alpha}{\sum_{i=h_1}^{h_2} \sum_{j=w_1}^{w_2} (g(x_i, y_j) - \beta)^\alpha} \quad (1)$$

式中: $g(x_i, y_j)$ 是 (x_i, y_j) 处像素的灰度值, (x_0, y_0) 是定位点的中心坐标, β 为灰度的偏移量, α 为加权指数,两者的取值范围都是正实数。根据研究知^[13],当该偏移量 β 等于图像噪声的期望值时能获得最佳的定位效果,由于测量系统使用了高质量的 CCD 再加上后期的预处理可将背景噪声基本控制为零,故可以令 $\beta=0$ 。目前尚没有 α 最优值的理论分析,经过实验验证 α 由测量场的一些相关因素决定,如角度、光照、标志点分布。经过实验的方法测试得到:当 α 为 2.0 时,点中心的计算结果残差均值较低,故本文令 $\alpha=2.0$,则式(1)变为:

$$x_0 = \frac{\sum_{i=h_1}^{h_2} \sum_{j=w_1}^{w_2} x_i g(x_i, y_i)^{2.0}}{\sum_{i=h_1}^{h_2} \sum_{j=w_1}^{w_2} g(x_i, y_i)^{2.0}}$$

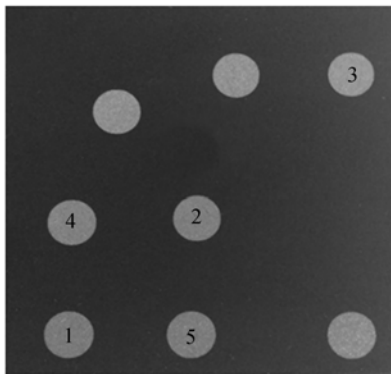
$$y_0 = \frac{\sum_{i=h_1}^{h_2} \sum_{j=w_1}^{w_2} y_i g(x_i, y_i)^{2.0}}{\sum_{i=h_1}^{h_2} \sum_{j=w_1}^{w_2} g(x_i, y_i)^{2.0}} \quad (2)$$

式(2)得到的特征点中心坐标定位结果通过基于直线拟合的特征点定位精度评估方法评估^[14-17],定位精度大于 0.02 pixel。

2.2.2 编码点识别算法

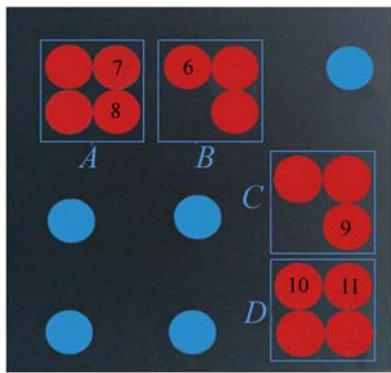
编码点是一类比较特殊的人工特征点集,它本身具有一定的几何关系。一般情况下,测量过程中的全自动靶标检测、识别和同名点匹配等都要依赖编码点。目前已经开发出来的编码点种类很多,本文采用的编码点是美国 GSI 公司 V-stars 测量系统中的点阵型编码点,如图 1 所示。

图 1(a)中,每个编码点包括 8 个点,其中 1~5 号点是 5 个基础点,以这种方式编码的编码点都有这样的 5 个基础点,并且这 5 个基础点的相对位置关系一定,另外 3 个点是码值点,每个码值



(a)点阵型编码点

(a)Code points of dot matrix type



(b)点阵编码点编码规则

(b)Encoding rule of code points for dot matrix

图 1 点阵型编码点原理图

Fig. 1 Principle of dot matrix code points

点代表的信息值不同。如图 1(b)(彩图见期刊电子版)中所示,红色点为码值点,一共有 A、B、C、D 4 组 14 个位置。为了识别和解码的可靠性,一个编码点中的 3 个码值点必须分别出现在其中的 3 组中,并且 6 号点与 7、8 号点不能同时出现,9 号点与 10、11 号点不能同时出现。

编码点识别是以每个特征点坐标为中心,在一定的半径内进行搜索,当搜寻到 8 个邻近点时,初步认为是一组候选点;当搜寻到大于 8 个点时,减小搜索半径重新搜索,直至搜索到的邻近点数小于等于 8 个。然后对候选点集进行精简,删除重复的点集,最后根据点的分布规则识别码值点的位置,以确定编码点的信息。

2.3 大尺寸动态视觉测量系统并行算法

针对大尺寸动态视觉测量系统中图像处理和图像识别这两个最耗时的部分,本文分别提出了特征点中心并行快速定位算法和编码点并行快速

识别算法。

2.3.1 特征点中心并行快速定位算法

特征点中心并行快速定位算法主要包括 4 个过程:Canny 边缘检测、感兴趣区域 (Region of Interest, ROI) 区域提取、区域约束判断和点中心计算。由于二值化区域扫描时相邻像素间具有很强的依赖关系,所以 ROI 区域提取应用并行计算很难起到加速的效果,而且经过优化的串行 ROI 区域提取所消耗的时间已经很少,故本文采用串行方式进行 ROI 区域提取。

特征点中心并行快速定位算法的流程如图 2 所示。

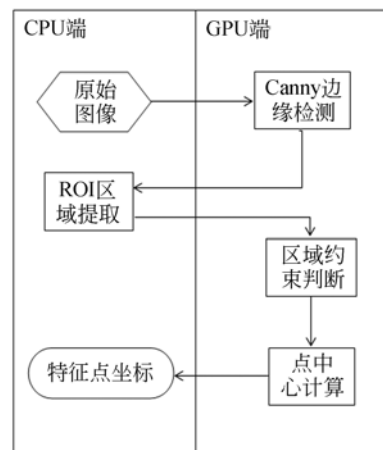


图 2 特征点中心并行快速定位算法流程图

Fig. 2 Flow chart of fast and parallel location of feature center

2.3.1.1 Canny 边缘检测并行实现

Canny 边缘检测的并行实现包括 4 个步骤:高斯低通滤波、计算梯度、非极大值抑制和边缘连接。Canny 边缘检测并行实现流程如图 3 所示。

Canny 并行检测边缘过程复杂,但是检测过程中的每步都具有很好的并行性。基于 CUDA 的 Canny 边缘并行检测实现过程如下:

(1) 把图像数据传入全局存储器;

(2) 合并访问全局存储器中的图像数据,每个线程分配 27 个寄存器,用来存储 3×3 的高斯模板数据。图像的边界数据不进行滤波,由于后期的约束判断确保目标区域不会出现在图像的边界,故其对最终的定位结果没有影响;

(3) 合并访问高斯低通滤波后的图像数据,采

用 Sobel 算子分别求像素 X 方向和 Y 方向的梯度,然后用哈密顿距离代替欧式距离求像素的梯度值;

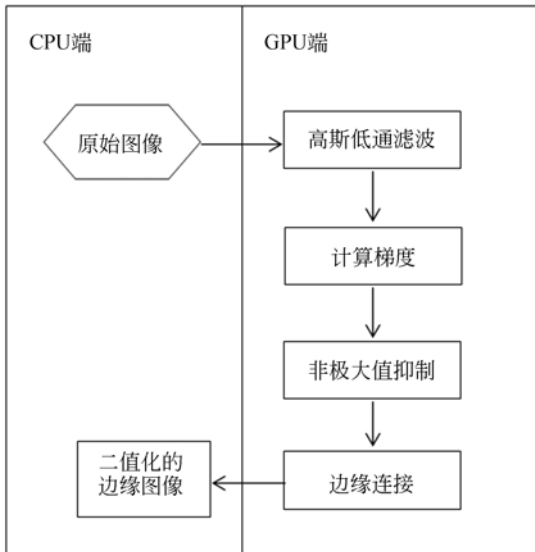


图 3 Canny 边缘检测并行实现流程图

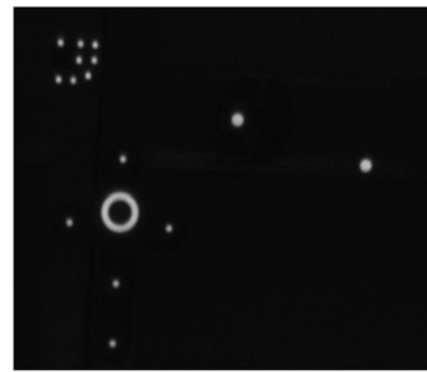
Fig. 3 Flow chart of fast Canny edge detection algorithm

(4)合并访问图像的梯度值、X 方向梯度值和 Y 方向的梯度值,计算梯度方向的角度。然后进行非极大值抑制,此处不采用线性插值的方法,而是直接采用更接近梯度方向的相邻像素的梯度值,最终得到该像素的非极大值抑制结果;

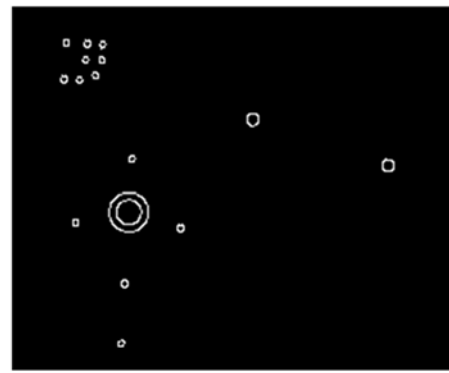
(5)合并访问非极大值抑制后的数据和图像梯度值数据,如果非极大值不等于零并且梯度值大于大阈值时即为边缘点,然后以该点为中心检测 8 邻域像素点的非极大值抑制值是否为零,若其不等于零,并且梯度值大于小阈值,则确定为边缘点。

为了提升计算性能,基于 CUDA 的 Canny 边缘并行检测对 3 个步骤做了简化,即(3)中用哈密顿距离代替欧式距离求像素的梯度值;(4)非极大值抑制时用接近梯度方向角度像素的梯度值代替线性插值;(5)中只检测了初始点 8 邻域内的点是否为边缘点。采用不同方法的边缘检测结果如图 4 所示。

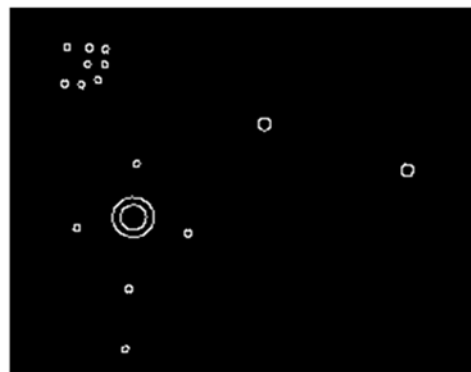
由于采用回光反射点和高质量的 CCD,测量过程中拍摄到的图像对比度很高,最终边缘检测结果即使没有完全连接,对后续的 ROI 区域提取也不会造成影响,只有在特殊的灰度分布情况下(彗差较大),提取区域的长或宽可能会偏大一个



(a)原始图像
(a)Original image



(b)OpenCV 检测结果
(b)Detection result of OpenCV



(c)并行检测结果
(c)Result of parallel detection

图 4 边缘检测结果图

Fig. 4 Edge detection results

像素。但是由于定位算法采用的是变权重灰度质心法,偏大区域中像素的灰度值已经非常小,大量实验验证显示其对最终定位结果的影响小于0.02 pixel。

2.3.1.2 区域约束判断并行实现

精确的点中心定位对后期的编码点识别、相机定向、同名点匹配等至关重要,约束判断是为了排除外界带来的干扰,确保定位的区域是有效的特征点区域。约束判断由一系列的约束条件构成,如长宽比约束、灰度约束和面积约束等。每个约束条件之间有先后顺序,但是每个区域之间的约束判断是相互独立的,因此大量的区域约束判断可以采用并行计算的方式同时进行。

并行区域约束判断流程如图 5 所示。

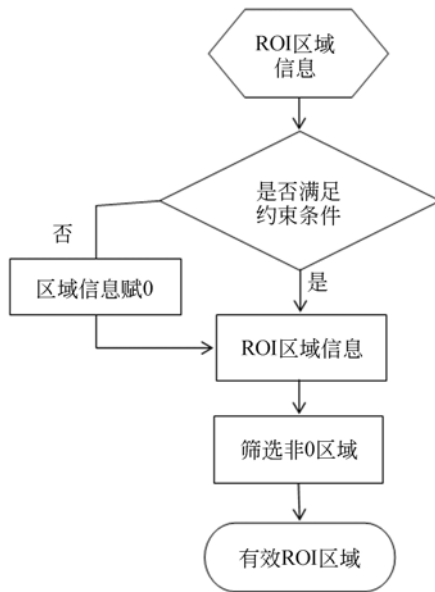


图 5 并行区域约束判断流程图

Fig. 5 Flow chart of judgment for parallel region constraint

基于 CUDA 的并行区域约束判断的实现过程如下:

(1)把 CPU 端提取的 ROI 区域信息传入全局存储器;

(2)大量的线程合并访问对应的区域信息,分别判断是否满足约束条件,不满足约束条件的区域信息赋 0,满足条件的区域信息不做处理;

(3)每个线程块在共享存储器上分配大小为 32 的数组,采用线程束集体原子操作的方法筛选判断后的区域信息,得到有效的目标区域。

并行区域约束判断有效提升了算法的处理速度。经过实验可知,当同时进行 30 000 个区域约束判断时,并行加速效果可达 28.3 倍,区域个数增多时,加速效果还会显著提升。

2.3.1.3 点中心计算并行实现

点中心计算属于典型的计算密集型问题。由于点中心计算中包含循环嵌套,导致在特征点数增多时消耗的时间急剧增加,例如串行计算 10 000 个区域的点中心时消耗的时间已经占整个算法的 50%以上。但是每个特征点之间相互独立,所以可以采用并行方法同时计算大量的点中心。

并行点中心计算是根据变权重灰度质心法的计算公式,同时计算大量特征点的中心坐标。基于 CUDA 的并行点中心计算实现过程如下:

(1)合并访问全局存储器中的 ROI 区域信息和图像预处理后的数据;

(2)执行配置参数,每个线程块分配 1 024 个线程,每个线程计算相应的 ROI 区域对应的点中心坐标,点中心的数据存储格式选用 CUDA 内部的 double4 类型进行向量化操作;

(3)特征点中心坐标传回内存。

在点中心计算中,不同区域之间具有很好的并行性,采用并行计算可以取得很好的加速效果。经过实验可知,当特征点数为 35 000 时,加速效果可以达到 214 倍,并且随着点数的增多,加速效果还会有显著提升。

2.3.2 编码点并行快速识别算法

编码点识别包括两步:搜索有效的编码点集和点集信息识别。搜索有效编码点集时,是因为以每个点坐标为中心搜索剩余的所有点,并且同一点可能不止搜索一次,所以当点数较多时,搜索有效编码点集的计算量非常庞大。由于测量过程中粘贴的编码点数量较少,这使得真正消耗在点集信息识别上的时间很少且不会随着特征点数的增多而增多,因此绝大部分时间被搜索编码点集

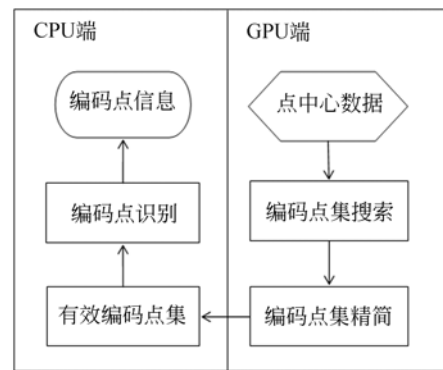


图 6 编码点快速识别算法流程

Fig. 6 Flow chart of fast recognition for code points

消耗。当图像中有 1 000 个点时,搜索点集消耗的时间占 76%,当有 10 000 个点时搜索时间占到 98%。搜索有效的编码点集时,不同点集之间相互独立,可以采用并行计算的方法提升点集的搜索速度。编码点并行快速识别算法流程如图 6 所示。

2.3.2.1 编码点集搜索并行实现

由于采用点阵分布型编码点,在合理布置测量场和拍照角度时,图像中编码点所占的区域大小相当,只需以一个点为中心搜索一定半径内数量的点,当点数恰好是 8 时,初步认为是一组候选点;当搜索到大于 8 个点时,减小搜索半径重新搜索,直至搜索到的邻近点数都小于等于 8。

基于 CUDA 的编码点集并行搜索实现过程如下:

(1)大量线程合并访问全局存储器中的点中心数据,每个线程分配大小为 20 的寄存器数组;

(2)每个线程以对应点为中心,搜索在半径为 radius 的圆内的点数,满足条件的点编号存储在(1)中分配的数组中;

(3)当每个线程搜索到的点数小于 8 时停止搜索,当搜索到的点数等于 8 时停止搜索并将 8 个点的编号信息存储到全局存储器,当搜索到的点数大于 8 时,减小搜索半径 radius 重复进行(2)中的过程,直到搜索到的点数小于等于 8 个;

(4)得到编码点候选点集。

2.3.2.2 编码点集精简并行实现

由于拍摄距离的不同导致最终像面上编码点集的尺寸不同,因此一般选择的初始搜索半径较大,这可能导致编码点的 8 个点都搜索到同一个编码点集,得到 8 个相同的候选点集,这些重复的候选点集没有任何意义并且会给后期的点集信息识别带来不必要的麻烦。本文提出了编码点集并行快速精简的方法。

编码点集并行快速精简法是使用并行计算的方法,通过采用线程束集体原子操作方法筛选出有效的、无重复的候选点集。基于 CUDA 的编码点集精简的并行实现过程如下:

(1)每个线程合并访问候选点集中的数据信息;

(2)每个线程将对应的候选点与小于等于该线程索引号的候选点进行比较,当搜索到相同的候选点并且该线程的索引号大于相同候选点的标

号时,将该线程的候选点信息赋-1;

(3)每个线程块在共享存储器上分配大小为 32 的数组,采用线程束集体原子操作的方式筛选(2)中得到的非-1 候选点,得到有效的编码点集;

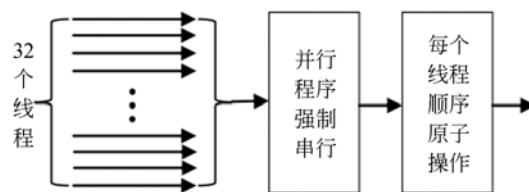
(4)将有效的编码点集信息传回内存。

3 线程束集体原子操作优化策略

在本文提出的并行算法中,当筛选有效的特征区域和编码点集精简时,为了避免写入冲突,必须采用原子操作的方式确保写入数据的正确性。原子操作能够强迫大量的线程在特定情况下按照设定的顺序执行。当并行程序中含有大量的原子操作时,原子操作将成为影响并行程序性能提高的瓶颈。针对此问题,本文提出了线程束集体原子操作的优化策略。

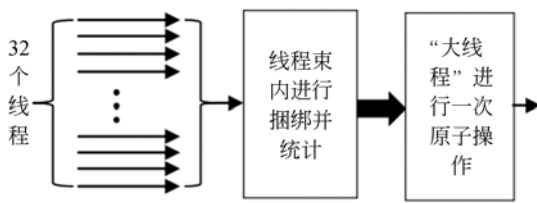
线程束集体原子操作改变了原有以线程为基本单位进行原子操作的方式,该优化策略以线程束为单位进行集体的原子操作,即将线程束中的 32 个线程人为地看作一个“大线程”,经过线程束内统计的方法得到需要进行写入操作的线程数量,然后应用一次原子操作的方法计算该线程束需要写入数据的起始地址,最后根据相应线程在线程束中的编号,结合上述的起始地址得到最终的写入地址。这种优化策略理论上可以将 32 次的原子操作缩减为 1 次,有效地减少了原子操作的数量。线程束集体原子操作的原理如图 7 所示。

在必须进行原子操作的情况下,采用本文提出的线程束集体原子操作的优化策略,最多可以降低 97%的原子操作数量,性能相对于每个线程分别采用原子操作的方法在理论上最高可以提高 32 倍,很好地解决了并行程序中由于大量原子操作所导致的性能急剧下降的问题。



(a) 每个线程进行原子操作

(a) Atomic operation for each thread



(b) 线程束集体原子操作

(b) Warp atom operation

图 7 线程束集体原子操作的原理图

Fig. 7 Schematic of warp atom operation

4 实验结果及分析

本文对所提出的大尺寸动态视觉测量系统并行算法进行了实验验证。实验硬件平台配置为：CPU i7-3930K, 6 核, 主频为 3.2 GHz, 系统内存为 32 GB, 显卡采用 GeForce GTX 780, 显卡内存为 3 GB, 计算能力为 3.5, 采用 AVT 公司的 GE4900 相机, 分辨率为 $4\ 872 \times 3\ 248$; 软件环境为 Windows7 + Visual Studio 2010 + CUDA Toolkit 5.0。

特征点中心并行快速定位算法和编码点并行快速识别算法将对包含不同特征点数量的图像进行实验, 在特征点中心定位精度和编码点识别率不降低的前提下, 分别测试串行和并行方案的耗时情况, 其中所有的并行耗时都包括算法所需数据的传输时间, 时间统计方法采用 5 次计时取平均值、加速效果采用 CPU 耗时/CUDA 耗时的方法。

图像中包含不同特征点数量时, 采用本文提出的特征点中心并行快速定位算法和串行定位算法定位特征点中心的耗时情况如表 2 所示。

表 2 定位特征点中心的并行和串行算法的耗时情况

Tab. 2 Comparison of time-consuming for target center located by serial and parallel methods

特征点数量	串行耗时/ms	并行耗时/ms	加速效果
270	190	70	2.7
3000	275	80	3.4
7000	410	91	4.5
10000	495	98	5.1
20000	796	137	5.8

本文提出的特征点中心并行快速定位算法充分挖掘了点中心定位算法过程中的并行性, 将具

有并行性的部分采用数据并行的处理方式实现。通过采用大量线程同时并行处理, 并且处理过程中有效地实现线程间的通信、同步和原子操作等方法, 有效提升了算法的定位速度。从表 2 可以看到: 相比于串行的特征点定位算法, 特征点中心并行快速定位算法有显著的加速效果, 在提取 10 000 个点中心时, 加速比达到 5.1 倍, 并且随着点数的增多加速比还会显著提升。

图像中包含不同特征点数量时, 采用本文提出的编码点并行快速识别算法和串行编码点识别算法的耗时情况如表 3 所示。

表 3 编码点识别算法并行和串行方案的耗时对比

Tab. 3 Comparison of time-consuming for serial and parallel schemes in code detection algorithm

特征点数量	串行耗时/ms	并行耗时/ms	加速效果
270	9.2	3.6	2.6
3000	377.0	50.1	7.5
7000	1869.2	172.6	10.8
10000	3754.6	308.2	12.2
20000	14899.9	1119.5	13.3

编码点并行快速识别算法是在研究串行算法并行性的基础上, 将其中最耗时的搜索有效编码点集部分采用数据并行方式处理, 以实现大量线程的同时搜索, 有效提升了搜索速度。从表 3 中可以看到: 编码点并行快速识别算法由于快速搜索部分具有很好的并行性, 采用并行计算获得了很好的加速效果, 在 10 000 个特征点中搜索、识别编码点, 加速比达到 12.2 倍, 随着点数的增加, 加速效果还会显著提升。

图像中包含不同特征点数量时, 采用本文提出的并行方案和全部算法串行的串行方案的耗时情况如表 4 所示。

表 4 采用不同方案实现的动态视觉测量系统耗时情况

Tab. 4 Time-consuming comparison of dynamic vision measurement system realized by different schemes

特征点数量	串行耗时/ms	并行耗时/ms	加速效果
270	299.2	173.6	1.7
3000	752.2	230.1	3.3
7000	2384.2	368.6	6.5
10000	4360.5	516.2	8.4
20000	15180.8	1352.5	11.7

从表 4 可以看出,并行方案比串行方案的处理速度有明显的提升,在测量场中包含 10 000 个特征点时,处理速度提升了 8.4 倍,并且随着点数的增多加速比还会显著提升,有效解决了大尺寸动态视觉测量系统实时性差的问题。

5 结 论

本文针对大尺寸动态视觉测量系统处理速度较慢,实时性差的问题,在不损失特征点定位精度

和编码点识别率的前提下分别提出了特征点中心并行快速定位算法和编码点并行快速识别算法。该算法充分发挥了 CUDA 通用并行计算架构的特点,有效地提高了测量系统的处理速度。最终的实验结果表明,当大尺寸动态视觉测量系统中的图像包含 270 个特征点时,并行计算方案比串行方案时间开销减少了 42%,特征点数量达到 20 000 个时,时间开销减少 91% 以上,并随着点数的增多时间开销减少的比重还会有明显提升。

参考文献:

- [1] 陈竹梅,平丽浩,徐东海,等. 机载大尺寸天线平面度控制与测试研究[J]. 现代雷达,2010,32(4):82-87.
CHEN ZH M, PING L H, XU D H, *et al.*. A study on control measures and measurement technology of flatness of airborne large size antenna [J]. *Modern Radar*, 2010,32(4):82-87. (in Chinese)
- [2] 李潜,刘子强. 大型飞机研制需要的风动试验技术[C]. 中国航空学会 2007 年学术年会气动专题,1-10.
LI Q, LIU Z Q. Pneumatic test technology for large aircraft development needs [C]. *Chinese Aviation Association 2007 Annual Meeting of the Academic Pneumatic Special*,1-10. (in Chinese)
- [3] 李磊刚,梁晋,唐正宗,等. 飞机结构件运动数据的动态视觉测量系统[J]. 光学精密工程,2012,20(9):1929-1938.
LI L G, LIANG J, TANG ZH Z, *et al.*. Optical and dynamic measuring system for movement data of aircraft structural parts [J]. *Opt. Precision Eng.*, 2012,20(9):1929-1938. (in Chinese)
- [4] THOMPSON E A, ANDERSON T R. A CUDA implementation of the continuous space language model [J]. *Journal of Supercomputing*, 2014, 68(1):65-86.
- [5] HAVEL J, DUBSKA M, HEROUT A. Real-time detection of lines using parallel coordinates and CUDA [J]. *Journal of Real-Time Image Processing*, 2014,9(1SD):205-216.
- [6] GOLDSWORTHY M J. A GPU-CUDA based direct simulation Monte Carlo algorithm for real gas flows [J]. *Computers & Fluids*, 2014,94:58-68.
- [7] 李大禹,胡立发,穆全全,等. CUDA 架构下的液晶自适应波面数值解析[J]. 光学精密工程,2010,18(4):848-854.
LI D Y, HU L F, MU Q Q, *et al.*. Wavefront calculation of liquid crystal adaptive optics based on CUDA [J]. *Opt. Precision Eng.*, 2010,18(4):848-854. (in Chinese)
- [8] 王晶,李仕. 运动模糊视频图像在图形处理器平台上的实时恢复[J]. 光学精密工程,2010,18(10):2262-2268.
WANG J, LI SH. Real-time restoration of motion-blurred video images on GPU [J]. *Opt. Precision Eng.*, 2010,18(10):2262-2268. (in Chinese)
- [9] 连远锋,赵剡,何晖光. 自适应耦合变分模型及其 GPU 实现[J]. 仪器仪表学报,2013,34(11):2520-2525.
LIAN Y F, ZHAO Y, HE H G. Adaptive coupled variation model and its GPU implementation [J]. *Chinese Journal of Scientific Instrument*, 2013,34(11):2520-2525. (in Chinese)
- [10] 李仕,张葆,孙辉. 航空斜视成像异速像移的实时恢复[J]. 光学精密工程,2009,17(4):895-900.
LI SH, ZHANG B, SUN H. Real-time restoration for aerial side-oblique images with different motion rates [J]. *Opt. Precision Eng.*, 2009,17(4):895-900. (in Chinese)
- [11] 许晓臣,董明利,王君,等. 基于 CUDA 的动态视觉测量像面特征点中心快速定位算法[J]. 计算机工程与科学,2014,36(12):2378-2385.
XU X CH, DONG M L, WANG J, *et al.*. A fast target center location algorithm for dynamic vision measurement based on CUDA [J]. *Computer Engi-*

- neering & Science, 2014, 36(12):2378-2385. (in Chinese)
- [12] ZHUANG W, DONG M L, SUN P, *et al.*. A variable weighted centroid method for retro reflective target locating[C]. *Proc of the sixth International Symposium on Precision Mechanical Measurements*, 2013.
- [13] SHORTIS M R, CLARKE T A, SHORT T. A comparison of some techniques for the subpixel location of distance target images[C]. *3rd Conference on Videometrics*, Boston, MA, 1994.
- [14] DONG M L, XU L, WANG J, *et al.*. Variable-weighted grayscale centroiding and accuracy evaluating [J]. *Advances in Mechanical Engineering*, 2013, 428608.
- [15] 王灿进, 孙涛, 陈娟. 局部不变特征匹配的并行加速度技术研究[J]. *液晶与显示*, 2014, 29(2):266-274.
- WANG C J, SUN T, CHEN J. Speeding up local invariant feature matching using parallel technology[J]. *Chinese Journal of Liquid Crystals and Displays*, 2014, 29(2):266-274. (in Chinese)
- [16] 王林波, 王延杰, 邸男, 等. 基于几何特征的圆形标志点亚像素中心定位[J]. *液晶与显示*, 2014, 6(6):1003-1009.
- WANG L B, WANG Y J, DI N, *et al.*. Subpixel location of circle target center based on geometric features[J]. *Chinese Journal of Liquid Crystals and Displays*, 2014, 6(6):1003-1009. (in Chinese)
- [17] 王洪涛, 李丹. 基于改进粒子群算法的图像边缘检测研究[J]. *液晶与显示*, 2014, 29(5):800-804.
- WANG H T, LI D. Image edge detection research based on improved particle swarm optimization algorithm[J]. *Chinese Journal of Liquid Crystals and Displays*, 2014, 29(5):800-804. (in Chinese)

作者简介:



董明利(1965—),女,辽宁鞍山人,博士,教授,硕士生导师,1989年于合肥工业大学获得硕士学位,2009年于北京理工大学获得博士学位,主要从事视觉测量技术、精密测量技术与仪器的研究。E-mail:dongml@sina.com



许晓臣(1987—),男,河北肃宁人,硕士研究生,2012年于石家庄铁道大学获得学士学位,主要从事视觉测量技术及并行计算方面的研究。E-mail:xxcamn@163.com

(版权所有 未经许可 不得转载)