

基于 CUDA 的高分辨率数字视频 图像配准快速实现^{*}

闫钧华, 杭谊青, 许俊峰, 储林臻

(南京航空航天大学航天学院 南京 210016)

摘要: 高分辨率数字视频图像数据量巨大, 基于 SIFT 图像配准算法在 CPU 上实现时用时巨大。针对此, 首先对配准算法中3个最耗时的部分: SIFT 特征提取; SIFT 特征匹配; RANSAC 算法提纯匹配点对, 求解变换模型参数。对此展开重点研究, 研究其并行算法。然后基于 CUDA 并行快速实现高分辨率数字视频图像配准。实验结果表明: 基于 SIFT 图像配准算法在 CPU 与 CUDA 上实现, 在配准效果相近时, 在 CUDA 上实现的处理速度比在 CPU 上实现的处理速度提高了 100 多倍, 并且随着图像像素数的增加加速比有显著提高。

关键词: 图像配准; 高分辨率; 数字视频; CUDA

中图分类号: TP391.4 **文献标识码:** A **国家标准学科分类代码:** 510.40

Quick realization of CUDA-based registration of high-resolution digital video images

Yan Junhua, Hang Yiqing, Xu Junfeng, Chu Linzhen

(College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: High-resolution digital video image contains a huge amount of data, and the (scale invariant feature transform) SIFT based image registration algorithm achieved on CPU costs enormous amount of time. Aiming at this problem, firstly we emphatically studied the parallel algorithms of the three most time-consuming parts in the image registration algorithm: SIFT feature extraction; SIFT feature matching; and random sample consensus (RANSAC) algorithm for purifying matching point pair and solving transformation model parameters. Then, the high-resolution digital video image registration was achieved quickly and in parallel based on compute unified device architecture (CUDA). The experiment results show that for the SIFT-based image registration algorithm implemented on CPU and CUDA, the processing speed for CUDA is over 100 times faster than that for CPU when the registration results are similar, and the acceleration ratio improves significantly while the number of the image pixels increases.

Keywords: image registration; high-resolution; digital video; compute unified device architecture (CUDA)

1 引言

高分辨率数字视频图像数据量巨大, 图像处理算法在 CPU 上实现时用时巨大。研究图像处理并行算法并且利用 CUDA 并行计算优势^[1] 编程实现, 可以减少图像处理算法用时, 快速实现高分辨率数字视频图像配准。

SIFT 特征^[2] 对于旋转、缩放、平移具有不变性, 并且对噪声、光照变化等有很好的稳定性^[3], 已成功应用于图像配准。但基于 SIFT 的图像配准算法本身复杂度较高, 且高分辨率数字视频图像数据量大, 导致在 CPU 上计算用时长, 算法效率较低^[4]。国内外学者对此进行了很多研究, 如 Sinha S 等用 OpenGL 实现了 SIFT 算法, 处理 $1\,024 \times 768$ 的图像加速比达 20 倍^[5]。

收稿日期: 2013-09 Received Date: 2013-09

^{*} 基金项目: 航空科学基金(20135152049)、南京航空航天大学基本科研业务费专项科研项目(NN2012083, NS2010214, NP2011048)资助

Changchang Wu 等人使用 CUDA 实现了 SIFT 算法,田文等人使用 CUDA 实现了 SIFT 算法及特征匹配,性能达到 CPU 实现的 30 ~ 50 倍^[6]。但是仍不能满足对高分辨率数字视频图像配准快速性的要求,因此本文主要研究基于 SIFT 的图像配准并行算法,利用 CUDA 架构的并行计算优势^[7],通过优化带宽、数据存取方式等策略^[8]对算法进一步优化、改进,快速实现高分辨率数字视频图像配准。

2 基于 SIFT 的图像配准并行算法研究

基于 SIFT 的高分辨率数字视频图像配准算法复杂度高,处理数据量大。若采用 CPU 串行运算,耗时长、算法效率低。配准算法中有 3 个最耗时的部分: SIFT 特征提取; SIFT 特征匹配; RANSAC 算法^[9]提纯匹配点对,求解变换模型参数。本文对这 3 个部分展开重点研究,研究其并行算法,基于 CUDA 并行快速实现高分辨率数字视频图像配准算法。

2.1 SIFT 特征提取的并行算法研究

SIFT 特征是一种图像局部特征,具有旋转、缩放、平移不变性^[10]。SIFT 特征提取流程如图 1 所示。

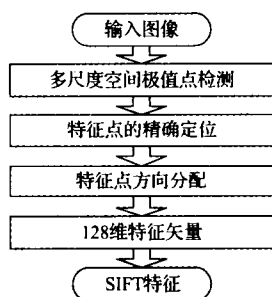


图 1 SIFT 特征提取流程图

Fig. 1 Flow chart of SIFT feature extraction

实验表明 SIFT 特征提取中尺度空间的构造、特征矢量的生成最为耗时,约占总时间的 70% 左右^[11]。本文重点研究了这 2 部分的可并行性,基于 CUDA 对 SIFT 特征提取进行并行化设计如下:

1) 并行构建高斯图像金字塔,独立并行的将图像与变尺度高斯函数卷积以分别建立每一子层。

2) 因为特征点的计算过程只与各自邻域有关,所以独立并行地通过 DoG (difference of gaussian) 相邻层之间比较、计算,完成尺度空间极值点检测及定位。

3) 因为子层与子层之间无任何数据通信,所以并行计算每个特征点邻域梯度和方向,累加构建梯度方向直方图,确定特征点主方向。

4) 利用特征点邻域信息,独立并行计算梯度方向直方图,生成每个特征点的 128 维特征矢量。

2.2 SIFT 特征匹配的并行算法研究

本文基于 CUDA 并行运算的特点,采用相关系数作为相似性度量,运用 CUBLAS 库中快速矩阵乘法实现特征矢量间的匹配。假定 2 幅图像进行 SIFT 特征提取后,基准图像得到 m 个特征矢量,生成 $m \times 128$ 的矩阵 M ; 待配准图像得到 n 个特征矢量,生成 $n \times 128$ 的矩阵 N ^[12]。将 2 个矩阵进行整理,使用并行矩阵乘法得到相关系数矩阵,其矩阵中的每个值即为对应特征矢量间的相关系数值。

为提高匹配的正确性及匹配精度,在待配准图像中同时查找与基准图像某一特征点相关系数最大及次大的特征点,当最大相关系数与次大相关系数的比值满足一定阈值时,接受这组匹配点对:

$$\text{次大相关系数} \leq \text{最大相关系数} \times THR$$

式中: THR 为阈值,范围为 (0, 1)。

基于 CUDA 对 SIFT 特征匹配进行并行化设计如下:

1) 独立并行计算基准图像中某一特征点与待配准图像中每个特征点之间的相关系数。

2) 独立并行计算 2 个 128 维矢量的对应乘积。

3) 运用 CUBLAS 库中快速并行矩阵乘法及转置得到相关系数矩阵。

4) 因为查询相关系数矩阵每一行最大及次大相关系数时,需进行数据通信及循环比较,所以基于 CUDA 利用 block 中 shared memory 实现数据通信,并且通过改造并行归约算法实现并行比较,得到最大及次大相关系数。

2.3 RANSAC 算法的并行性研究

随机抽样一致性法 RANSAC 可以求解最多样本满足的数学模型的参数^[13],分 2 步实现:

1) 假设。由 SIFT 特征匹配得到的 N 对粗匹配点作为数据集 P , $p_i (i = 1, 2, \dots, N)$ 为基准图像中的特征点, $q_i (i = 1, 2, \dots, N)$ 为相应的待配准图像中的特征点。采用仿射变换模型^[14],基准图像的点 (x, y) 经过仿射变换到待配准图像的点 (x_1, y_1) :

$$\begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} T \quad (1)$$

式中: T 为变换矩阵,形式为:

$$T = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix}, a_{ij} \text{ 为仿射变换的 6 个参数,每次}$$

从数据集中采样模型需要的最少数量样本,计算模型的参数。

2) 检验。用数据集 P 中其他点对验证解算出的模型,统计符合该模型参数的样本数目,确定最多样本满足的数学模型的参数。基于 CUDA 对 RANSAC 算法进行并行化设计如下:

1) 传统 CPU 串行算法在确定采样次数 M 后,对数据集进行循环串行计算,时间消耗大。因为算法每次进行

采样都是随机的、独立的,所以基于 CUDA 并行进行随机采样计算。

2) 因为根据采样数据进行模型解算时,计算过程只与本次采样数据有关,每次解算也都是独立的,所以基于 CUDA 并行进行模型解算。

3) 检验每次求解的模型参数时需将数据集中所有样本代入计算,传统 CPU 串行算法需循环实现,在样本数较大的情况下耗时长。因为每个样本的检验只与该样本有关,无需数据通信,也是互相独立的,所以基于 CUDA 并行进行模型参数检验。

3 基于 CUDA 的图像配准快速实现

3.1 SIFT 特征提取并行优化及 CUDA 实现

SIFT 特征提取中最耗时的部分是:构建高斯图像金字塔及 DoG、特征点检测及定位、特征点方向分配、特征矢量的生成。基于 CUDA 快速实现 SIFT 特征提取流程图如图 2 所示。

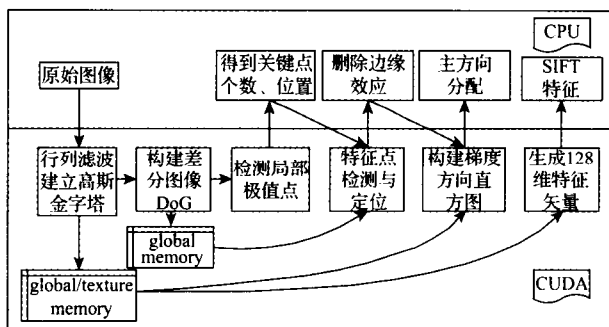


图2 基于 CUDA 快速实现 SIFT 特征提取流程图

Fig. 2 Flow chart of quick-achieved SIFT feature extraction based on CUDA

1) 构建高斯图像金字塔及 DoG

通过变尺度高斯函数与图像进行卷积生成高斯金字塔,高斯核数据是相对不变量,并且需要频繁访问。因此,将高斯卷积核存放于 GPU 中带 Cache 的 constant memory 中以便快速读取。

高斯函数具有各向同性,可将高斯卷积分解为分别沿着 x 轴和 y 轴进行的 2 个一维行、列高斯卷积。若二维高斯模板大小为 $N \times N$,需要计算、存储每一个值。对大小为 $W \times H$ 的图像进行二维高斯卷积需进行 $W \times H \times N \times N$ 次乘法操作。若使用 $1 \times N$ 的一维行列卷积,则只需计算、存储 $1 \times N$ 个值,且只需执行 $2 \times W \times H \times N$ 次乘法操作。减少内存读取次数,提高计算效率。

通过每组相邻 2 层的高斯图像金字塔相减得到 DoG。每个 block 分配 16×16 个线程,每个 grid 有 $((\text{图像宽度} + \text{dimBlock.x} - 1) / \text{dimBlock.x}) \times ((\text{图像高度} +$

$\text{dimBlock.y} - 1) / \text{dimBlock.y})$ 个 block,将图像进行棋盘划分^[15]。每个线程对应一个像元,同时并行相减得到 DoG 并存入 global memory。

2) 特征点检测及定位

读取 global memory 中相邻 3 层 DoG,根据执行参数索引检测点和它同尺度的 8 个相邻点及上下相邻尺度对应的 9×2 个点进行比较,生成记录极值点位置的压缩数据并传回主机内存,减少 CPU 端与 GPU 端之间传输的数据量,提高效率。在 CPU 端得到极值点位置信息。

利用三维子像元插值的方法进行特征点定位。为有效利用执行单元,每个 block 分配 32 个线程,grid 的个数为: $(\text{极值点个数} + \text{dimBlock.x} - 1) / \text{dimBlock.x}$ 。每个线程计算得到一个特征点的精确定位及对比度、主曲率比值信息,存入 global memory。

由于 GPU 的硬件限制,CUDA 架构对于分支、判断等逻辑语句的处理效率较低,因此将特征点信息传回主机内存,在 CPU 端剔除低对比度的点,删除边缘效应,得到精化后的特征点信息,再传回 global memory。

3) 特征点方向分配

计算特征点邻域内梯度的幅角及幅值,使用直方图进行统计,梯度方向直方图的峰值对应角度即为该特征点的主方向。考虑到 GPU 的硬件架构及执行效率,将 $0^\circ \sim 360^\circ$ 的范围分为 32 个柱进行梯度方向直方图统计,每 11.25° 为一个柱。每个 block 分配 32 个线程,grid 的个数即为特征点的个数。因邻域像素值、梯度方向直方图及高斯卷积核数据需要频繁访问,因此每个 block 分配相应大小 shared memory 存放这些数据,减少对 global memory 重复读取次数,提高数据访问效率。在计算过程中同步生成高斯核,减少内存读取并增加算法的计算密集度。

每个 block 计算一个特征点邻域内的梯度幅角与幅值,采用并行归约方法累加生成该特征点的梯度方向直方图,存入 shared memory。每个 block 的 0 线程根据总梯度方向直方图记录该特征点的主方向,并判断是否存在超过主峰值 80% 能量的峰值,若存在,即标记对应角度为辅方向,将方向信息存入 global memory。

将各特征点方向信息传回主机内存,在 CPU 端处理有多个方向的特征点,生成新的特征点列表并传回 GPU 端 global memory。

4) 特征矢量的生成

考虑到纹理内存在 GPU 内部针对二维数据访问具有优化的纹理缓存^[16],因此在特征矢量生成时将特征点对应的高斯图像绑定纹理,实现大量数据访问的加速效果。特征矢量生成的 kernel 函数配置的 grid 个数即为特征点个数,每个 block 分配 16 个线程,生成该特征点的特征矢量。

根据特征点主方向将特征点邻域内图像进行旋转,再以特征点为中心,划分 4×4 子区域,每个子区域内计算8个方向的梯度方向直方图,在每个梯度方向进行累加,生成 $4 \times 4 \times 8 = 128$ 维SIFT特征矢量,存入global memory。

3.2 SIFT特征匹配并行优化及CUDA实现

基于CUDA快速实现SIFT特征匹配分为3部分:kernel 0生成相关系数矩阵;kernel 1查找最大及次大相关系数;在CPU中根据CUDA计算得到的结果生成2幅图像的粗匹配点对,并将结果传回GPU内存。主要实现步骤:

1) 将待配准图像得到的 $n \times 128$ 特征向量矩阵 N 整理为 $128 \times n$ 矩阵,以便实现与基准图像得到的 $m \times 128$ 特征矢量矩阵 M 间的矩阵乘法。

2) 启动CUBLAS库,分配显存,设置参数,调用cublasSgemm函数实现2个特征矢量矩阵乘法,将结果矩阵进行转置得到相关系数矩阵。矩阵中的每一个元素代表相对应特征点之间的相关系数值。

3) 配置执行参数,调用查找最大及次大相关系数的kernel。分配的block个数为:基准图像特征点数/16 + 1,每个block分配 16×16 个thread。根据CUDA线程索引方法,在每一行中进行并行比较。

4) 每个block中分配相应大小的shared memory,记录最大、次大相关系数值及对应点的序号。每个block得到基准图像中16个特征点的最大及次大相关系数信息。

5) 在CPU内存中得到2幅图像粗匹配点对,并将结果传回GPU内存。

3.3 RANSAC算法并行优化及CUDA实现

基于CUDA快速实现RANSAC算法:在GPU中并行计算模型参数及模型符合程度,分为随机数产生kernel函数和模型解算及检验kernel函数;在CPU内存中选取符合程度最好的变换模型参数。基于CUDA快速实现RANSAC算法的具体流程如图3所示。

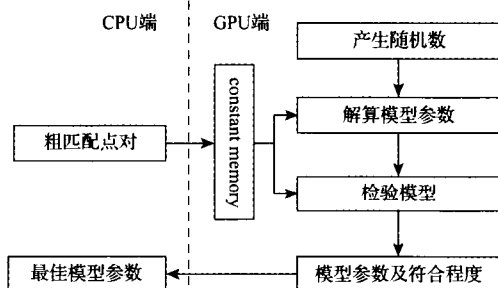


图3 基于CUDA快速实现RANSAC算法流程图

Fig. 3 Flow chart of quick-achieved RANSAC algorithm based on CUDA

主要实现步骤:

1) 确定循环次数 M ,将 M 作为kernel执行参数中block的个数。可将 M 设置成16的整数倍,提高访问global memory和shared memory的效率。 N 为每个block中thread值,可将其设置成32的倍数。

2) 配置执行随机数产生kernel函数。利用CURAND库中的curand函数在每个block中产生3个不同随机数,作为采样的样本序号。

3) 因为各个样本数据在解算过程中为不变量,且每个block中的thread都需访问,所以将其由CPU内存载入GPU的constant memory中,利用constant memory的缓存机制来提高数据访问速度。

4) 配置执行模型解算及检验kernel函数。每个block中某一线程读取由3个随机数确定的样本序号,从constant memory中读取对应粗匹配点对,解算模型参数,存入shared memory。

5) 每一个block进行一组模型参数的检验。根据配置参数进行线程索引,每一个thread读取对应constant memory中不同样本,计算其对该block中模型的符合程度。使用原子加atomicAdd函数累计shared memory中符合该模型的样本数。

6) 将每个block计算得到的符合该模型的样本数从GPU显存传回CPU内存,计算得到最佳模型参数及精匹配点对。

4 实验结果及分析

本文基于CUDA快速实现基于SIFT的高分辨率数字视频图像配准算法,通过实验验证在配准效果相近时,其比使用CPU实现同样的配准算法速度提高100多倍。所采用的实验环境:CPU采用Intel Core 2 Quad Q8200,4核,主频2.33 GHz。GPU采用支持CUDA的NVIDIA GeForce GTX 480,计算能力2.0,GPU运行频率1.4 G,有15个SM,120个SP,每个block最多提供1024个线程。配准算法中有3个最耗时的部分:SIFT特征提取;SIFT特征匹配;RANSAC算法提纯匹配点对,求解变换模型参数。对这3部分分别进行实验,在效果相近时,分别测CUDA和CPU实现的用时。在配准效果相近时,分别测CUDA和CPU实现同样的配准算法的总用时。

选取3组不同像素数的图像进行配准实验:(a)1280×960像素;(b)1920×1080像素;(c)2880×1215像素。CPU、CUDA实现配准算法效果如图4、5所示。

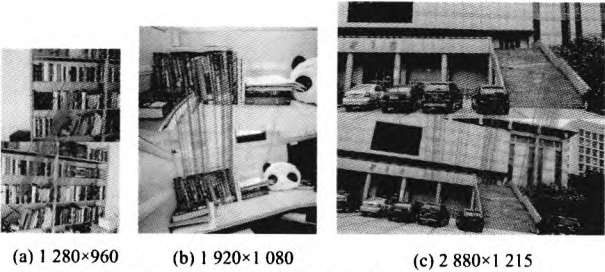


图 4 基于 SIFT 的图像配准算法 CPU 实现效果

Fig. 4 SIFT image registration results based on CPU

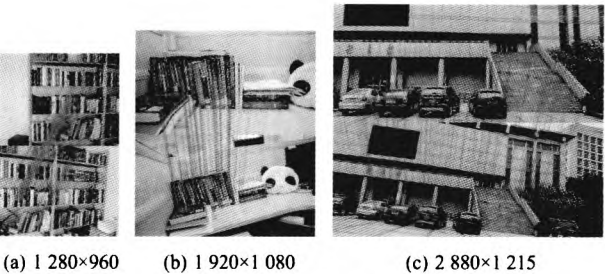


图 5 基于 SIFT 的图像配准算法 CUDA 实现效果

Fig. 5 SIFT image registration results based on CUDA

本文采用均方根误差 (root mean-square error, RMSE) 参量作为图像配准效果的评价指标。分别计算 3 组图像在 CPU 与 CUDA 下实现配准后基准图像与配准图像的 RMSE, 数值比较如表 1 所示。

表 1 基于 SIFT 的图像配准算法在 CPU 与 CUDA 上实现的配准效果比较

Table 1 The comparison of Registration effects of the SIFT image registration algorithm based on CPU and CUDA

图像序号	CPU 实现配准的 RMSE	CUDA 实现配准的 RMSE
(a)	0.073 6	0.032 9
(b)	0.036 9	0.021 2
(c)	0.004 2	0.005 1

实验结果表明:3 组图像基于 SIFT 的图像配准算法经 CPU 与 CUDA 实现后的 RMSE 值都很小, 配准后图像与基准图像的相似度高, 配准效果好。

SIFT 特征提取在 CPU 与 CUDA 上实现的性能比较如表 2 所示。

表 2 SIFT 特征提取在 CPU 与 CUDA 上实现的性能比较

Table 2 The performance comparison of SIFT feature extraction based on CPU and CUDA

图像序号	SIFT 特征提取 CPU 实现		SIFT 特征提取 CUDA 实现		加速比
	基准图像特征点/待配准图像特征点/个	CPU 用时/ms	基准图像特征点/待配准图像特征点/个	CUDA 用时/ms	
(a)	803/124 5	14 047. 67	816/124 9	90. 80	154. 71
(b)	2 067/160 9	23 160. 62	2 070/155 5	140. 36	165. 01
(c)	2 008/220 3	36 700. 47	2 007/219 6	204. 33	179. 61

实验结果表明:SIFT 特征提取在 CPU 与 CUDA 上实现时提取到的特征点数相近, 如在 (c) 组中, CPU 实现提取到 2 008/2 203 个特征点, CUDA 实现提取到 2 007/2 196 个特征点, 这说明 SIFT 特征提取在 CPU 与 CUDA 上实现的效果相近。SIFT 特征提取在 CUDA 上实现的速

度比在 CPU 上实现的速度提高了 100 多倍, 如在 (c) 组中, 加速比达到 179. 61。随着图像像素及特征点数的增加加速比还会有所提高。

SIFT 特征匹配在 CPU 与 CUDA 上实现的性能比较如表 3 所示。

表 3 SIFT 特征匹配在 CPU 与 CUDA 上实现的性能比较

Table 3 The performance comparison of SIFT feature matching based on CPU and CUDA

图像序号	SIFT 特征匹配算法 CPU 实现		SIFT 特征匹配算法 CUDA 实现		加速比
	粗匹配点/对	CPU 用时/ms	粗匹配点/对	CUDA 用时/ms	
(a)	312	555. 25	317	2. 12	216. 91
(b)	377	1 112. 36	377	3. 60	308. 99
(c)	543	1 395. 02	540	4. 21	331. 36

实验结果表明:SIFT 特征匹配在 CPU 与 CUDA 上实现时得到的粗匹配点对数相近, 如在 (c) 组中, CPU 实现与 CUDA 实现分别得到了 543、540 对粗匹配点, 这说明 SIFT 特征匹配在 CPU 与 CUDA 上实现的效果相近。SIFT 特征匹配在 CUDA 上实现的速度比在 CPU

上实现的速度提高了 300 多倍, 如在 (c) 组中, 加速比达到 331. 36。随着图像像素及匹配点对数的增加加速比还会有所提高。

RANSAC 算法在 CPU 与 CUDA 上实现的性能比较如表 4 所示。

表 4 RANSAC 算法在 CPU 与 CUDA 上实现的性能比较

Table 4 The performance comparison of RANSAC algorithm based on CPU and CUDA

图像序号	RANSAC 算法 CPU 实现		RANSAC 算法 CUDA 实现		加速比
	精匹配点(对)	CPU 用时/ms	精匹配点/对	CUDA 用时/ms	
(a)	223	40.06	221	4.73	8.47
(b)	249	103.70	221	4.96	20.91
(c)	265	156.34	260	5.20	30.07

实验结果表明:RANSAC 算法在 CPU 与 CUDA 上实现时得到的精匹配点对数相近,如在(c)组中,CPU 实现与 CUDA 实现分别得到了 265、260 对精匹配点,这说明 RANSAC 算法在 CPU 与 CUDA 上实现的效果相近。RANSAC 算法在 CUDA 上实现的速度比在 CPU 上实现的速度提高了 30 多倍,如在(c)组中,加速比达到 30.07。随着图像像素及匹配点对数的增加加速比还会有所提高。

基于 SIFT 的图像配准算法在 CPU 与 CUDA 上实现的总用时和加速比如表 5 和图 6 所示。

表 5 基于 SIFT 的图像配准算法在 CPU 与 CUDA 上实现的总用时和加速比

Table 5 Total time consumption and acceleration ratio of SIFT image registration algorithm based on CPU and CUDA

图像序号	图像配准算法 CPU	图像配准算法 CUDA	加速比
	实现总用时/ms	实现总用时/ms	
(a)	15 088.20	98.45	153.26
(b)	25 418.80	147.07	172.83
(c)	39 418.10	214.35	183.89

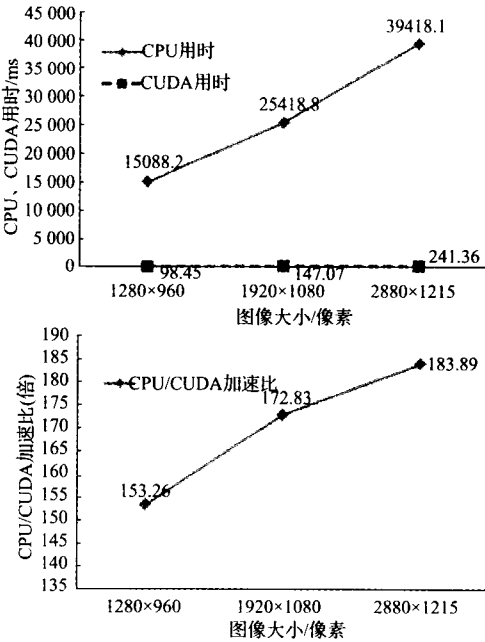


图 6 基于 SIFT 的图像配准算法在 CPU 与 CUDA 上实现的总用时和加速比

Fig. 6 Total time consumption and acceleration ratio of SIFT image registration algorithm based on CPU and CUDA

从图 6 可直观地看到基于 SIFT 的图像配准算法在 CPU 上实现时用时巨大且随着图像像素数的增加增长较快,而在 CUDA 上实现时用时较少且随着图像像素数的增加增长较少。实验结果表明:基于 SIFT 图像配准算法在 CPU 与 CUDA 上实现,在配准效果相近时,在 CUDA 上实现的处理速度比在 CPU 上实现的处理速度提高了 100 多倍,并且随着图像像素数的增加加速比还会有所提高。由于配准算法中 SIFT 特征提取最为耗时,占总耗时的 90% 以上,所以配准算法总加速比接近 SIFT 特征提取的加速比。

5 结 论

本文利用 CUDA 并行计算优势,快速实现基于 SIFT 的高分辨率数字视频图像配准,比较了配准算法在 CPU 与 CUDA 上实现时的运行效率。实验结果表明:在配准效果相近时,其 CPU/CUDA 加速比达到 100 多倍,并随着图像像素数增大加速比有显著提高。

参考文献

[1] MEMBARTH R, HANNIG F, TEICH J, et al. Frameworks for GPU accelerators: A comprehensive evaluation using 2D/3D image registration more authors[C]. Symposium on Application Specific Processors, SASP, San Diego, CA, 2011: 78-81.

[2] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[3] 蒋建国, 蔡志祥, 齐美彬, 等. 旋转摄像机下的快速目标检测算法[J]. 电子测量与仪器学报, 2012, 26(7): 624-628.

JIANG J G, CAI ZH X, QI M W, et al. The method of rapid moving object detection in rotating camera [J]. Journal of Electronic Measurement and Instrument, 2012, 26(7): 624-628.

[4] 戴金波, 赵宏伟, 刘君玲, 等. 一种针对于描述子的 SIFT 简化方法[J]. 仪器仪表学报, 2012, 32(10): 2255-2262.

DAI J B, ZHAO H W, LIU J L, et al. Simplified SIFT

- algorithm based on descriptor[J]. Chinese Journal of Scientific Instrument, 2012, 32(10): 225-2262.
- [5] SINHA S, FRAHM J M, POLLEFEYS M, et al. Feature tracking and matching in video using programmable graphics hardware[J]. Machine Vision and Applications, 2011, 22(1): 207-217.
- [6] 田文, 徐帆, 王宏远, 等. 基于 CUDA 的尺度不变特征变换快速算法[J]. 计算机工程, 2010, 36(8): 219-221.
- TIAN W, XU F, WANG H Y, et al. Fast scale invariant feature transform algorithm based on CUDA[J]. Computer Engineering, 2010, 36(8): 219-221.
- [7] ADAM H, RADOVAN J, ROMAN J, et al. Real-time object detection on CUDA[J]. Journal of Real-Time Image Processin, 2011, 6(3): 159-170.
- [8] 肖汉, 张祖勋. 基于 GPGPU 的并行影像匹配算法[J]. 测绘学报, 2010, 39(1): 46-51.
- XIAO H, ZHANG Z X. Parallel image matching algorithm based on GPGPU[J]. Acta Geodaetica et Cartographica Sinica, 2010, 39(1): 46-51.
- [9] 周骏, 陈雷霆, 刘启和, 等. 基于序贯概率及局部优化随机抽样一致性算法[J]. 仪器仪表学报, 2012, 33(9): 2037-2044.
- ZHOU J, CHEN L T, LIU Q H, et al. Fast and accurate RANSAC based on optimal sequential probability test and local optimization[J]. Chinese Journal of Scientific Instrument, 2012, 33(9): 2037-2044.
- [10] MIKOLAJCZYK K, SCHMID C. Scale & affine invariant interest point detectors[J]. International Journal on Computer Vision, 2004, 60(1): 63-86.
- [11] HEYMANN S, MULLER K, SMOLIC A. SIFT implementation and optimization for general-purpose GPU[C]. Proceedings of the 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2007: 317-322.
- [12] HUANG Y, LIU J S, TU M S, et al. Research on CUDA-based SIFT registration of SAR image[C]. The Fourth International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), Tianjin, China, 2011: 100-104.
- [13] FISCHLER M A, BOLLES R C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6): 381-395.
- [14] 李志华, 陈耀武. 基于多摄像头的目标连续跟踪[J]. 电子测量与仪器学报, 2009, 23(2): 46-51.
- LI ZH H, CHEN Y W. Continuous target tracking based on multiple cameras[J]. Journal of Electronic Measurement and Instrument, 2009, 23(2): 46-51.
- [15] NICKOLLS J, BUCK I, GARLAND M, et al. Scalable parallel programming with CUDA [J]. Queue, 2008, 6(2): 40-53.
- [16] 张舒, 褚艳利. GPU 高性能运算之 CUDA[M]. 北京: 中国水利水电出版社, 2009: 58-68.
- ZHANG SH, CHU Y L. GPU high performance computing of CUDA[M]. Beijing: China Water & Power Press, 2009: 58-68.

作者简介



闫钧华, 分别于 1993、2001 和 2004 年在南京航空航天大学获得学士、硕士和博士学位, 现为南京航空航天大学副教授, 主要研究方向为多源信息融合, 目标检测、跟踪与识别。

E-mail: yjh9758@126.com

Yan Junhua received her B. Sc., M. Sc. and Ph. D. degrees all from Nanjing University of Aeronautics and Astronautics in 1993, 2001 and 2004, respectively. Now, she is an associate professor in Nanjing University of Aeronautics and Astronautics. Her main research interests include multi-source information fusion, target detection, tracking and recognition.



杭谊青, 2012 年于南京航空航天大学获得学士学位, 现为南京航空航天大学硕士研究生, 主要研究方向为目标检测与定位。

E-mail: hyq58qitian@126.com

Hang Yiqing received B. Sc from Nanjing University of Aeronautics and Astronautics in 2012. Now, she is an M. Sc. candidate in Nanjing University of Aeronautics and Astronautics. Her main research direction is target detection and localization.