

GPU 加速与 L-ORB 特征提取的全景视频实时拼接

杜承焱¹ 袁景凌^{1,2} 陈旻骋¹ 李涛³

¹(武汉理工大学计算机科学与技术学院 武汉 430070)

²(交通物联网技术湖北省重点实验室(武汉理工大学) 武汉 430070)

³(佛罗里达大学电气与计算机工程系 美国 佛罗里达州 盖恩斯维尔 32611)

(duchengyao@whut.edu.cn)

Real-Time Panoramic Video Stitching Based on GPU Acceleration Using Local ORB Feature Extraction

Du Chengyao¹, Yuan Jingling^{1,2}, Chen Mincheng¹, and Li Tao³

¹(School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070)

²(Hubei Key Laboratory of Transportation Internet of Things (Wuhan University of Technology), Wuhan 430070)

³(Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA 32611)

Abstract Panoramic video is a sort of video recorded at the same point of view to record the full scene. The collecting devices of panoramic video are getting widespread attention with the development of VR and live-broadcasting video technology. Nevertheless, CPU and GPU are required to possess strong processing abilities to make panoramic video. The traditional panoramic products depend on large equipment or post processing, which results in high power consumption, low stability, unsatisfying performance in real time and negative advantages to the information security. This paper proposes a L-ORB feature detection algorithm. The algorithm optimizes the feature detection regions of the video images and simplifies the support of the ORB algorithm in scale and rotation invariance. Then the features points are matched by the multi-probe LSH algorithm and the progressive sample consensus (PROSAC) is used to eliminate the false matches. Finally, we get the mapping relation of image mosaic and use the multi-band fusion algorithm to eliminate the gap between the video. In addition, we use the Nvidia Jetson TX1 heterogeneous embedded system that integrates ARM A57 CPU and Maxwell GPU, leveraging its Teraflops floating point computing power and built-in video capture, storage, and wireless transmission modules to achieve multi-camera video information real-time panoramic splicing system, the effective use of GPU instructions block, thread, flow parallel strategy to speed up the image stitching algorithm. The experimental results show that the algorithm mentioned can improve the performance in the stages of feature extraction of images stitching and matching, the running speed of which is 11 times than that of the traditional ORB algorithm and 639 times than that of the traditional SIFT algorithm. The performance of the system accomplished in the article is 59 times than that of the former embedded one, while the power dissipation is reduced to 10 W.

收稿日期:2017-02-27;修回日期:2017-04-13

基金项目:国家自然科学基金项目(61303029)

This work was supported by the National Natural Science Foundation of China (61303029).

通信作者:袁景凌(yuanjingling@126.com)

Key words panoramic video; image stitching; heterogeneous computing; embedded GPU; oriented FAST and rotated BRIEF (ORB)

摘要 全景视频是在同一视点拍摄记录全方位场景的视频.随着虚拟现实(VR)技术和视频直播技术的发展,全景视频的采集设备受到广泛关注.然而制作全景视频要求 CPU 和 GPU 都具有很强的处理能力,传统的全景产品往往依赖于庞大的设备和后期处理,导致高功耗、低稳定性、没有实时性且不利于信息安全.为了解决这些问题,首先提出了 L-ORB 特征点提取算法,该算法优化了分割视频图像的特征检测区域以及简化 ORB 算法对尺度和旋转不变性的支持;然后利用局部敏感 Hash (Multi-Probe LSH)算法对特征点进行匹配,用改进的样本一致性(progressive sample consensus, PROSAC)算法消除误匹配,得到帧图像拼接映射关系,并采用多频带融合算法消除视频间的接缝.此外,使用整合了 ARM A57 CPU 和 Maxwell GPU 的 Nvidia Jetson TX1 异构嵌入式系统,利用其 Teraflops 的浮点计算能力和内建的视频采集、存储、无线传输模块,实现了多摄像头视频信息的实时全景拼接系统,有效地利用 GPU 指令的块、线程、流并行策略对图像拼接算法进行加速.实验结果表明,算法在图像拼接的特征提取、特征匹配等各个阶段均有很好的性能提升,其算法速度是传统 ORB 算法的 11 倍、传统 SIFT 算法的 639 倍;系统较传统的嵌入式系统性能提升了 29 倍,但其功耗低至 10 W.

关键词 全景视频;图像拼接;异构计算;嵌入式 GPU;ORB

中图法分类号 TP391.41

全景视频是在同一视点拍摄记录全方位场景的视频^[1-3].全景视频在视频监控、机器人视觉、数字城市、赛事直播以及新兴的虚拟现实都具有广泛的应用领域.利用全景摄像获取的全景视频,可以在虚拟现实中进行任意角度的沉浸式交互漫游、旋转以及缩放观察^[4];将全景设备应用于军事监控,将提高部队的战场感知能力,进而增强部队的单兵作战能力;应用于无人机遥感、机器人视觉,将克服以往单一摄像头视野有限的问题,增加探测及识别效率.

生成全景视频需要对视频中的每一帧进行图像

配准和图像融合,如图 1 所示.图像配准的关键为特征提取^[5].曹世翔等人^[6]提出一种快速有效的边缘特征点提取方法,实现多分辨率图像的融合拼接.该方法通过构建边缘图像的 Gauss 金字塔,从中提取稳定的特征点完成图像配准,并复用金字塔信息实现图像融合,极大地缩短整体拼接的时间. Ethan Rublee 于 2011 年提出 ORB(oriented FAST and rotated BRIEF)算法^[7],对 FAST 角点加入尺度不变特性,解决了 BRIEF(binary robust independent element feature)的旋转不变性和噪声敏感问题,效

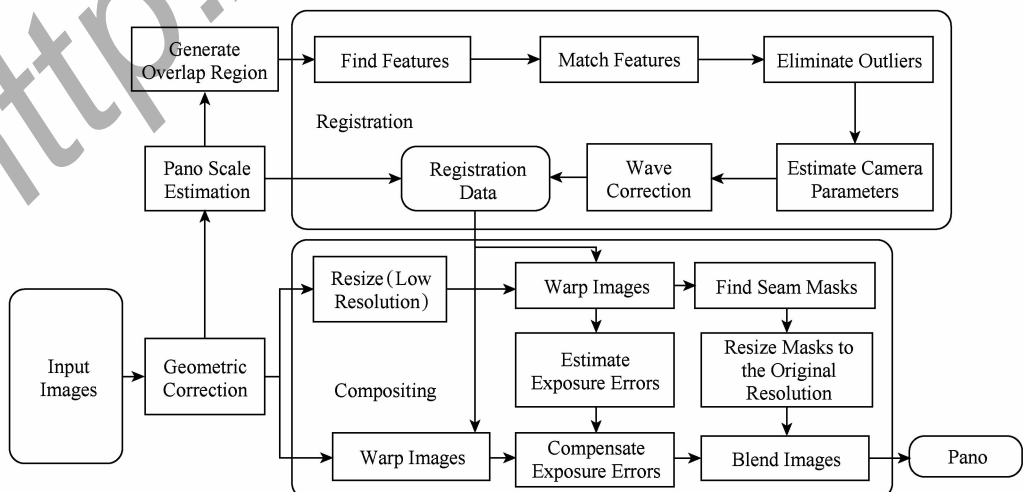


Fig. 1 Image stitching pipeline

图 1 图像拼接流程

率较 SIFT 算法提升了 100 倍. 姜桂圆等人^[8]提出一种 SIFT 特征的分布式并行提取算法 DP-SIFT, 该算法根据 SIFT 特征算法特点, 设计了高度宽度受限的数据块划分方法、数据分配方法以及特征信息调整方法, 并优化了数据分块原则与数据发送策略, 极大减少了数据通信时间, 提高算法的效率. 崔哲等人^[9]提出了在 CUDA 并行计算环境下的扩展 SURF 算法, 该算法在低分辨率下已经满足了实时性. Parker^[10]在 CUDA 并行计算环境下应用 ORB 算法进行 3D 重建. 智喜洋等人^[11]实现了基于 CUDA 的实时图像配准和定位系统, 较 CPU 快 20 倍, 且满足实时处理的要求.

创建全景图像需要经过很复杂的采集及拼接过程, 因此要求 CPU 和 GPU 都具有很强的处理能力^[12]. 传统的实时拼接设备配备了广角镜头和 FPGA 等处理组件, 其成像分辨率低且无法通过复杂算法对图像进行细致的矫正^[13]; 此外, 传统的基于软件的方法是不实时的^[14-16], 因为图像被摄像机捕获之后, 需要离线传输到拼接软件进行拼接. 这为广大学者提出了新的挑战. 本文主要工作如下:

1) 提出了 L-ORB 图像特征提取算法, 该算法优化分割了特征检测区域, 并对尺度不变性、旋转不变性进行了简化. 然后利用多探寻的局部敏感 Hash (Multi-Probe LSH) 算法对特征点进行匹配, 用改进的样本一致性 (progressive sample consensus, PROSAC) 算法消除误匹配, 得到帧图像拼接映射关系, 并采用多频带融合算法消除视频间的接缝.

2) 利用整合了 ARM A57 CPU 和 Maxwell GPU 的 Nvidia Jetson TX1 异构嵌入式系统中 Teraflops 的浮点计算能力和内建的视频采集、存储、无线传输模块, 实现了多摄像头视频信息的实时全景拼接系统. 有效地利用了 GPU 指令的块、线程、流并行策略对图像拼接算法进行加速. 实验结果表明本文的算法在图像拼接的特征提取、特征匹配等各个阶段均有很好的性能提升.

1 L-ORB 图像特征提取算法

传统的图像拼接需要对整个图片的内容进行特征点检测及匹配, 匹配到的特征点需要兼顾尺度不变与旋转不变性. 在全景摄像机群中, 相机的相对位置和方向固定不变, 通过预先矫正好的参数对图像进行预处理, 可以降低特征点检测的时空复杂度.

L-ORB 图像特征提取算法首先通过相机之间的位置参数对图像进行粗略的对齐, 然后计算图像的视野重合部分, 得出特征点的分布范围^[17], 减少了检测区域面积; L-ORB 算法将 FAST 特征点与 Harris 角点度量方法相结合, 并生成 BRIEF 特征描述因子, 相对于原始的 ORB 算法简化了尺度、旋转不变性, 使得效率大幅提升.

1.1 几何调整以及分割特征检测区域

图像拼接需要提取图片之间重叠区域的特征点, 我们通过全景摄像机群的相对位置计算出每个相机视野的重叠区域, 以减少特征点检测的数据规模. 全景摄像机群是将摄像机分散均匀地放置在一个点为圆心等半径的圆上, 再与顶部、底部的摄像机构成摄像机阵列^[18]. 如图 2 所示, A, B 为 2 架摄像机, 2 架摄像机所在圆的半径为 r , 角度为 θ , 摄像机视野的角度为 θ_{range} , L 为物体到摄像机的距离, 那么, 重影区对应的角度

$$\alpha = \pi - (\eta + \beta), \quad (1)$$

其中,

$$\eta = \frac{\pi + \theta - \theta_{\text{range}}}{2}, \quad (2)$$

$$\beta = \arctan \left(\frac{\sin \eta \times L}{L \times \cos \eta - 2 \times r \times \sin \frac{\theta}{2}} \right). \quad (3)$$

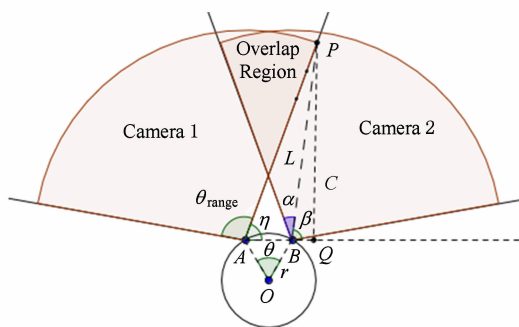


Fig. 2 Camera array

图 2 摄像机阵列

重影区的角度 α 随景物与摄像机距离 L 的增加而非线性增加, 最终收敛到一个固定值. 当景物与摄像机距离较近时, 重叠区域差异明显; 景物与摄像机较远时, 差异趋于稳定. 通过对重影区域角度极值的测试分析得出, 在不同摄像机群配置中, 重影区域的面积处于 10%~40% 之间, 如图 3 所示. 通过分割特征检测区域, 筛选出重影区域进行特征检测, 可缩短特征检测和特征匹配的时间.

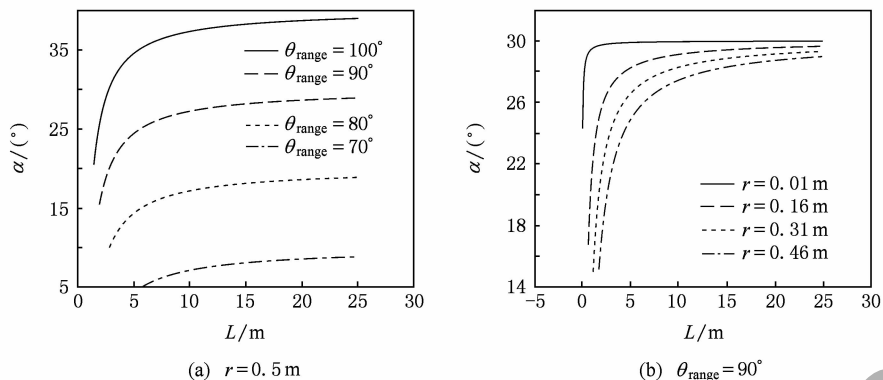


Fig. 3 Relationship between overlapping area and distance from object to camera

图 3 重影区域与物体到相机距离的关系

1.2 基于 Harris 特征的 FAST 角点检测

把图像重叠区域的图像进行粗略对齐后,图像匹配所需要的特征点不需要尺度、旋转不变性,因此我们对 ORB 算法中的 oFAST 和 rBRIEF 进行简化,去掉其中的尺度、旋转不变性以得到性能的提升。

FAST 特征是由 Rosten 等人提出的一种快速特征检测方法,不具有尺度不变性,且相比传统的 SIFT 和 SURF 方法具有明显的速度优势^[19]. FAST 算法检测的特征点定义为:若像素点 P 周围邻域内有足够多的连续像素点与该点相差较大,则认为该点是 FAST 特征点.但 FAST 特征点不具备角点的属性,因此需要利用 Harris 角点^[20]的度量方法,从 FAST 特征点中挑选出角点响应函数值最大的 N 个特征点,其中响应函数为 $R = \det M - \alpha(\text{trace } M)^2$. 该方法具体步骤如算法 1 所示。

算法 1. 基于 Harris 特征的 FAST 角点检测方法.

输入:图像 I ;

输出:图像中的 N 个角点.

- ① 计算图像 $I(x, y)$ 在 X 和 Y 两个方向的梯度 I_x, I_y ;
- ② 计算图像 2 个方向梯度的乘积;
- ③ 使用高斯函数对 I_x^2, I_y^2 和 I_{xy} 进行高斯加权 (取 $\sigma=1$), 生成矩阵 M 的元素 A, B 和 C
- ④ 计算图像 I 的 FAST 角点;
- ⑤ 计算 FAST 角点的 Harris 响应值 R , 并删除小于阈值 t 的角点;
- ⑥ 进行非极大值抑制, 邻域内局部最大值点即为 Harris 角点;
- ⑦ 获取 Harris 角点中响应值最大的 N 个角点.

1.3 BRIEF 图像特征描述因子

传统的 SIFT 与 SURF 特征采用 128 b 和 64 b

浮点型数据作为特征描述因子,将占用大量的存储空间且会增加特征匹配的时间. BRIEF 利用图像邻域内随机点对的灰度关系来建立图像特征描述因子,具有时间复杂度、空间复杂度低的特性^[21].

BRIEF 特征描述因子的建立首先需要平滑图像,然后在特征点周围一定范围内通过特定的方法来挑选出来 n_d 个点对. 对于每一个点对 (p, q) , 如果这 2 个点的亮度值 I 符合 $I(p) > I(q)$, 则这个点对生成的值为 1; 如果 $I(p) < I(q)$, 则对应在二值串中的值为 -1, 否则为 0. 对所有的点对就可以生成一个 n_d 长的二进制串.

当图像旋转角度大于 45° 时, BRIEF 图像特征描述因子的识别率几乎 0, 因此不具有旋转不变性; 但当旋转角度为 15° 以内时, 图像的识别率大于 70%, 且旋转角度越低, 识别率越高; 当旋转角度极小时, 识别率明显高于其他方法, 如图 4 所示. 在全景摄像机群中, 对齐后的重叠图像旋转幅度仅仅在 $0 \sim 15^\circ$ 的范围内, 使用 BRIEF 算法可以获得相对于传统方法更快的速度和更好的效果.

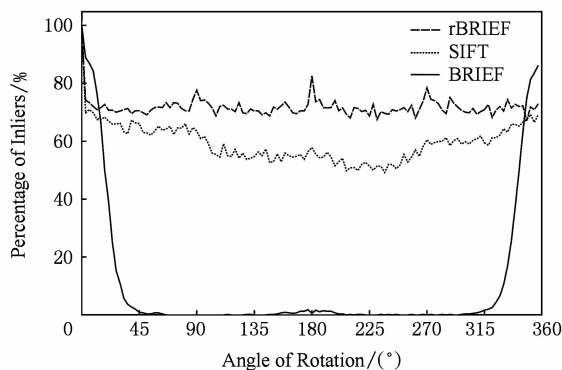


Fig. 4 Matching performance under synthetic rotations

图 4 图像旋转角度与特征识别率的关系

通过在 ORB 算法上对特征检测区域的优化分割,以及对尺度不变性、旋转不变性的简化,形成 L-ORB 图像特征提取算法,详见算法 2.

算法 2. L-ORB 图像特征提取算法.

输入: n 幅图像;

输出: P 组特征描述因子.

- ① 对全部的 n 幅图像分割为 m 个特征检测区域;
- ② 对 m 个特征检测区域进行 FAST 角点检测;
- ③ 从 FAST 特征点中挑选出 Harris 角点响应值最大的 P 个特征点;
- ④ 把 P 个 Harris 特征点建立 BRIEF 图像特征描述因子.

2 基于 LSH 的特征点匹配算法

特征点匹配是将 2 幅图片重影区域描述因子相同的特征点进行匹配,从而确定 2 张图片的对应关系.在图像重合后,其相匹配的特征点相邻的概率很大,因此通过近似最邻近查找(ANN)可以将搜索范围由全部特征点的集合减小为相邻特征点的集合,相对于线性搜索、 K 临近搜索(KNN)等算法具有更小的空间复杂度和时间复杂度.

2.1 局部敏感 Hash(Multi-Probe LSH)特征点搜索

LSH 是 ANN 中一种比较快速的方法.原始的 LSH 利用多个 Hash 函数通过对向量对象进行 Hash 映射对数据进行降维.通过对查询向量进行多次 Hash 操作,综合多个 Hash 表中的查询操作得到最终的结果.为了保证涵盖大部分近邻数据,原始的 LSH 索引需要建立很多 Hash 表,空间复杂度很高.Lü 等人提出 Multi-Probe LSH 算法^[22],使用一个经过仔细推导出的探测序列,得到和查询数据近似的多个 Hash 桶,增加了查找近邻数据的机会.

原始的 LSH 方法对查询数据 q 得到的 Hash 桶是 $g(q) = (h_1(q), h_2(q), \dots, h_m(q))$, 定义一个 Hash 扰动向量 $\Delta = (\delta_1, \delta_2, \dots, \delta_m)$, 可以探测到 Hash 桶 $g(q) + \Delta$. 在 LSH 函数 $h_{a,b}(V) = \left(\frac{a \times v + b}{W}\right)$ 中,相似的数据的映射差最多一个单位. Multi-Probe LSH 算法可以减少 Hash 表的规模以降低内存占有量,加快搜索速度.

2.2 PROSAC 特征点筛选

在图像匹配过程中,可能会有各种原因产生匹配错误的外点(outliers).如果在进行图像融合时引入了外点,则会给图像融合带来很大的错误.因此必

须去除外点,选择正确的内点(inliers)进行参数的估计.在这里采用改进的样本一致性(progressive sample consensus, PROSAC)^[23] 算法来去除误匹配点,使得接下来求得的参数矩阵会更加接近真实值.

传统的 RANSAC 算法随机抽取样本,忽略了样本之间的好坏差异,效率低下. PROSAC 算法将样本按质量进行排序,从较高质量的数据子集中抽取样本.经过若干次的假设、验证后,得到最优解.效率为 RANSAC 的 100 倍且鲁棒性更高.

基于 LSH 的特征点匹配算法由 Multi-Probe LSH 特征点搜索和 PROSAC 特征点筛选组成,其具体步骤如算法 3 所示.

算法 3. 基于 LSH 的特征点匹配算法.

输入: 2 幅图片的特征点集合;

输出: 筛选好的特征点对.

- ① 对特征点集合使用 Multi-Probe LSH 进行特征点匹配;
 - ② 利用 PROSAC 算法去除错误匹配.
- 重复以下步骤直到找到满意的结果
- I 将匹配好的特征点对按照匹配质量将数据从好到差排序,然后选取前 n 个较高质量的数据;
 - II 从 n 个数据中随机取出 m 个,计算模型参数和内点的数量;
 - III 验证模型参数.

3 基于 CUDA 的 GPU 并行化视频拼接算法

本文提出的 L-ORB 和 LSH 算法需要对图像进行复杂矩阵的运算,CPU 的串行处理模式性能无法满足实时性要求.统一计算架构(compute unified device architecture, CUDA)是由 NVIDIA 提出的并由其制造的图形处理单元(GPUs)实现的一种并行计算平台及程序设计模型,对大量并发线程的并发流架构具有很好的加速能力^[24-25].我们利用 GPU 的众核运算特性,把本文提到的算法由串行执行转化为并行执行,利用 CUDA 架构对并发的矩阵运算进行加速可以成倍提高视频拼接速度.

3.1 块、线程并行

CUDA 函数中存在块和线程 2 个层次的并行方式,各个块之间互相独立,但同一块中的线程可以通过共享存储器来交换数据. CPU 通过优先级和时间片轮转法实现线程调度;而 GPU 上线程只有等待资源和执行 2 种状态,如果资源满足运行条件就会立即执行.当 GPU 资源充裕时,所有线程都是并

发执行的,加速效果很接近理论加速比;而 GPU 资源少于总线线程个数时,有一部分线程就会等待前面执行的线程释放资源,从而变为串行化执行。

对 L-ORB 算法进行 CUDA 并行加速的关键是把算法原本串行计算的部分分割成多个子任务,对于 FAST 特征提取、非极大值抑制、建立 BRIEF 图像描述因子、图像的变换,均满足可拆分成多个计算过程相同且数据互不相关子任务的条件下,对全景视频实时拼接算法的 GPU 并行化设计如下:

算法 4. 全景视频实时拼接算法。

输入:多个摄像头同时采集不同方向的视频;

输出:全景视频。

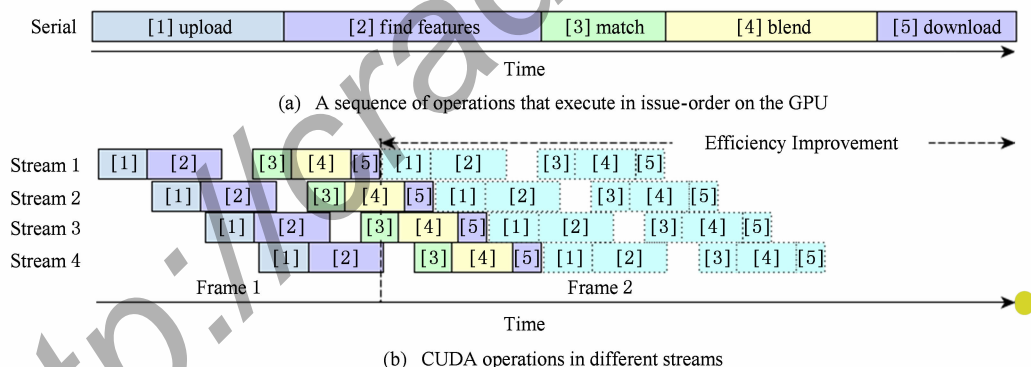
- ① 根据预先矫正的参数,对每个摄像头采集的视频利用 GPU 进行裁剪和变换;
- ② 对特征检测区域内的每个像素进行并行检测,筛选出符合 FAST 特征点的像素,并计算出其 Harris 响应值;
- ③ 利用 GPU 对每个符合 FAST 特征点的像素进行非极大值抑制;
- ④ 获取 Harris 角点中响应值最大的 P 个角点,并建立 BRIEF 图像特征描述因子;

- ⑤ 把 2 个视频中提取的角点进行匹配及筛选,并计算出变换矩阵;
- ⑥ 使用 GPU 对视频进行变换,并融合成全景视频。

3.2 流并行

图像拼接的串行 CUDA 编程模式分为 3 步:1)将图像从主内存上传到 GPU 内存;2)在 GPU 上顺序执行角点检测、特征匹配、图像融合;3)将结果从 GPU 内存下载到主内存,如图 5(a)所示。然而图像数据从主内存到 GPU 内存的传输速度依赖于 PCI-E 总线的带宽,数据传输过程中会导致 GPU 运算资源闲置,且 GPU 在执行单个任务时会导致流处理器资源闲置。

CUDA stream 把程序的指令分为多个操作队列,可以实现队列之间的操作并行。由于不同 stream 的操作是异步执行的,我们通过精心构造操作队列,使得队列之间互相协调来充分利用 GPU 资源。利用 CUDA stream,在上传第 2 张图像的同时可以对第 1 张图像进行角点检测;且可以对多副图像同时进行角点检测、特征匹配等 GPU 操作,大大节约了时间,提升了效率,如图 5(b)所示。



The solid frame represents the current frame of the image stitching, and the dashed frame represents the next frame of the image splicing.

Fig. 5 CUDA stream list

图 5 图像拼接的 CUDA 流并行队列

4 实验分析

本文 CUDA 和 CPU 实验在 Ubuntu 14.04 LTS 环境下使用配置为 Intel Core i7-6700HQ, RAM 16 GB, NVIDIA GeForce GTX 970M 的 Terrans Force X411 计算机运行;嵌入式实验使用 NVIDIA TX1 开发板;使用了 NVIDIA CUDA Toolkit 8.0 和 OpenCV 3.2.0 等软件开发包。

4.1 准确性实验

为验证本文提出算法的准确性,我们使用了 Adobe Panoramas Dataset 等图片集进行了图片拼接测试,如图 6(a)~(b)所示。图 6(c)~(e)分别为传统的 SIFT、传统的 ORB 和本文的 L-ORB 特征点提取及匹配的结果。其中,圆圈表示算法提取的特征点,连线表示算法匹配到的特征点对,三角形表示匹配错误的特征点对。从图 6(c)~(e)中可看出,SIFT 算法提取了大量的特征点,但匹配的特征点对占比

较少;传统的 ORB 算法准确率不高,产生了错误匹配;本文的 L-ORB 算法在相较于 SIFT 算法提取较少特征点的前提下,匹配到大量有效的的特征点,且

准确率高于传统的 ORB 算法.图 6(f)为 2 张图片拼接后的结果.实验证明我们的算法能够有效地匹配到有效的特征点并完成图像拼接.

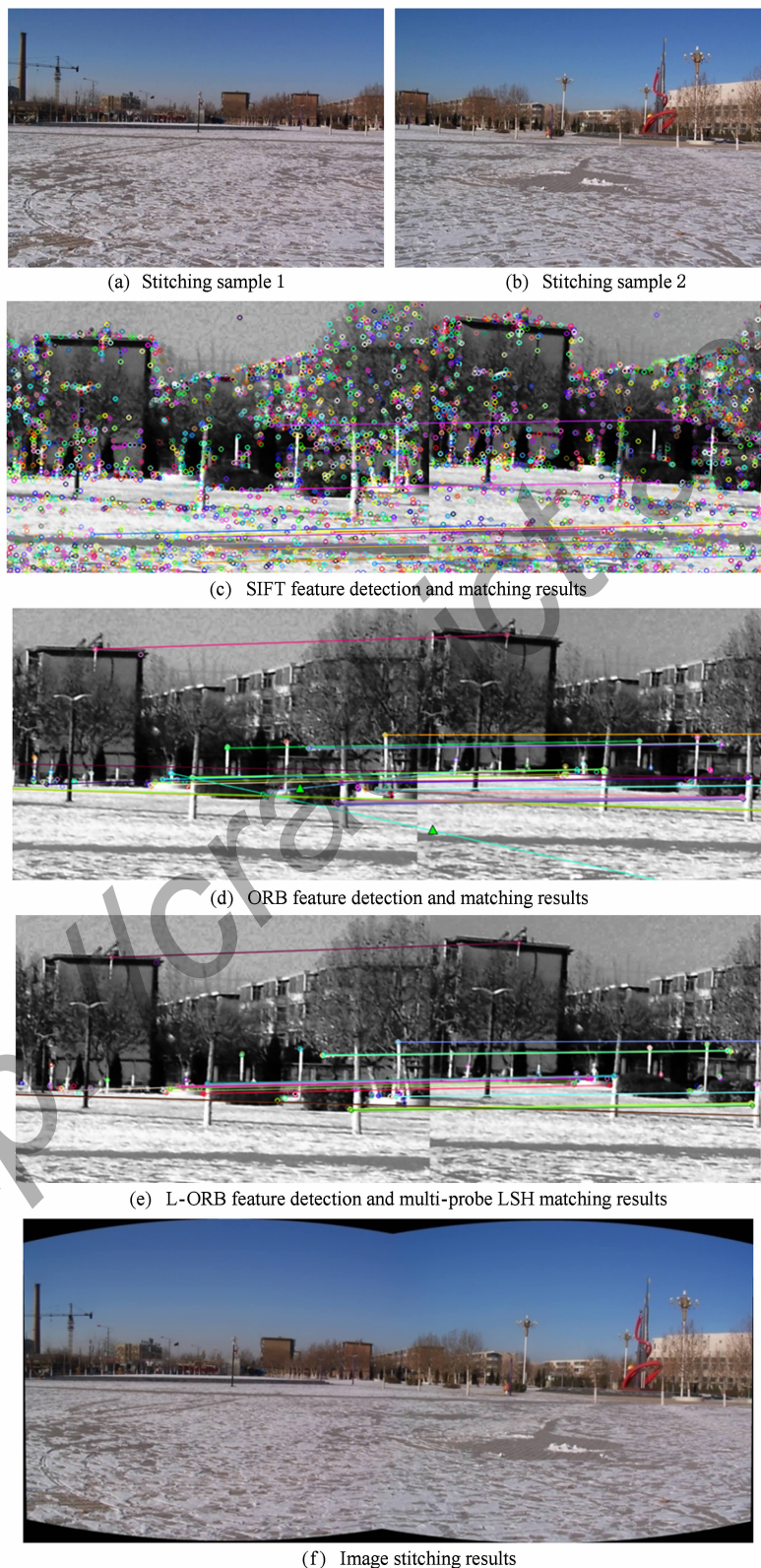


Fig. 6 Stitching sample matching results and stitching results

图 6 图像拼接的样本、特征点匹配结果及拼接结果

4.2 实时性实验

为了验证本文提出算法的实时性,我们进行了一系列实验. 我们用 SIFT, SURF, ORB, L-ORB 对分辨率为 800×600 , 1920×1080 , 2034×1728 像素的视频进行特征点检测, 如图 7 所示. 结果证明, 在 Intel i7 2.8 GHz 单线程下, 采用本文的 L-ORB 特征提取算法可将时间缩短为 ORB 算法的 1/3, 是传统 SIFT 算法的 1/1000.

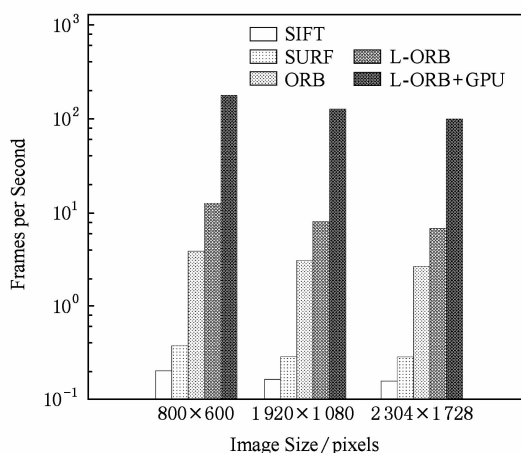


Fig. 7 Comparison of feature detection algorithm efficiency

图 7 不同算法对图像特征提取效率的对比

我们使用 SIFT, SURF, ORB, L-ORB, L-ORB+GPU 分别对实时采集的 2304×1728 pixels 视频进行拼接, 记录每个阶段的拼接时间. 实验证明, 本文提出的 L-ORB+GPU 算法在图像拼接的特征提取、特征匹配等各个阶段均有很好的性能提升, 其算法速度是传统 ORB 算法的 11 倍、传统 SIFT 算法的 639 倍, 如图 8 所示:

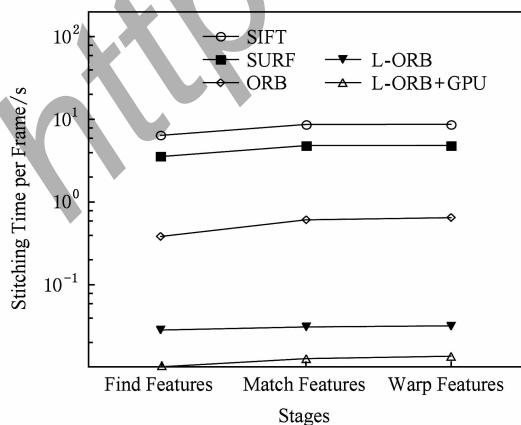
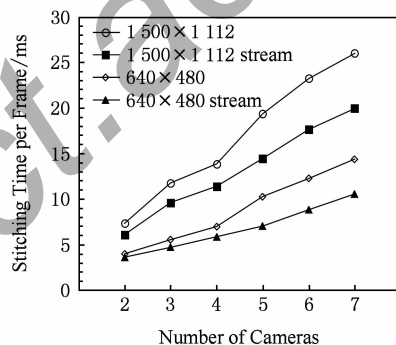


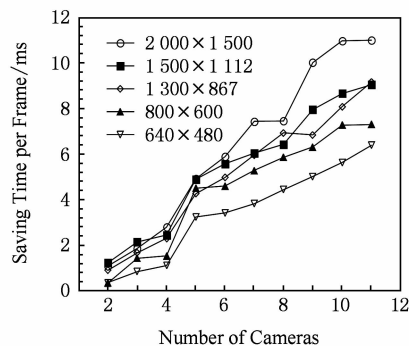
Fig. 8 Comparison of different algorithms in three stages of image stitching

图 8 不同算法在图像拼接 3 个阶段的时间对比

为了验证 CUDA Stream 的有效性, 我们使用 NVIDIA TX-1 分别对不同分辨率、不同摄像头个数、长度为 1000 帧的视频进行拼接, 并计算出拼接每帧图片时间的平均值. 图 9(a) 展示了是否使用 CUDA Stream 对 2~7 个视频进行拼接的时间, 实验证明, CUDA Stream 对各种分辨率的视频拼接均有明显的加速效果. 图 9(b) 展示了拼接 2~11 个视频流时使用 CUDA Stream 与未使用 CUDA Stream 平均每帧节约的时间, 因为不同分辨率视频采集的内容不同, 提取角点数量不同, 且磁盘 I/O 和图像的压缩算法都会影响拼接时间, 在此不对不同分辨率做比较. 实验证明, 在 GPU 资源足够时, 性能随着视频流的增多而提高.



(a) Comparison between CUDA stream and traditional algorithm



(b) Saving time using CUDA stream

Fig. 9 Performance improvement using CUDA stream

图 9 CUDA Stream 加速的性能提升

我们使用 L-ORB 算法分别在 BeagleBone Black, Raspberry Pi 3B, NVIDIA TX-1 CPU, NVIDIA TX-1 GPU 四种嵌入式开发版中对大小为 2304×1728 pixels 的同一组数据集进行图像拼接实验. 实验结果如图 10 所示, 本文提出的算法在 NVIDIA TX-1 GPU 中的拼接速度较同类速度最快的 Raspberry Pi 3B 提升 29.2 倍.

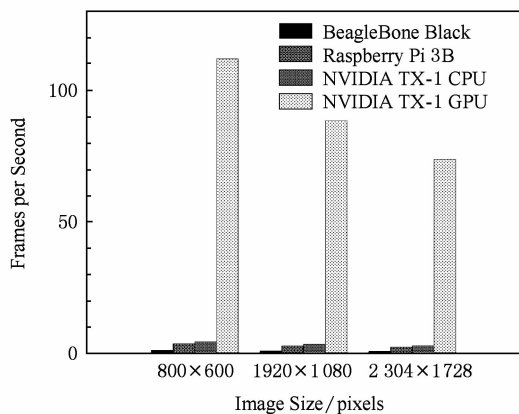


Fig. 10 Comparison of image splicing time in embedded devices

图 10 嵌入式设备中图像拼接时间对比

5 总 结

本文提出了 L-ORB 特征提取算法,该算法减少了特征点检测区域,简化了传统 ORB 算法对尺度、旋转不变性的支持,降低了算法的时间复杂度;并利用 GPU 块、线程、流并行方法对该算法进行加速优化,进一步提升了算法的运行效率.此外还设计实现了一种全景视频实时拼接系统,解决了传统方法不能满足全景视频拼接实时性的问题.实验结果表明,本文提出的方法可以频率为 60 Hz 的视频进行拼接,满足实时性.我们进一步的研究工作将对利用剩余运算资源对融合算法进行优化,提高全景视频质量.

参 考 文 献

[1] Gaddam V R, Riegler M, Eg R, et al. Tiling in interactive panoramic video: Approaches and evaluation [J]. IEEE Trans on Multimedia, 2016, 18(9): 1819-1831

[2] Wang X, Tieu K, Grimson W E L. Correspondence-free multi-camera activity analysis and scene modeling [C] //Proc of 2008 IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2008: 1-8

[3] Wang Xiaogang. Intelligent multi-camera video surveillance: A review [J]. Pattern Recognition Letters, 2013, 34(1): 3-19

[4] Wu Yuezhou, Liu Changjiang, Lan Shiyong, et al. Real-time 3D road scene based on virtual-real fusion method [J]. IEEE Sensors Journal, 2015, 15(2): 750-756

[5] Zeng Dan, Chen Jian, Zhang Qi, et al. Global topology based image stitching using hierarchical triangulation [J]. Journal of Computer Research and Development, 2012, 49(1): 144-151 (in Chinese)

(曾丹, 陈剑, 张琦, 等. 基于全局拓扑结构的分级三角剖分图像拼接[J]. 计算机研究与发展, 2012, 49(1): 144-151)

[6] Cao Shixiang, Jiang Jie, Zhang Guangjun, et al. Multi-scale image mosaic using features from edge [J]. Journal of Computer Research and Development, 2011, 48(9): 1788-1793 (in Chinese)

(曹世翔, 江洁, 张广军, 等. 边缘特征点的多分辨率图像拼接[J]. 计算机研究与发展, 2011, 48(9): 1788-1793)

[7] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF [C] //Proc of 2011 Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2011: 2564-2571

[8] Jiang Guiyuan, Zhang Guiling, Zhang Dakun. A distributed parallel algorithm for SIFT feature extraction [J]. Journal of Computer Research and Development, 2012, 49(5): 1130-1141 (in Chinese)

(姜桂圆, 张桂玲, 张大坤. SIFT 特征分布式并行提取算法[J]. 计算机研究与发展, 2012, 49(5): 1130-1141)

[9] Cui Zhe, Meng Fanrong, Yao Rui, et al. Multi-video fusion with extended SURF based on CUDA parallel computing framework [J]. Journal of Nanjing University (Natural Sciences), 2016, 52(4): 627-637 (in Chinese)

(崔哲, 孟凡荣, 姚睿, 等. CUDA 并行计算下基于扩展 SURF 的多摄像机视频融合方法[J]. 南京大学学报: 自然科学版, 2016, 52(4): 627-637)

[10] Parker C, Daiter M, Omar K, et al. The CUDA LATCH binary descriptor: Because sometimes faster means better [C] //Proc of 2016 Computer Vision-ECCV Workshops. Berlin: Springer, 2016: 685-697

[11] Zhi Xiyang, Yan Junhua, Hang Yiqing, et al. Realization of CUDA-based real-time registration and target localization for high-resolution video images [J]. Journal of Real-Time Image Processing, 2016: 1-12

[12] Brown M, Lowe D G. Automatic panoramic image stitching using invariant features [J]. International Journal of Computer Vision, 2007, 74(1): 59-73

[13] Peng Bo, He Bin. Application and realization of FPGA in video mosaicing [J]. Computer Engineering and Design, 2013, 34(5): 1635-1639 (in Chinese)

(彭勃, 何宾. FPGA 在视频拼接中的应用与实现[J]. 计算机工程与设计, 2013, 34(5): 1635-1639)

[14] Xiong Y, Pulli K. Fast panorama stitching for high-quality panoramic images on mobile phones [J]. IEEE Trans on Consumer Electronics, 2010, 56(2): 298-306

[15] Cha J H, Jeon Y S, Moon Y S, et al. Seamless and fast panoramic image stitching [C] //Proc of 2012 IEEE Int Conf on Consumer Electronics (ICCE). Piscataway, NJ: IEEE, 2012: 29-30

[16] Anderson R, Gallup D, Barron J T, et al. Jump: Virtual reality video [J]. ACM Trans on Graphics, 2016, 35(6): 198

- [17] Zhu Z, Hanson A R. Mosaic-based 3D scene representation and rendering [J]. *Signal Processing: Image Communication*, 2006, 21(9): 739-754
- [18] Tzavidas S, Katsaggelos A K. A multicamera setup for generating stereo panoramic video [J]. *IEEE Trans on Multimedia*, 2005, 7(5): 880-890
- [19] Rosten E, Drummond T. Machine learning for high-speed corner detection [C] // *Proc of European Conf on Computer Vision*. Berlin: Springer, 2006: 430-443
- [20] Harris C, Stephens M. A combined corner and edge detector [C] // *Proc of the Alvey Vision Conf*. Manchester, UK: Alvey Vision Club, 1988: 147-151
- [21] Calonder M, Lepetit V, Strecha C, et al. Brief: Binary robust independent elementary features [C] // *Proc of European Conf on Computer Vision*. Berlin: Springer, 2010: 778-792
- [22] Lv Qin, Josephson W, Wang Zhe, et al. Multi-probe LSH: Efficient indexing for high-dimensional similarity search [C] // *Proc of the 33rd Int Conf on Very Large Data Bases*. New York: ACM, 2007: 950-961
- [23] Chum O, Matas J. Matching with PROSAC-progressive sample consensus [C] // *Proc of 2005 IEEE Computer Society Conf on Computer Vision and Pattern Recognition (CVPR'05)*. Piscataway, NJ: IEEE, 2005: 220-226
- [24] Lončar V, Balaž A, Bogojević A, et al. CUDA programs for solving the time-dependent dipolar Gross-Pitaevskii equation in an anisotropic trap [J]. *Computer Physics Communications*, 2016, 200 (2016): 406-410
- [25] De Angelis F, Gentile F, Mecarini F, et al. Breaking the diffusion limit with super-hydrophobic delivery of molecules to plasmonic nanofocusing SERS structures [J]. *Nature Photonics*, 2011, 5(11): 682-687



Du Chengyao, born in 1992. MSc candidate. His main research interests include computer architecture and computer vision.



Yuan Jingling, born in 1975. PhD. Professor, PhD supervisor. Senior member of CCF. Her main research interests include green computing, machine learning and data mining.



Chen Mincheng, born in 1990. PhD candidate. His main research interests include green computing and data mining.



Li Tao, born in 1972. PhD. Professor, PhD supervisor. His main research interests include computer architecture and green computing.

<http://crad.iit.ac.cn>