

基于 GPU 的散斑三维重建系统

韩磊¹ 徐波¹ 黄向生¹ 张彦峰²

(中国科学院自动化研究所 北京 100190)¹

(武汉大学计算机科学与工程学院 武汉 100039)²

摘要 散斑相关算法可以用来估计场景的深度信息,但因易受到噪声干扰且计算量大而难以应用在基于普通计算机的三维重建系统中。采取零均值归一化互相关函数(ZNCC)作为相关算法的匹配代价函数,对传统的ZNCC快速计算方法进行修改并将其应用于计算机的通用图形处理器(GPU),实现了实时的场景三维重建效果。对比实验表明,在精度一致的前提下,提出的GPU计算方法的速度是CPU算法的39倍。

关键词 结构光,图形处理器,散斑,三维重建

中图分类号 TP391.41 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.8.060

Speckle Projection Systems Based on GPU

HAN Lei¹ XU Bo¹ HUANG Xiang-sheng¹ ZHANG Yan-feng²

(Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)¹

(Department of Computer Science and Engineering, Wuhan University, Wuhan 100039, China)²

Abstract Speckle correlation algorithms can be used to estimate the depth information of the scene. However, such methods are easy to be disturbed by noises and inherent with high computational cost. This paper presented a 3D-reconstruction system based on structured light implemented on GPU. Depth was estimated from the correlation of speckle projection image. Zero-mean normalized cross correlation(ZNCC) was adopted as the correlation function. Traditional fast-ZNCC calculation method was modified to implement on GPU platforms. Experimental results show a 39x speed-up is achieved using GPU compared with the same computation cost implemented on CPU platforms.

Keywords Structured light, Graphic processing unit, Speckle, 3D-reconstruction

场景的三维数据可以被广泛应用于机器视觉、物体分割、人机交互等诸多领域,通过对场景的色彩、位置、深度信息的综合考虑,可以在很大程度上简化传统计算机视觉面临的问题。快速、准确地获取场景的三维信息一直是计算机视觉领域中的热点与难题之一。传统的双目视觉或者激光测距方法由于计算复杂度高、要求场景中有明显的纹理变化或者硬件成本高昂等原因,难以实现室内应用领域的实时三维重建。

散斑作为实现三维重建的一种解决方案,由于对场景纹理的无约束性(室内场景)、硬件组成简单而备受瞩目。使用散斑进行三维重建的一般方法是:将分布规律已知的结构光投影在场景上,使用散斑相关算法,统计每一个像素点的相对位移即视差,通过视差分析出场景在该像素点对应位置处的深度信息。然而,这种算法的计算复杂度过高,使用CPU难以达到实时进行三维重建的目的;而且由于结构光的随机分布,单个像素上的噪声很容易影响该像素临近位置点上的深度信息统计。本文通过基于散斑截断和自适应滤波的方法对散斑图像进行预处理以保证三维重建效果的鲁棒性,并使用

GPU作为加速计算的工具。

本文第1节介绍了使用散斑进行三维重建的相关工作;第2节介绍了三维重建计算系统的硬件组成和重建算法;基于GPU的重建算法在第3节中进行深入解释;第4节为实验部分;最后是总结和对下一步工作的展望。本文的主要贡献在于:把传统的基于CPU的散斑三维重建算法在GPU上实现,设计了一套更适合GPU的计算框架。在这套计算框架的基础上,可以使用普通的GPU实现实时的场景三维重建效果。

1 相关工作

从20世纪80年代开始,已有大量基于结构光进行三维重建的相关论文发表[6-8],其中,文献[6]是一篇很好的关于使用光学方法进行三维重建的文献综述,文献[7]总结整理了结构光三维重建方法在20世纪最后20年的快速发展以及面向市场的三维重建设备。散斑也是结构光的一种,通过激光发射器向场景中投射特定分布的肉眼不可见的红外散斑,红

到稿日期:2014-08-31 返修日期:2015-02-25 本文受国家高技术研究发展计划(863计划):基于超多视角成像的三维重建关键技术研究(61175034),国家杰出青年科学基金(61103154)资助。

韩磊(1992-),男,主要研究方向为三维重建、机器视觉,E-mail:15210560937@163.com;徐波(1966-),博士,研究员,博士生导师,主要研究方向为模式识别、语音识别,E-mail:xubo@ia.ac.cn;黄向生(1975-),博士,副研究员,主要研究方向为三维重建、机器视觉,E-mail:xiangsheng.huang@ia.ac.cn;张彦峰(1990-),博士生,主要研究方向为三维重建、机器视觉,E-mail:15210560937@163.com。

外相机被用来捕捉散斑图像,通过散斑相关算法确定每一个像素点上的视差值,使用三角化测量^[12]的方法根据视差推测出场景的深度信息。基于散斑的三维重建系统的工作原理如图 1 所示(相机成像平面实际是在相机中心的左侧,为了便于表述,将其对称地放到相机中心右侧)。

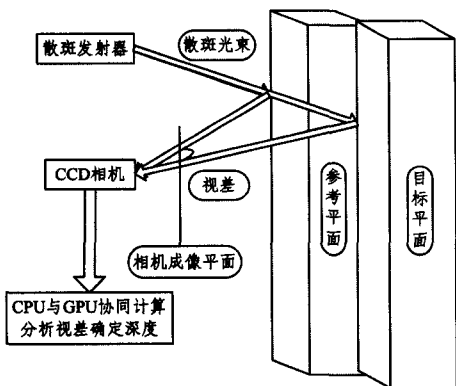


图 1 基于散斑的三维重建系统

与立体视觉匹配算法中的局部窗口算法^[5]类似,散斑相关算法通过比较目标图像和参考图像在不同视差下平移图像的相关程度来确定相关程度,选择相关程度最高的平移图像所对应的视差作为像素点的视差。根据 Scharstein^[5]的总结,选取了零均值归一化自相关函数(ZNCC)作为相关函数。文献[20]介绍了 ZNCC 的快速计算方法,将 ZNCC 函数公式组织成区域和的方式,通过积分图算法快速计算区域和再计算相关函数。然而积分图算法并不适合于 GPU 并行平台的处理,Berkin^[20]将积分图分解成二维的前缀和计算,通过上行扫描和下行扫描两次遍历操作实现前缀和计算过程,实现了基于 GPU 的积分图计算方法。本文在这种方法的框架下,参考了 Wang^[16]将积分图分段处理的策略,在计算积分图的过程中计算出区域和,实现了基于 GPU 的散斑相关函数的高效计算。

直接将散斑图像进行相关容易受到高亮度噪声点的影响。Wang^[10]将散斑二值化,通过分析散斑的分布规律来进行相关,然而这种方法丢失了散斑亮度变化的大量有效信息,而且需要精巧设计的散斑相关算法来分析散斑的分布规律。本文提出的对散斑图像进行预处理的方法计算逻辑简单,可以在通用图形处理器上运行以实现加速效果,通过截断的方法减小高亮度噪声点的影响,通过自适应滤波的方法消除了背景噪声的干扰。对比实验表明此方法可以明显改善三维重建效果。

2 散斑三维重建算法

三维重建系统通过红外散斑激光发射器向场景发射散斑,由红外相机、捕捉散斑的亮度信息。相机拍摄的散斑图像如图 2 所示。

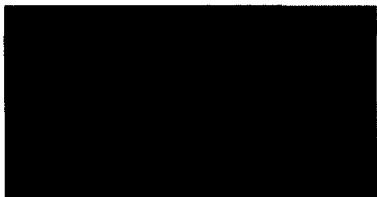


图 2 散斑效果图

散斑投射器和摄像机的光轴是平行的。如图 1 所示,同一束散斑,打在不同深度的平面上,在相机上的成像位置会发生横向偏移,这种横向偏移被称为视差,结合三角化测量原理可以将视差转化为深度信息。

预先存储的打在垂直于相机光轴的已知深度的平面上的散斑图像被称为参考图像,相机捕获的场景图像被称为目标图像。三维重建的流程如图 4 所示。首先对参考图像和目标图像进行预处理以减少噪声点的影响。通过散斑相关算法计算视差,具体做法是对参考图像横向平移,得到不同层次的参考图像。对于目标图像上的像素点,分别比较目标图像与每一层参考图像的相关程度,选择相关值最大的参考图像所对应的位移层次作为该像素点上的视差,使用线性插值来提高视差估计的像素精度,将视差转换为深度,最后使用中值滤波减少噪声点的干扰,优化重建效果。下面分小节来介绍每一步的具体计算过程。

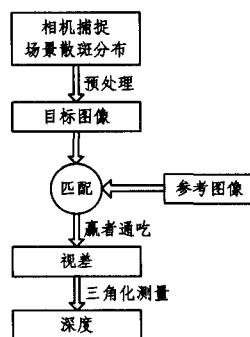


图 3 三维重建流程

2.1 预处理

对相机采集到的目标图像的预处理主要包括两个部分:截断和自适应滤波。截断的目的是减小亮度过强的点对于匹配代价的影响,函数化表示为:

$$I(x, y) = \begin{cases} I(x, y), & I(x, y) < T \\ T, & \text{otherwise} \end{cases} \quad (1)$$

$I(x, y)$ 是目标图像亮度值, x, y 为像素坐标。 T 为预设的截断阈值,一般设置为可以区分开亮斑与背景噪声即可,通过实验验证其设置为 16 就可以取得合适的截断效果。

自适应滤波通过将不属于亮斑的背景点置零,来突出散斑中的亮点与背景的区别。判断像素点是否属于亮斑的标准在于背景点的亮度小于以像素为中心的邻域内亮度平均值。自适应滤波的函数化表示如下:

$$I(x, y) = \begin{cases} I(x, y), & \text{if } I(x, y) \geq \frac{\sum_{i=-l}^l \sum_{j=-l}^l I(x+i, y+j)}{(2l+1)^2} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

经过实验验证,这两种预处理方法针对散斑三维重建的应用背景可以取得有效的优化效果。优化效果会在第 4 节加以展示与说明。

2.2 匹配

ZNCC 相关函数的公式化表述为:

$$C_{ZNCC}(x, y, d) = \frac{\sum_{j=-l}^l \sum_{i=-l}^l [(I_{obj}(x+i, y+j) - \overline{I_{obj}(x, y)}) (I_{ref}^d(x+i, y+j) - \overline{I_{ref}^d(x, y)})]}{\sqrt{\sum_{j=-l}^l \sum_{i=-l}^l (I_{obj}(x+i, y+j) - \overline{I_{obj}(x, y)})^2 \sum_{j=-l}^l \sum_{i=-l}^l (I_{ref}^d(x+i, y+j) - \overline{I_{ref}^d(x, y)})^2}} \quad (3)$$

表1 符号说明

符号	作用	取值
W	图像的宽	1024
H	图像的高	768
D	视差空间范围	80
l	局部窗口半径	12

$C_{ZNCC}(x, y, d)$ 表示目标图像上 (x, y) 点在视差 d 处的匹配程度, I_{obj} 表示目标图像, I_{ref}^d 表示第 d 层参考图像, $\overline{I_{obj}(x, y)}$ 表示目标图像在以 (x, y) 为中心、 l 为半径的正方形窗口内的灰度平均值, $\overline{I_{ref}^d(x, y)}$ 表示目标图像在以 (x, y) 为中心、 l 为半径的正方形窗口内的灰度平均值。

ZNCC快速算法^[20]通过减少重复计算加速相关函数的次数,可以将ZNCC相关函数重写成:

$$C_{ZNCC}(x, y, d) = \frac{(N * S_{Ij}^d - S_I * S_J)}{\sqrt{(N * S_{II} - S_I^2) * (N * S_{JJ} - S_J^2)}}$$

$$S_{Ij}^d(x, y) = \sum_{j=-l}^l \sum_{i=-l}^l (I_{obj}(x+i, y+j) * I_{ref}^d(x+i, y+j))$$

$$S_I(x, y) = \sum_{j=-l}^l \sum_{i=-l}^l I_{obj}(x+i, y+j)$$

$$S_J(x, y) = \sum_{j=-l}^l \sum_{i=-l}^l I_{ref}^d(x+i, y+j) \quad (4)$$

$$S_{II}(x, y) = \sum_{j=-l}^l \sum_{i=-l}^l I_{obj}(x+i, y+j)^2$$

$$S_{JJ}(x, y) = \sum_{j=-l}^l \sum_{i=-l}^l I_{ref}^d(x+i, y+j)^2$$

$S_{Ij}/S_I/S_J/S_{II}/S_{JJ}$ 被称为区域和图像,通过积分图^[2]算法可以快速计算。

通过比较像素点在不同层参考图像上的匹配值,选择最佳匹配位置作为像素点的视差,用于深度的计算。

2.3 线性插值与三角化测量方法

文献[10, 16]介绍并比较了不同的插值函数,本文选择了线性插值方法来达到次像素精度的视差估计。

$$C_L = |C_{ZNCC}(x, y, d_1 - 1) - C_{ZNCC}(x, y, d_1)|$$

$$C_R = |C_{ZNCC}(x, y, d_1 + 1) - C_{ZNCC}(x, y, d_1)| \quad (5)$$

$$d_2 = \begin{cases} d_1 - 0.5 + \frac{C_L}{C_R}, & C_L \leq C_R \\ d_1 + 0.5 - \frac{C_R}{C_L}, & C_L > C_R \end{cases}$$

d_1 表示整像素点精度的视差, d_2 表示次像素点精度的视差。

三角化测量原理^[12]可以用来从视差估计深度信息:

$$Z = \frac{Z_0}{1 + \frac{d_2}{s} Z_0} \quad (6)$$

式中, Z_0 指参考图像深度; s 是相机的固有参数,可以通过标定的方法确定。

3 三维重建算法的GPU实现

ZNCC快速计算方法依然有很高的计算复杂度。为了实现高精度实时三维重建系统,本文使用NVIDIA公司提出的统一设备计算架构(Compute Unified Device Architecture, CUDA)编程模型,将第2节介绍的三维重建算法在通用图形处理器(Graphic Processing Unit, GPU)上实现,利用其多核的特点实现并行计算,达到实时获取场景三维信息的目的。3.1节介绍了CUDA编程模型的优缺点,3.2节介绍了将第2节的三维重建算法在GPU上实现所采取的改进算法和实现思路。为了表述方便,表1列出符号说明。

3.1 CUDA

相比于CPU单指令单数据(SISD)的执行方式,GPU利用上百个计算核心以单指令多数据(SIMD)的方式对大量的数据执行相同的操作,实现细粒度(数据)的并行计算方式。NVIDIA公司提出的CUDA编程模型^[19],把对于GPU的编程简化为C语言的扩展,对数据的操作放在核函数里,一个核函数被大量的线程同时执行,每个线程有单独的索引以处理不同的数据。一定数量的线程被划分成线程块。每个线程有私有寄存器,同一个线程块内的线程可以通过共享内存传递信息。

使用GPU的好处在于GPU有大量的计算单元可以实现上百个核心的并行计算,但也有其局限性:CPU与GPU使用不同的内存,而且不可以相互访问,CPU与GPU之间的数据交换只能通过系统总线实现,速度很慢;GPU对于主存的访问很慢;GPU上每一个线程可以利用的高速缓存有限。受到这些因素的限制,GPU不可能像CPU一样灵活地对数据进行操作,在设计算法的GPU实现时必须减少每一步骤之间的数据依赖性。

3.2 基于GPU的三维重建算法

重建算法使用GPU作为计算工具,通过优化算法中最耗时的环节来达到加速的目的。为了减少内存与显存交换数据的时间消耗,算法完全使用GPU作为计算资源。只有作为输入的目标图像和作为输出的深度结果在CPU和GPU内存之间相互传递。

基于GPU的三维重建算法的过程与第2节中图3所示的三维重建流程相同,都是流水线式地对输入的散斑图像进行预处理,然后与参考图像进行匹配找到最佳匹配位置作为视差,根据视差计算出相应的深度信息。与CPU算法不同的是,GPU实现的算法更加强调并行化。在第2节中介绍的预处理算法、线性插值和三角化测量方法中每个像素点都是独立进行的,其本身就具有并行化的特性,而且计算复杂度不高。本节将重点介绍三维重建算法中最耗时的两个步骤:目标图像与参考图像的匹配过程和赢家通吃策略的实现。同时,这两个步骤也是难以并行化的。其中,匹配过程采用了快速ZNCC相关算法,具体计算过程又可以分为计算区域和和计算匹配代价。本文结合GPU计算特点,为了最大限度地重复利用线程从主存中访问到的数据,提出了如下的计算框架:

(1)计算区域和,分成两步完成:先计算行区域,再对行区域和图像计算列区域和,得到真正的区域和图像。区域和图像的形象化表示如图4所示。

(2)一个线程计算一个像素点在不同视差下的匹配代价,这样可以在计算匹配代价的同时实现赢家通吃策略,找出最佳匹配视差,在此线程内再进行线性插值和三角化测量。

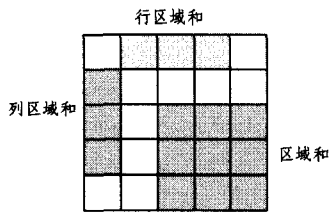


图4 区域和的形象化表示

这种计算框架抛弃了使用积分图来计算区域和图像的传统方法,优势在于在不增加计算复杂度的同时,直接计算出区域和,而不再根据在以像素点为中心的矩形区域内4顶点处的积分图像值之差的和来计算区域和,从而显著减少了GPU计算核心与显存的数据交换。同时,在计算匹配代价的同时实现赢家通吃策略,每个像素点在计算完匹配代价后立即进行代价函数的比较,使得GPU上每一个线程的被利用率更高。

在计算与匹配代价有关的区域和中, S_{ij} 是目标图像与每一个整数像素位移得到的参考图像之间点乘的区域和,如式(4)所示, S_{ij} 的计算量最大。在介绍区域和的计算过程中将以 S_{ij} 的计算过程为重点。

控制GPU线程执行计算的函数被称为核函数。按照本文提出的计算框架,主要的计算过程被分成3个核函数依次执行。第一个核函数计算目标图像与不同视差所对应的参考图像的点积,并计算出结果的行区域和和图像;第二个核函数以行区域和和图像为输入,计算输入的列区域和和图像,即 S_{ij} ;第三个核函数依次计算目标图像与所有视差所对应的参考图像的ZNCC代价函数,并通过比较得出最佳匹配位置,进而进行线性插值、视差深度转换的计算。

GPU的计算过程以线程为最小计算单元,以线程块为最小功能实现单位。线程块之间的线程可以通过共享内存来进行数据交换,从而提高数据重复利用率。

核函数一:计算行区域和图像

目标图像的一行保存在线程的局部寄存器上,原始参考图像的一行保存在共享内存中,线程中的局部寄存器通过不同的偏置可以读取不同视差所对应的参考图像。目标图像与参考图像在线程内进行点乘,并参考了Wang^[10]基于分段前缀和的方法计算行前缀和,将行前缀和放入共享内存中,行区域首位所对应像素的前缀和之差即为行区域和。

线程的具体分配方法是:一个线程块有 $W/8$ 个线程, H 个线程块并发执行。

核函数二:计算列区域和图像

每一个线程负责一列区域和的计算。遍历当前列,通过滑动窗口的方法计算列前缀和。具体做法是:在线程中依次扫描相应列的像素值,把当前像素位置的列区域和保留在寄存器中,计算下一个像素点的列区域和时,读取下一个像素点窗口内下边缘的像素值和当前像素点窗口内上边缘的像素值,当前列区域和加上 x 减去 y 就是下一个像素点的列区域和。

线程的具体分配方法是:一个线程块有256个线程, $\frac{W}{256} * D * H$ 个线程块并发执行。

核函数三:相关代价的计算与视差深度转换

一个线程对应一个像素点, W 个线程组成线程块对应重建图像中的一行。目标图像所对应的区域和 S_I/S_{II} 被放到线程块的共享内存中,通过移位得到的不同层的参考图像所对应的区域和 S_I/S_{II} 被放到线程寄存器中。遍历计算每一层参考图像与目标图像的ZNCC相关代价。在遍历过程中使用寄存器记录并更新当前像素点对应的最佳匹配位置和匹配代价,以及最佳匹配位置左邻层、右邻层的匹配代价。遍历结束后,寄存器中的结果直接用于视差的线性插值和深度转换。将深度结果存储在寄存器中。

线程的具体分配方法是:一个线程块有 W 个线程, H 个线程块并发执行。

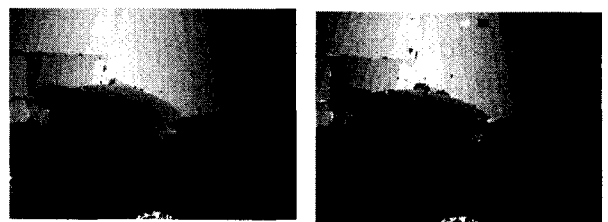
4 实验

本文通过重建效果与重建速度两方面来评价三维重建效果。首先通过分析系统在不同深度下对平面的三维重建效果来判断系统的实用性;将算法分别在GT 650M GPU和Intel i7 3630m CPU平台上进行实现,测量系统在一定分辨率下的三维重建速度。本文还通过对比实验验证了预处理能明显改善三维重建的效果。实验证实本文提出的算法在重建精度满足实际应用需求的前提下,利用普通图形处理器可以取得高精度(1024×768分辨率)的实时三维重建效果。

4.1 预处理效果

为了证明预处理的实验效果,将算法直接实现在目标图像和经过预处理之后的目标图像上。由于实验环境的限制,缺少用于量化重建效果的场景真实三维深度信息,这也是本文下一步努力的方向之一。实验结果表明,该系统可以有效减小散斑噪声点对重建结果的影响,原因在于截断函数通过将高亮度的像素截断成固定的阈值,在保留了该像素亮度信息的同时减小了这一像素点在相关函数的计算中对周围像素的影响;而自适应滤波算法则通过判别背景点并将背景点亮度信息归零,突出了散斑的亮度信息。两者的组合既保证了散斑可以作为估计像素视差的有效标准,又减少了噪声点对临近像素深度值计算的干扰。

实验结果如图5所示。图5是对学生寝室场景的三维重建结果,图中深度信息用灰度值表示,亮度越低表示距离越近。重建对象是卧室内的床铺。图5(a)是有预处理过程的三维重建结果,图5(b)是没有预处理过程的三维重建结果。没有预处理过程的三维重建结果中有很多噪声点,有的地方还会因出现高亮度散斑而影响一整块方形区域内的三维重建效果的情况。而经过预处理的三维重建结果则显得比较自然,而且噪声点比较少。



(a)有预处理过程的三维重建效果图 (b)无预处理过程的三维重建效果图

图5 预处理对三维重建结果的影响

4.2 三维重建精度

通过实验测得的对于平面的重建精度如表2所列。

表2 三维平面重建精度

平面深度(mm)	理论最大误差(mm)	实际最大误差(mm)
956	7.5	3
1498	14	6
1918	29	11
2021	31	12

实验选取平直的墙面作为平面,将固定好的镜头与散斑发射器垂直面对墙壁进行三维重建。真实距离的测量使用了激光测距仪(精确到1mm)。理论最大误差指的是整像素点视差间隔所对应的深度分辨率。

从重建实验可以得出结论:对于平面,由于采取了合适的次像素点插值策略,实际重建误差小于理论的最大误差。三维重建相对精度在1%以内,已经足够满足实际应用的要求。

4.3 三维重建速度

将算法分别在GPU和CPU上实现,通过实验验证实时三维重建的可行性与本文提出的GPU计算框架的高效性。表3列出了GPU计算的每一步所花费的时间,实验对象是1024×768分辨率、视差范围为80的三维重建结果。表4比较了在不同分辨率下CPU与GPU的计算效率。

表3 GPU分步计算时间

步骤	耗时时间(ms)
预处理	4
核函数一	15.5
核函数二	15
核函数三	16.4
总时间	51

表4 GPU与CPU计算效率对比

分辨率	视差范围	CPU	GPU	加速比
640×480	37	372	11.4	32.6
1024×768	80	1992	51	39.1

GPU实验平台为NVIDIA GT 650M图形处理器,CPU实验平台为Intel Core i7 CPU。

本文比较了CPU与GPU计算效率之间的差别,当采用不同分辨率时,由于视差范围的不同,GPU相对于CPU的加速比也有变化。当视差范围更大时,数据在GPU线程内的重复利用次数更多,GPU的计算效率也更高。在1024×768分辨率下,GPU相对于CPU的加速比可以达到39.1,每一帧的重建时间为51ms。

结束语 本文利用GPU来加速计算基于散斑相关算法的三维重建过程,可以达到在高分辨率(1024×768)下的实时场景三维信息获取能力。与在CPU上实现同类算法相比,本文提出的GPU计算框架在普通笔记本的GPU(NVIDIA GT 650M)平台上可以取得相比于CPU(Intel i7 3630m)平台39倍的加速效果。实验证明,本文提出的基于GPU的散斑三维重建方案可以达到高分辨率(1024×768)高精度的实时三维重建效果。这种GPU解决方案可以推广到其他例如立体视觉等使用到ZNCC相关函数的图像处理系统中。在本文提供的三维重建解决方案下,GPU满负荷工作而CPU是空闲的,在下一步的工作中,准备使用GPU和CPU的异构计算方法,用GPU计算出每个点在不同视差下的匹配代价,并找出最优匹配位置和匹配,利用CPU优化误匹配点的精度,来进一步优化三维重建效果。另一方面,建立具有可靠置信度的散斑图像数据库来量化三维重建效果的好坏也是下一步努力的方向。

参考文献

- [1] Schaffer M, Kowarschik G M. High-speed pattern projection for three-dimensional shape measurement using laser speckles[J]. Applied optics, 2010, 49(10): 3622-3629
- [2] Barrientos B, Cerca M, Garcia-Ma R J. Three-dimensional displacement fields measured in a deforming granular-media surface by combined fringe projection and speckle photography[J]. Journal of Optics A: Pure and Applied Optics, 2008, 10(10): 104027
- [3] Yang Rui-gang, Pollefeys M. Multi-resolution real-time stereo on commodity graphics hardware[C]//IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Madison, 2003: 211-217
- [4] Yang R G, Pollefeys M, Li S F. Improved real-time stereo on commodity graphics hardware[C]//IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop. Washington, 2004: 36-36
- [5] Hirschmuller H, Scharstein D. Evaluation of stereo matching costs on images with radiometric differences[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(9): 1582-1599
- [6] Chen F, Brownm G, Song M. Overview of three-dimensional shape measurement using optical methods[J]. Optical Engineering, 2000, 39(1): 10-22
- [7] Batlle J, Salvi M E. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey[J]. Pattern recognition, 1998, 31(7): 963-982
- [8] Stockman H G. 3-D surface solution using structured light and constraint propagation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, 11(4): 390-402
- [9] Smisek J, Jancosek M, Pajdla T. 3D with Kinect[M]. German: Springer, 2013: 3-25
- [10] Wang G, Yin X, Pei X. Depth estimation for speckle projection system using progressive reliable points growing matching[J]. Applied Optics, 2013, 52(3): 516-524
- [11] Gong M, Yang R, Wang L. A performance study on different cost aggregation approaches used in real-time stereo matching [J]. International Journal of Computer Vision, 2007, 75(2): 283-296
- [12] Arieli Y, Freedman B, Machline M. Depth mapping using projected patterns; U. S. Patent 8150142[P]. 2012-4-3
- [13] Faugeras O, Hotz B, Mathieu H. Real time correlation-based stereo: algorithm, implementations and applications[R]. INRIA, 1993
- [14] Bay H, Tuytelaars T, Gool V. Surf: Speeded up robust features [M]. German: Springer Berlin Heidelberg, 2006: 404-417
- [15] Hallers I, Nedeveschi S. Design of interpolation functions for sub-pixel-accuracy stereo-vision systems[J]. IEEE Transactions on Image Processing, 2012, 21(2): 889-898
- [16] 王志国, 王贵锦, 施陈博. 积分图像的快速GPU计算[J]. 计算机应用研究, 2011, 28(10): 3913-3916
Wang Zhi-guo, Wang Gui-jin, Shi Chen-bo, et al. Fast Integral Image Computation on GPU[J]. Journal of Computer Applications, 2011, 28(10): 3913-3916

- [17] Harris M, Senguptaj S, Owens J D. Parallel prefix sum(scan) with CUDA[J]. GPU gems, 2007, 39(3):851-876
- [18] NVIDIA. NVIDIA C best practices guide[OL]. http://clocos.nvidia.com/cuda/pdf/CUDA_C_Best_Practices_Guide.pdf
- [19] Mei X, Sun X, Zhou M. On building an accurate stereo matching system on graphics hardware[C]// 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). 2011:467-474
- [20] Bilgic B, Horn B K P, Masaki I. Efficient integral image computation[C]// Intelligent Vehicles Symposium(IV) on the GPU. 2010 IEEE, 2010:528-533
- [21] NVIDIA. Nvidia cuda compute unified device architecture programming guide[OL]. http://moss.csc.ncsu.edu/~mueller/cluster/nvidia/2.0/Programming_Guide_2.0Beta2.pdf
- [22] Ortega J S. Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach[D]. Toulouse: Institut National Polytechnique de Toulouse-INPT, 2007
- [23] 汤颖, 肖廷哲, 范菁. 基于 GPU 加速的快速图像相似区域查找[J]. 计算机科学, 2014, 41(2):290-296
Tang Ying, Xiao Ting-zhe, Fan Jing. GPU-based Fast Search of Similar Patches in Images[J]. Computer Science, 2014, 41(2):290-296
- [24] 党建武, 杭利华, 王阳萍, 等. 基于 GPU 的 2D-3D 医学图像配准[J]. 计算机科学, 2013, 40(4):306-309
Dang Jian-wu, Hang Li-hua, Wang Yang-ping, et al. 2D-3D Medical Image Registration Based on GPU[J]. Computer Science, 2013, 40(4):306-309

(上接第 287 页)

- [12] 周仕飞. 3G 与 WiFi 中多媒体传输策略研究[D]. 长沙: 中南大学, 2010
Zhou Shi-fei. The research of multimedia transmission strategy of 3G and WiFi [D]. Changsha: Central South University, 2010
- [13] 章翠芝. 基于 WiFi 的时钟同步技术研究[D]. 杭州: 浙江大学, 2012
Zhang Cui-zhi. Research on clock synchronization technologies based on WiFi [D]. Hangzhou: Zhejiang University, 2012
- [14] 罗娜. 基于 88W8686 手持终端的 WiFi 功能的设计与实现[D]. 武汉: 武汉理工大学, 2010
Luo Na. Design and implementation of WiFi function based on 88W8686 hand-held terminal [D]. Wuhan: Wuhan University of Technology, 2010
- [15] 赵峙岳. 基于 WiFi 无线网络的嵌入式打印终端的设计[D]. 广州: 华南理工大学, 2011
Zhao Zhi-yue. The design of embedded printing terminal based on WiFi wireless network [D]. Guangzhou: South China University of Technology, 2011
- [16] 崔小东. 基于 WiFi 的无线校园网建设研究[D]. 南京: 南京理工大学, 2010
Cui Xiao-dong. Research on wireless campus network construction based on WiFi [D]. Nanjing: Nanjing University of Science and Technology, 2010
- [17] 曾强. WiFi 无线传感器网络的设计与实现[D]. 太原: 中北大学, 2012
Zeng Qiang. Design and implementation of WIFI wireless sensor network [D]. Taiyuan: North University of China, 2012
- [18] 李飞. WiFi 网络中基于测距的定位算法研究[D]. 成都: 电子科技大学, 2013
Li Fei. Research on localization algorithm based on WiFi network location [D]. Chengdu: University of Electronic Science and Technology of China, 2013
- [19] 王剑飞. 基于 WiFi 技术的井下视频传输系统的设计与实现[D]. 鞍山: 辽宁科技大学, 2014
Wang Jian-fei. Design and Realization of downhole video transmission system based on WiFi Technology [D]. Anshan: University of Science and Technology Liaoning, 2014
- [20] 刘艳虹. 3G 和 WiFi 双模终端的无缝切换算法研究[D]. 长春: 长春科技大学, 2013
Liu Yan-hong. Research on Seamless Handover Algorithm in 3G and WiFi dual mode terminal [D]. Changchun: Changchun University of Science and Technology, 2013
- [21] 刘标. 基于无线局域网的多信道快速接入算法研究[D]. 西安: 西安电子科技大学, 2012
Liu Biao. Research on multi channel fast access algorithm based on wireless LAN [D]. Xi'an: Xidian University, 2012
- [22] 温三宝. 密集覆盖下无线局域网的动态资源分配研究[D]. 西安: 西安电子科技大学, 2012
Wen San-bao. Study on dynamic resource allocation strategies in high density WLAN [D]. Xi'an: Xidian University, 2012
- [23] Wang G, Yang KH. A new approach to sensor node localization using RSS measurements in wireless sensor networks[J]. IEEE Trans. on Wireless Communications, 2011, 10(5):1389-1395
- [24] 叶苗, 王宇平. 基于变方差概率模型和进化计算的 WSN 定位算法[J]. 软件学报, 2013, 24(4):859-872
Ye Miao, Wang Yu-ping. Location Estimation in Wireless Sensor Networks Based on Probabilistic Model with Variant Variance and Evolutionary Algorithm[J]. Journal of Software, 2013, 24(4):859-872
- [25] Gholami M R, Vaghefi R M. RSS-Based Sensor Localization in the Presence of Unknown Channel Parameters[J]. IEEE Transactions on Signal Process, 2012, 61(15):3752-3759
- [26] Wang W D, Zhu Q X. Sequential Monte Carlo location in mobile sensor networks[J]. Wireless Network, 2009, 15:481-495
- [27] Amundson I, Sallai J, Koutsoukos X, et al. Mobile sensor waypoint navigation via RF-based angle of arrival location[J]. International Journal of Distributed Sensor Networks, 2012, 5:1-15
- [28] Baggio A, Langendoen K. Monte-Carlo localization for mobile wireless sensor networks[J]. Ad hoc Networks, 2008, 6:718-733
- [29] Wang J, Han T. A self-adapting dynamic location algorithm for mobile nodes in wireless sensor networks[J]. Procedia Environmental Sciences, 2011, 11:270-274
- [30] Hu G, Zhang R, Liu Y J. Research of location problem for mobile node in underwater sensor network[J]. Chinese Journal of Sensors and Actuators, 2012, 2(25):265-270