简化球谐近似模型的图形处理器加速求解

贺小伟,陈　政,侯榆青,郭红波

(西北大学 信息科学与技术学院,陕西 西安 710127)

摘　要: 作为辐射传输方程的高阶近似,简化球谐近似模型成为近年光学分子成像研究的重点,但计算效率低限制了它的广泛应用,为此提出一种基于图形处理器的并行加速策略,采用 NVIDIA 公司推出的统一计算设备架构,对求解过程中耗时最多的两个模块——有限元刚度矩阵的生成和线性方程组的求解进行基于图形处理器的并行加速;根据统一计算设备架构的特点,进行计算任务的分配、存储器的合理使用以及数据的预处理三方面的优化;仿体及数字鼠仿真实验对比刚度矩阵生成时间以及平均迭代时间,以评价所提出方法的加速效果。实验结果表明,该方法可使求解速度提高 30 倍左右,展示了该方法在光学分子成像中的优势及潜力。

关键词: 简化球谐近似模型;　有限元法;　统一计算设备架构;　并行计算

中图分类号: TP391　　**文献标志码:** A　　**DOI:** 10.3788/IRLA201645.0624002

Graphics processing units-accelerated solving for simplify spherical harmonic approximation model

He Xiaowei, Chen Zheng, Hou Yuqing, Guo Hongbo

(College of Information Science and Technology, Northwest University, Xi′an 710127, China)

Abstract: As a high-order approximation model to Radiative Transfer Equation, simplify spherical harmonic (SPN) approximation has become a hot research topic in optical molecular imaging research. However, low computational efficiency imposes restrictions on its wide applications. This paper presented a graphics processing units (GPU)-parallel accelerated strategy for solving SPN model. The proposed strategy adopted compute unified device architecture (CUDA) parallel processing architecture introduced by NVIDIA Company to build parallel acceleration of two most time-consuming modules, generation of stiffness matrix and solving linear equations. Based on the feature of CUDA, the strategy optimized the parallel computing in tasks distribution, use of memory units and data preprocessing. Simulations on phantom and digital mouse model are designed to evaluate the accelerating effect by comparing the time for system matrix generation and average time of each step iteration. Experimental results show that the overall speedup ratio is around 30 times, which exhibit the advantage and potential of the proposed strategy in optical molecular imaging.

Key words: simplify spherical harmonic approximation model;　finite element method;
compute unified device architecture;　parallel computing

收稿日期: 2015-10-08;　**修订日期:** 2015-11-08

基金项目: 国家自然科学基金(61372046);陕西省科技计划项目(2012KJXX-29, 2013K12-20-12)

作者简介: 贺小伟(1977-),男,副教授,博士生导师,主要从事医学图像处理及可视化方面的研究工作。Email:hexw@nwu.edu.cn

0 引 言

自 1999 年，美国哈佛大学 Weissleder 等人首次提出了分子影像学(Molecular Imaging,MI)概念以来[1],分子影像技术由于能够在细胞和分子水平观测疾病的发生和发展,得到研究人员的广泛关注,出现了基于核素、磁共振和光学等技术的分子成像模态。由于光学分子成像具有安全、无创、低成本、高特异性、探针制备储存方便等优点,成为现今分子影像技术的研究热点[2]。

准确描述光在生物组织中传输的数学模型是研究光学分子成像的核心问题之一，其中辐射传输方程(Radiative Tansfer Equation, RTE)的准确性被普遍认可。但其作为一个复杂的积分-微分方程，求解 RTE 的时间代价非常大[3-4]。作为 RTE 的高阶近似,简化球谐波(Simplified Spherical Harmonics Equations, SPN) 模型由于求解精度较高及求解代价相对较低成为近几年光学分子成像研究的重点[5-8]。

目前 SPN 的求解方法以有限元为代表的数值解法为主。由于有限元法需要构建系统矩阵,尤其对于精密网格问题的求解,其计算代价依然不容忽视。20 世纪 90 年代以来,计算机硬件进入了高速发展期。在人们致力于发展高速 CPU 的同时，图形处理器(Graphic Processing Unit, GPU) 由于其设计初衷不同，使其在浮点计算能力方面的发展远超同期 CPU。目前 GPU 浮点计算能力是同期 CPU 的 10 倍以上,带宽也是同期 CPU 的五倍以上[9]。基于以上原因,越来越多的科研人员将 GPU 应用于科学计算上[10-12]。

2007 年 NVIDIA 发布计算统一设备架构 (Compute Unified Device Architecture, CUDA)以来,越来越多的科研人员应用 CUDA C 将传统的串行应用程序改编为能在 GPU 上运行的并行程序。孟晓林等人实现了基于 GPU 对三维医学图像的快速分割[13]。Bolz 等人采用共轭梯度法完成了基于 GPU 的有限元的稀疏方程组求解问题[14]。王鑫等人利用 GPU 实现了荧光分子断层成像(FMT)的加速重建,加速比最高可达 34.5 倍[15]。彭宽等人利用 GPU 实现了对 RTE 求解的加速[16]。由于基于 GPU 的通用计算起步不久,随着人们对 GPU 架构的进一步研究,基于 GPU 的通用计算还有巨大的发展潜力。文中将采用基于 GPU 的方法对光学分子影像学中的前向问题进行加速求解。

1 方 法

此节简要介绍基于 GPU 的加速求解策略,重点介绍基于 CUDA 框架特点的并行计算优化,最后给出基于 CPU/GPU 混合平台的求解框架。

1.1 基于 GPU 的加速求解策略

使用有限元法求解 SPN 问题的时间复杂度可由表 1 表示，其中 n 为有限元节点数目与问题的规模相关。

表 1 有限元法的时间复杂度

Tab.1 Time complexity of the finite element method

Step	Content	Time complexity
(1)	Read mesh data	$O(n)$
(2)	Calculate stiffness matrix	$O(n^3)$
(3)	Calculate source power	$O(n^2)$
(4)	Solve linear equations with Jacobi iteration	$O(n^3)$
(5)	Output results	$O(1)$

由表 1 可以看出,在基于有限元法求解 SPN 模型过程中，时间代价最大两部分为刚度矩阵的生成及线性方程组的求解，文中主要实现这两部分的并行加速。

SPN 核心思路是将一维球谐近似映射到多维上。以 SP3 方程为例,如公式(1)所示:

$$-\nabla\frac{1}{3\mu_{a1}}\nabla\varphi_1+\mu_a\varphi_1=S+(\frac{2}{3}\mu_a)\varphi_2 \tag{1}$$

$$-\nabla\frac{1}{7\mu_{a3}}\nabla\varphi_2+(\frac{4}{9}\mu_a+\frac{5}{9}\mu_{a2})\varphi_2=-\frac{2}{3}S+(\frac{2}{3}\mu_a)\varphi_1$$

式中:S 为光源函数；$\mu_{ai}=\mu_a+\mu_s(1-g^i)$,其中 μ_a 为吸收系数，μ_s 代表散射系数，g 为组织间的各项异性因子；φ_1 为辐射度的勒让德矩 ϕ_l(Legendre Moments)的线性组合,由下式给出:

$$\varphi_1=\phi_0+2\phi_2 \tag{2}$$

$$\varphi_2=3\phi_2$$

根据伽辽金(Galerkin)方法,首先在公式(1)两边乘上函数 $\psi(r)$,并在整个区域 Ω 上积分。根据积分变化和边界条件,有限元离散网格中每个单元的形状函数可由区域 Ω 中的总光场密度 Φ 离散而成,如公式(3)所示:

$$\Phi(r)\approx\Phi^{h}(r)=\sum_{l=1}^{N}\left[\sum_{i=1}^{N^{(l)}}\phi_{i}^{(l)}\psi^{(l)}(r)\right] \tag{3}$$

将每个有限元单元的形状函数代入公式(1)可得到所有单元刚度矩阵,将所有有限元单元刚度矩阵叠加和拼接后即可得到总刚度矩阵。可以看出,若模型的离散网格由 N 个节点组成,则需要计算 N 次单元刚度矩阵,并完成 N 个单元刚度矩阵的叠加和拼接,在 CPU 串行架构中这无疑非常耗时。不难发现,每个单元刚度矩阵的计算之间相互独立,属于天然可并行特性。因此可根据 CUDA 的线程结构将每个单元形状函数的计算任务分配给对应的线程,并由线程间并行完成单元刚度矩阵计算,以达到并行加速的目的。

对于有限元方程组的求解,传统的高斯消元法受限于舍入误差和病态性,相比之下,迭代法更适合求解大规模线性方程组。其中,Jacobi 迭代需要保存上一步迭代的所有点,可以通过合适的程序设计做到以空间换时间,使其并行化以大大减少计算代价。研究 Jacobi 迭代中循环部分可以发现,该方法每次迭代时需要计算 k 个未知量(k 为刚度矩阵的阶数),且 k 个未知量只与上一步迭代的结果有关,因此在每一步迭代中可将 k 个未知量的计算分配给对应的线程,将串行的迭代改写为基于 GPU 的并行迭代。

1.2 基于 CUDA 架构特点的并行优化

上述 GPU 并行移植过程,实际是将原本串行计算映射到 GPU 上,利用线程间的高度并行性完成任务。在实际操作过程中,由于 GPU 硬件性能及 CUDA 架构特点等因素,往往会造成加速效果并不理想。下面给出三种优化思路。

(1) 稀疏矩阵存储格式的优化

对于有限元法,总刚度矩阵中非零元素的数量远小于矩阵元素的数量,即为典型的稀疏矩阵。但不同种类的稀疏矩阵拥有不同的存储格式,而不同存储格式之间的优劣并没有广泛认可的最优标准,常用的存储格式有对角线存储法、坐标存储法、Ellpack-Itpack 存储法以及 CSR (Compressed Sparse Row)存储法。对于在有限元法中得到的稀疏刚度矩阵,由于其为对称矩阵,目前研究中应用较为广泛的 CSR 存储法存储对称稀疏矩阵时可通过只存储上三角矩阵或下三角矩阵的方法减少存储量。文中将采用 CSR 存储法对稀疏刚度矩阵进行预处理以减少所占用的存储空间。

(2) 程序结构的优化

Jacobi 迭代法本质是通过循环使两次迭代结果相差在一定的误差范围内(误差阈值一般为 $c\leqslant1.0\times10^{-8}$),从而使迭代结果逼近最优解;总刚度矩阵的生成也需要进行大量循环操作,在执行任务时要处理大量分支判断操作。从微观架构上来说 GPU 并不适合处理如指令调度、判断、循环等任务,而 CPU 更擅长处理这些任务。因此在程序结构设计时,应将如循环操作等条件判断任务分配给 CPU 端,而 GPU 端则负责处理大规模数据任务,使分支判断对加速效果的影响降低。

(3) 存储器及数据传输的优化

在 CUDA 架构中,存储器的访问延迟是影响计算效率的一个重要因素,如何有效缓解 GPU 中显存和计算能力之间的不平衡是有效提高计算效率的关键。由于 GPU 中不同存储器在结构关系、空间大小及访问速度上有较大差别,因此根据刚度矩阵生成以及 Jacobi 迭代法的特点来分配数据及存储器的关系是提高效率一大要点。

在 CUDA 架构中全局存储器存储容量最大,但访问延迟最高;而纹理存储器、缓存以及寄存器几乎没有访问延迟,但存储容量很小。因此在程序优化中应主要考虑如何降低全局存储器的访问延迟以及灵活分配纹理存储器、缓存以及寄存器。

1.3 并行优化的具体实现策略

1.3.1 稀疏矩阵存储的预处理

(1) CSR 存储法

CSR 存储法也称为行格式存储法,该方法的基本思路是对稀疏矩阵进行逐行压缩存储。对于稀疏矩阵 A,若其中共有 h 个非零元素,则需要用三个向量 x、$x^{(j)}$和 $x^{(r)}$来存储该稀疏矩阵。其中 h 维向量 x

以先行后列的顺序依次存放稀疏矩阵 A 中的非零元素,非零元素的列号则依次存放在 h 维向量 $x(j)$中,同时用 k+1 维向量 $x(r)$来存放稀疏矩阵 A 中每一行第一个非零元素在向量 x 中的位置。

对于文中实验所用的稀疏刚度矩阵,使用 CSR 存储法存储后可大大减少其占用的存储空间,继而减少其在 CPU 端和 GPU 端之间的传输时间及对显存资源的占用,使其可以实现更大规模矩阵的计算。

(2) 针对 CUDA 特性改进的 CSR 存储法

在 CUDA 架构中,如果 16 个连续线程的访问地址是存放在连续的存储器片段内,则这 16 个连续线程对全局存储器的访问可以整合到一条数据交换指令中完成,而不满足要求部分对全局存储器的访问会转换为 16 次串行的访问,从而大大降低访问效率。在 CSR 存储法中,三个向量维数由稀疏矩阵阶数及非零元素个数决定,显然不满足该条件。为使其满足条件,对每个起始访问地址后不足 16 的部分进行补零处理,使其每一次访问都能由一条数据交换指令完成,具体方法如图 1 所示。

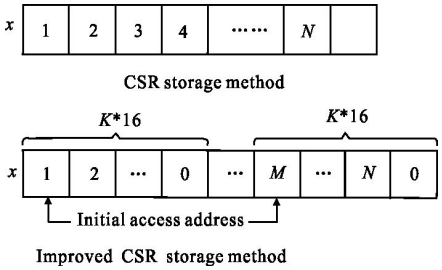

图 1 改进 CSR 存储法

Fig.1 Improved CSR storage method

可以看出,改进后的 CSR 存储法实际上增加了存储负担,但在不影响计算精度的情况下,该方法可以大幅降低对全局存储器的访问延迟,这是以较小存储负担换取计算效率的一个典型例子。

1.3.2 程序任务的分配

在 CUDA 程序中,如果循环判断位于内核函数中,GPU 执行过程受到循环条件判断的限制会增加其运行时间。若将该判断放置于内核函数外交由 CPU 处理,则可有效增加程序效率,见图 2。CUDA 程序中满足条件的循环判断语句都可通过合理的程序设计将其交由 CPU 处理,而 GPU 则负责处理大规模数据的并行计算。合理的任务分配可以显著降低程序执行时间。

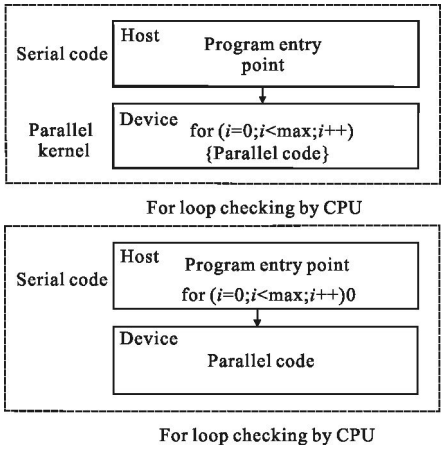

图 2 循环判断的分配

Fig.2 Allocation of loop checking

1.3.3 存储器的合理使用

在调用 CUDA 程序的过程中,CUDA 会根据线程中数据的大小自动分配存储器,但不同存储器之间差异非常大,如表 2 所示。

表 2 各种存储器的比较

Tab.2 Contrast of memory units

Memory	Location	Access permission	Variable life cycle
Register	GPU	Device RW	Same as thread
Local memory	Onboard video memory	Device RW	Same as thread
Shared memory	GPU	Device RW	Same as block
Constant memory	Onboard video memory	Device R Host RW	Stay until the process ends
Texture memory	Onboard video memory	Device R Host RW	stay until the process ends
Global memory	Onboard video memory	Device RW	Stay until the process ends
Host memory	Host memory	Host RW	Stay until the process ends
Pinned memory	Host memeory	Host RW	Stay until the process ends

CUDA 自动分配存储空间时只考虑数据大小及申请存储空间的顺序,根据问题特点自行选择存储器的使用则可以大大提高计算效率。表 2 中寄存器是线程内的高速缓存,每个寄存器大小为 32 bit。当寄存器被耗尽时,数据会被挤到本地存储器中,而本地存储器的访问速度远大于寄存器。在程序设计的过程中,应将需要多次调用的数据存放于寄存器中,

如刚度矩阵生成时的光场密度和 Jacobi 迭代中每一步的表面光信号分布 x_n；纹理存储器的访问速度与寄存器相当，由于其容量较小且在 GPU 中只可读取，应用来存放处理后的稀疏矩阵；模型的光学参数等常数则存放于常数存储器中。只有调用次数最少且所占空间最多的数据才应存放在全局存储器中。

1.4 整体加速策略

综上所述，基于 GPU 的加速求解主要策略为并行生成刚度矩阵及 Jacobi 迭代，针对 CUDA 架构特点优化后的求解流程如图 3 所示。

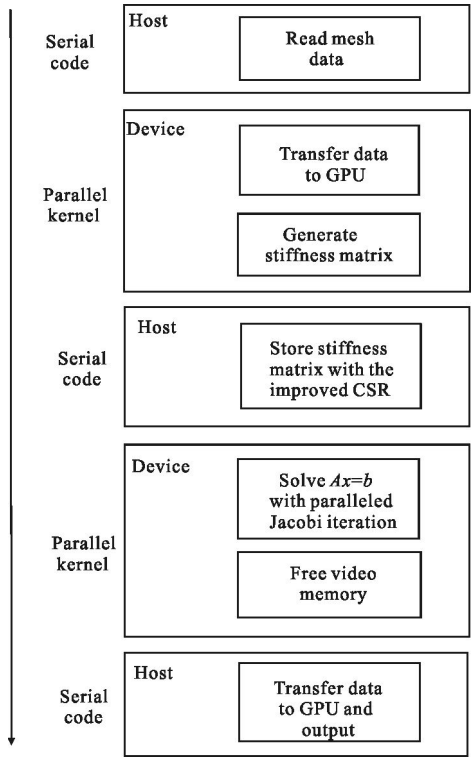

图 3 求解流程图

Fig.3 Flowchart of processing

2 实验与验证

为验证第二节所提出加速策略的有效性，设计了以下一系列实验。测试加速性能时，CPU 端运行的所有程序都使用单核串行方法，不采用多线程技术。对于刚度矩阵生成的加速测试，使用 T_{CPU}/T_{GPU} 评定加速比。对于线性方程组求解的加速测试，由于基于 GPU 的并行计算并不会对 Jacobi 迭代的收敛速度产生影响，且每次迭代的时间也有所不同，因此使用平均迭代时间(Average Time of Each Step,ATES)评定加速比。整体加速效果则使用不计仿真读取以及参数文件处理的整体计算时间评定加速比。

文中所有仿真实验所运行的平台是基于 Intel® Core™I7-3770K 处理器以及 NVIDIA® GeForce™ GTX 660 Ti显示卡的 PC。

2.1 匀质模型仿体实验

实验采用如图 4 的匀质仿体，其中圆柱体半径为 10 mm，高度为 30 mm，球形光源半径为 1 mm，其中心坐标为(-4,-4,15)，光源功率密度设为 1 nW/mm^3。吸收系数 μ_a=0.19 286 mm^{-1}，散射系数 μ_s=16.169 mm^{-1}，各向异性系数 g=0.90，介质折射率 n=1.37。

对图 4 中仿体进行网格剖分，得到有限元网格基本信息如表 3 所示。

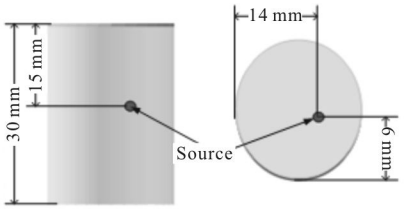

图 4 包含球型光源的圆柱仿体

Fig.4 Cylindrical phantom contains a sphere light source

表 3 四组圆柱方体离散网格

Tab.3 Four meshes for cylindrical phantom

No.	Number of nodes	Number of elements
1	9 046	50 113
2	19 479	102 541
3	25 762	151 369
4	32 304	198 677

对表 3 中四组有限元网格分别进行仿真实验，得到生成刚度矩阵的加速比如表 4 所示，Jacobi 迭代的平均迭代时间的加速比如表 5 所示。最终不同阶数 SPN 整体计算时间加速比如图 5 所示。

表 4 生成刚度矩阵的加速比

Tab.4 Speedup ratio of stiffness matrix generation

No.	SP1	SP3	SP5	SP7
1	7.75	9.87	12.61	14.13
2	9.44	13.63	13.09	15.82
3	10.92	14.96	16.15	18.06
4	12.13	16.67	18.96	20.37

表 5 Jacobi 迭代的 ATES

Tab.5 ATES of Jacobi iteration

Grid No.	SP1	SP3	SP5	SP7
1	5.01	8.93	11.34	13.36
2	8.69	14.47	18.15	19.94
3	12.06	22.17	27.29	29.18
4	15.13	30.37	34.87	36.25

从表 4 可以看出,采用 GPU 方法生成刚度矩阵的加速效果较为稳定，随着网格精密度上升及模型阶数提高，加速比上升有限。表 5 中 Jacobi 迭代 ATES 的加速比随着网格精密度上升及模型阶数提高上升更明显，由于每次迭代中各计算任务能够满足相互独立，且通过合理的程序设计使相互并行的线程越多越能掩盖存储器访问延迟。从图 5 可以看出实验中最高加速比为 29.3 倍。

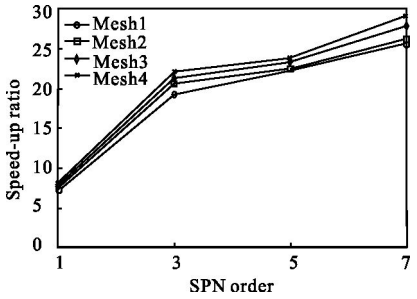

图 5 匀质仿体的加速性能对比

Fig.5 Acceleration performance comparison of homogeneous phantoms

2.2 数字鼠实验

在实际应用中,生物组织通常是复杂非匀质的,因此对此类模型的研究在生物发光成像研究中更有意义。对此采用如图 6 的数字鼠仿体,其中光源为半径为 0.5 mm,高度为 1 mm 的圆柱体,光源功率密度设为 1 nW/mm^3。对其进行有限元网格离散后得到如表 6 的四组网格信息，各主要器官的光学参数如表 7 所示。

对表 6 中四组网格分别进行数字鼠模型仿真实验，得到不同阶数 SPN 整体计算时间加速比如图 7 所示。

对比图 6 和图 7 可以发现,采用较为复杂的数字鼠模型，该方法的加速性能并没有快速下降,在图 7 中最高加速比为 28 倍。

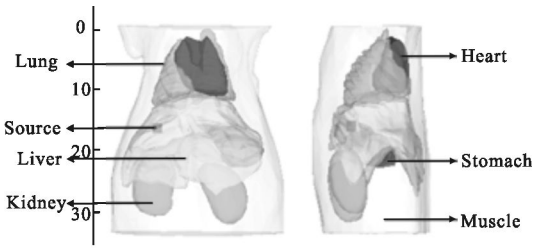

图 6 数字鼠的几何结构信息

Fig.6 Geometry information of digital mouse

表 6 数字鼠模型的四组离散网格

Tab.6 Four meshes for digital mouse model

No.	Number of nodes	Number of elements
1	7 145	37 652
2	12 953	70 558
3	18 609	109 713
4	22 364	151 869

表 7 数字鼠模型光学参数

Tab.7 Optical parameters of digital mouse

Organ	μ_a/mm^{-1}	μ_s/mm^{-1}	g
Muscle	0.005 7	0.237 4	0.9
Heart	0.091 0	1.029 1	0.85
Lung	0.304 5	2.227 3	0.94
Liver	0.545 8	0.711 5	0.9
Stomach	0.017 1	1.502 2	0.9
Kidney	0.102 1	2.414 4	0.9

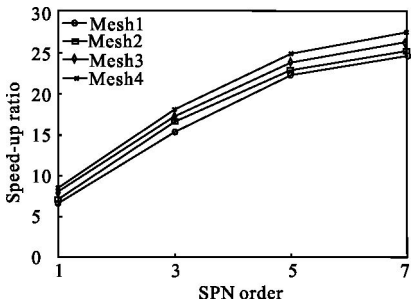

图 7 数字鼠模型的加速性能对比

Fig.7 Acceleration performance comparison of digital mouse model

3 结束语

文中实验生成刚度矩阵的加速比最高为 20.4 倍，Jacobi 平均迭代时间的加速比最高超过 36.2 倍，两种模型的总体加速比都达到接近 30 倍，对精密有限元网格及高阶 SPN 可以有效减少计算代价。该方法与传统试图减少迭代次数的加速方法不同，由于不受光学参数的影响，并且对于 Jacobi 迭代法的并行计算并不影响其收敛性质，使得任何基于 Jacobi 迭代法的优化算法都可以较容易与文中方法结合。通过两组实验可以看出，随着网格精密度上升及模型阶数提高，加速效果有效提高，因此随着 GPU 硬件性能的提升，该方法还有许多优化空间。基于 GPU 的加速策略相比于超级计算机及多 CPU 并行方法更为低廉的成本也是该方法的一大优势。

所提出的 GPU 方法需要进行两次数据传输，造成了耗费在低效率数据传输上的时间过多；同时 Jacobi 算法每次迭代之间无法并行加速降低了整体加速效果。如何有效减少数据传输次数及所需传输数据的大小是后续研究的一个重点，如何改进迭代算法使其能通过 GPU 加快收敛性也是进一步提高加速效果的关键。

参考文献：

[1] Weissleder R. Molecular imaging: exploring the next frontier [J]. ***Radiology***, 1999, 212(3): 609–614.

[2] Massoud T F. Molecular imaging in living subjects: seeing fundamental biological processes in a new light [J]. ***Genes & Development***, 2003, 17(5): 545–580.

[3] Ishimaru A. Diffusion of light in turbid material [J]. ***Applied Optics***, 1989, 28(12): 2210–2215.

[4] Liu Yongchuan, Song Enmin, Jin Renchao, et al. A tomographic reconstruction model for highly scattering media [J]. ***Infrared and Laser Engineering,*** 2014, 43(9): 3094–3098. (in Chinese)

[5] Klose A D, Larsen E W. Light transport in biological tissue based on the simplified spherical harmonics equations[J]. ***Journal of Computational Physics***, 2006, 220(1): 441–470.

[6] Liemert A, Kienle A. Analytical solutions of the simplified spherical harmonics equations [J]. ***Optics Letters***, 2010, 35 (20): 3507–3509.

[7] Klose AD, Poschinger T. Excitation–resolved fluorescence tomography with simplified spherical harmonics equations[J]. ***Physics in Medicine and Biology***, 2011, 56(5): 1443–1469.

[8] Guo Hongbo, Hou Yuqing, He Xiaowei, et al. Adaptive hp finite element method for fluorescence molecular tomography with simplified spherical harmonics approximation [J]. ***Journal of Innovative Optical Health Sciences***, 2014, 7(2): 342–345.

[9] NVIDIA. CUDA_C_Programming_Guide5.5 [R]. 2013,05.

[10] Chen Xi, Qiu Yuehong, Yi Hongwei. Parallel programming design of star image registration based on GPU [J]. ***Infrared and Laser Engineering***, 2014, 43(11): 3756–3761. (in Chinese)

[11] Li Dayu, Hu Lifa, Mu Quanquan, et al. Wavefront calculation of liquid crystal adaptive optics based on CUDA [J]. ***Optics and Precision Engineering***, 2010, 18(4): 848–854. (in Chinese)

[12] Wang Maozhi, Guo Ke, Xu Wenxi. Hyperspectral remote sensing image parallel processing based on cluster and GPU [J]. ***Infrared and Laser Engineering***, 2013, 42(11): 3070–3075. (in Chinese)

[13] Meng Xiaolin, Qin An, Xu Jian, et al. Fist 3D medical image segmentation based on CUDA [J]. ***Chinese Journal of Medical Physics***, 2010, 27(2): 1716–1720. (in Chinese)

[14] Bolz J, Farmer I, Grinspun E, et al. Sparse matrix solvers on the GPU:conjugate gradients and multigrid [J]. ***ACM Transactions on Graphics (TOG)***, 2003, 22(3): 917–924.

[15] Wang Xin, Zang Bin, Cao Xu, et al. Acceleration of early–photon fluorescence molecular tomography with graphics processing units [J]. ***Computational & Mathematical Methods in Medicine***, 2013, 9(1): 84–104.

[16] Peng Kuan, Gao Xinbo, Qu Xiaochao, et al. Graphics processing unit parallelaccelerated solution of the discrete ordinates for photon transport in biological tissues. [J]. ***Applied Optics***, 2011, 50(21): 3808–3823.