

内镜超声合成孔径成像算法的并行实现

李 激, 陈晓冬, 李嘉科, 汪 毅, 郁道银

天津大学精密仪器与光电子工程学院光电信息技术教育部重点实验室, 天津 300072

摘要 为了提高合成孔径成像算法在医学超声内镜系统中的计算效率, 提出一种在图形处理器(GPU)上并行实现的合成孔径成像方法。首先介绍了合成孔径算法的基本原理和图像重构过程; 然后对该算法进行并行化处理分析; 最后采用 CUDA 编程模式单指令多线程(SIMT)的灵活架构, 实现了基于 GPU 的内镜超声合成孔径成像算法。对多组散射点仿真成像实验进行对比分析, 并采用自行搭建的超声内镜实验系统对铁丝、肿囊假体及猪皮组织进行成像实验验证。实验结果表明, 所提方法在保证成像结果和成像质量不变的前提下, 大幅度提高了计算效率, 在计算数据规模为 1.47 GB($5305 \times 581 \times 64 \times 8$ byte)时, 获得了 50.93 倍的最大加速比。

关键词 光计算; 内镜超声成像; 并行实现; 合成孔径; 统一计算设备架构

中图分类号 TP391 **文献标识码** A

doi: 10.3788/LOP54.102001

Parallel Implementation of Synthetic Aperture Imaging Algorithm for Endoscopic Ultrasound

Li Wei, Chen Xiaodong, Li Jiake, Wang Yi, Yu Daoyin

Ministry of Education Key Laboratory of Optoelectronic Information Technology, College of Precision Instrument and Opto-Electronics Engineering, Tianjin University, Tianjin 300072, China

Abstract In order to improve the computational efficiency of synthetic aperture imaging algorithm in the medical endoscopic ultrasound system, the synthetic aperture imaging approach with parallel implementation on graphics processing unit (GPU) is proposed. Firstly, the basic principle and image reconstruction process of synthetic aperture algorithm are introduced. Then, the algorithm is analyzed in parallel processing. Finally, synthetic aperture imaging algorithm for endoscopic ultrasound based on GPU is implemented by using the flexible architecture of single instruction multiple threads (SIMT) of compute unified device architecture (CUDA) programming mode. Multiple simulation experiments of scattering points imaging are compared and analyzed, and the imaging experiment verifications of iron wire, cyst prosthesis and pigskin tissue are carried out by using a self-built endoscopic ultrasound experimental system. The experimental results show that the proposed method can greatly improve the computational efficiency while keeping the same imaging quality and results. When the calculated data size is 1.47 GB ($5305 \times 581 \times 64 \times 8$ byte), the maximum speedup ratio reaches 50.93.

Key words optics in computing; endoscopic ultrasound imaging; parallel implementation; synthetic aperture; compute unified device architecture

OCIS codes 200.4960; 110.1758; 100.2000

1 引言

医用超声内镜系统(EUS)是内镜和超声相结合的消化道检查技术, 其特点是将内镜观察和超声扫查的

收稿日期: 2017-03-23; **收到修改稿日期:** 2017-04-24

基金项目: 国家科技支撑计划(2012BAI19B02)

作者简介: 李 激(1992—), 男, 硕士研究生, 主要从事医学图像处理 and 并行计算方面的研究。

E-mail: lycanleeway@163.com

导师简介: 陈晓冬(1975—), 男, 博士, 教授, 主要从事光电成像技术及系统方面的研究。

E-mail: cxdchen@tju.edu.cn(通信联系人)

优势相结合,在使用内镜直接观察黏膜表面病变形态的同时,又可以对消化道器官进行超声扫描,获取消化道器官的组织学特征,扩大内镜和超声的诊断范围,提高诊断能力^[1-2]。

EUS中的超声扫查与传统的体外B超成像不同,由于超声换能器需要进入人体进行检测,因此对其尺寸有严格的要求。但是,换能器尺寸的减小使得阵元发射能量与信噪比(SNR)降低,所以很难保证成像质量。因此,为了实现较好的成像效果,必须对超声成像算法提出更高的要求。

在EUS中,延时和叠加算法(DAS)是应用最为广泛的超声成像算法,但是DAS却存在低SNR、低分辨率的缺点^[3]。为了提高图像质量,Jensen等^[4]将雷达领域的合成孔径技术应用到超声成像领域,提出了一种合成孔径超声成像算法(SA)。在SA中,通过求解多幅低分辨率图像(LRI),合成一幅高分辨率图像(HRI)。成像结果与DAS的相比,分辨率和SNR均得到明显的改善。但是,该算法存在大量复杂的重复运算,需要对每幅低分辨率图像中的每个像素点进行计算,因此非常耗时。

图形处理器(GPU)因具有高数据吞吐量、强大的并行计算以及浮点计算能力,已成为目前研究的热点之一^[5]。例如赵春晖等^[6]使用GPU实现了采用字典递归更新的目标检测稀疏算法;李路遥等^[7]实现了基于GPU的自适应模拟退火算法的光学模式转换技术;在机器学习、数据挖掘、语音识别等通用计算领域,GPU也得到了广泛的应用^[8]。NVIDIA公司提出统一计算设备架构(CUDA),利用CUDA平台提供的应用程序接口(API),编程人员可以实现程序的并行化处理^[9]。由此,本文提出一种在GPU上并行实现,适用于医学超声内镜系统的合成孔径成像方法,利用GPU的多线程处理能力,对成像的计算过程进行并行化处理,以提高算法的计算效率。

2 SA原理

2.1 算法概述

受限于延时和叠加算法的SNR低及成像分辨率低,Jensen等^[4]受到合成孔径雷达技术的启发,提出了SA,显著改善了图像质量。不同于传统的超声成像算法,SA使用单阵元发射一个球面波,将发射声场拓展至整个成像区域,改善了传统算法发射声场空间范围较窄的缺点^[10]。SA算法的基本成像原理如图1所示。单个阵元发射超声波后,所有阵元接收成像区域中每一个散射点的回波信息,通过延时叠加处理,得到一幅完整的低分辨率图像;将上述过程遍历所有发射阵元,把得到的所有低分辨率图像叠加合成,最终得到一幅高分辨率图像,即为最后的显示图像。

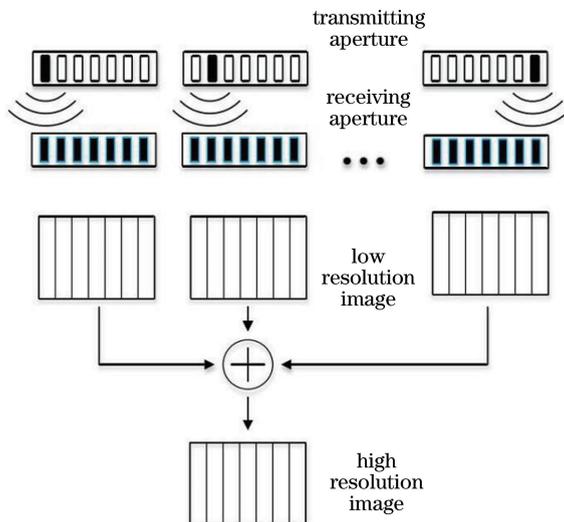


图1 SA算法基本成像原理

Fig. 1 Basic imaging principle of SA

2.2 图像重构

在医用超声内镜系统中,超声换能器阵元尺寸较小,因此单阵元发射的超声波可视为球面波^[10]。在介

质中声速 c 已知的情况下,对于任意一对发射阵元和接收阵元而言,从超声波发射到达空间中任一成像点,再到接收回波,整个过程的渡越时间可根据图 2 中的几何关系精确计算得出^[11]。上述过程的数学表达式为:

$$t_{TR}(M_p) = \frac{|M_p - T_p| + |R_p - M_p|}{c} \cdot f_s, \quad (1)$$

式中, T 和 R 分别代表的是发射阵元和接收阵元, T_p 表示发射阵元 T 的位置, R_p 表示接收阵元 R 的位置, M_p 表示成像点 M 的空间位置, f_s 是系统的采样频率。在发射阵元 T 不变的条件下,对所有接收阵元进行上述计算,可以实现回波数据的接收聚焦,合成一幅低分辨率图像,这一过程可以表示为:

$$I_{LRI_T} = \sum_{R=1}^r \omega_R \cdot X_R [t_{TR}(M_p)]. \quad (2)$$

求解所有发射阵元产生的低分辨率图像,把所有的低分辨率图像进行叠加合成,就得到了一幅所有成像点发射和接收都聚焦的高分辨率图像,即:

$$I_{HRI} = \sum_{T=1}^t \omega_T \cdot I_{LRI_T}, \quad (3)$$

式中, r 和 t 分别表示接收阵元和发射阵元的数目, $X_R(t)$ 表示回波信号, ω_R 是接收阵元 R 的权值函数, ω_T 是发射阵元 T 的权值函数。

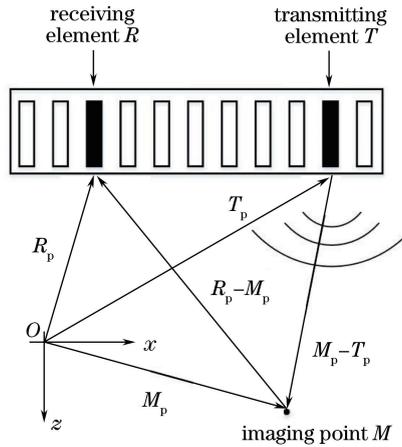


图 2 SA 算法图像重构的几何表示

Fig. 2 Geometric representation of the image reconstruction of SA

3 内镜超声合成孔径成像算法的并行实现

3.1 SA 算法并行化处理分析

由于回波数据量较大,运算过程较复杂,SA 算法的求解过程如果采取传统的串行计算模式,将会非常耗时,因此需要对 SA 算法进行并行化处理分析。

结合 2.2 节中图像重构的过程不难发现,回波信号的分析求解过程需要执行大量的重复操作。假设使用的医用超声内镜系统具有 64 个超声阵元,那么对于求解一幅图像尺寸为 $200 \text{ pixel} \times 100 \text{ pixel}$ 的低分辨率图像而言,就需要对图像中 20000 个像素点进行渡越时间的计算,通过延时叠加计算来实现 64 个阵元的接收聚焦,并从回波信号中筛选出成像点,构成一幅所需的低分辨率图像。通过分析可以发现,在上述具有 20000 个像素点的低分辨率图像中,各个点执行的计算操作都是相同的,求解计算过程互不影响,因此可以同时进行。也就是说,这一成像过程具有很好的并行性。

对于上述系统,需要求解 64 幅低分辨率图像,叠加合成一幅最终显示的高分辨率图像。叠加求和的过程,在数学上可以表述为一个加权求和的过程。求和的每一项相对于其他项而言,存在计算的无关性,即各项之间的计算相互独立,可以并行执行。

CUDA 作为 NVIDIA 提出的通用计算的 GPU 架构,能够实现单指令多线程的并行处理^[12]。在 CUDA

架构下,程序的最小执行单位是线程,线程包含于线程块中,线程块包含于线程格中。实现过程为:1) 使用 GPU 中读写速度较快,但存储空间较小的共享内存^[13]来存储 SA 算法中的权值函数 ω ;2) 使用 GPU 中的全局内存来存放输入的回波数据和处理后的数据;3) 对低分辨率图像以及高分辨率图像进行重构的过程中,分别进行多线程计算,实现 SA 算法的并行化处理。

3.2 加速算法具体实现流程

利用中央处理器(CPU)和 GPU 的 CUDA 联合编程模式,设计了两个核函数,并行实现了基于 GPU 的内镜超声合成孔径成像算法。所提出算法的主要实现流程如图 3 所示。首先,在显存中为回波数据分配存储空间,将主机端(CPU 端)的数据拷贝至设备端(GPU 端)。其次,利用关键字 `_shared_` 为权值函数开辟一块共享内存空间。然后在 GPU 端调用第一个核函数来求解低分辨率图像,这其中主要包含以下并行操作:1) 对回波数据进行噪声补偿,使回波数据更切合实际;2) 计算像素点和阵元的三维坐标,求解像素点的渡越时间;3) 进行延时叠加操作,实现像素点的接收聚焦;4) 进行成像点筛选,接着调用第二个核函数,对上述过程求解所得的低分辨率图像进行加权叠加合成。最后,将合成结果拷贝至 CPU,主机端对合成的高分辨率图像进行显示。

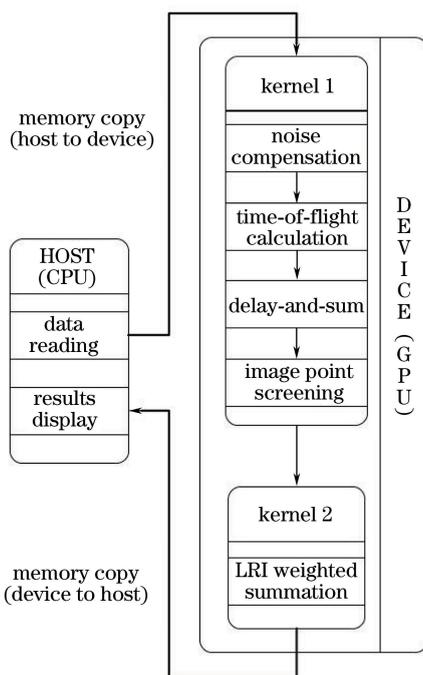


图 3 并行算法的主要流程

Fig. 3 Main flow of the parallel algorithm

4 超声内镜实验系统

4.1 整体结构

内镜超声系统采用 SA,所成图像清晰,SNR 和分辨率远高于其他常用算法。但是该算法复杂度高,计算量大。因此,使用 GPU 进行硬件加速,通过对计算过程并行化处理,达到算法加速的目的。系统采用的是相控阵超声换能器,结合相控阵超声内镜的成像扫描方向与电子内镜观察方向一致的特点,利用电子进行声束聚焦和扫描,采用 GPU 对回波数据进行并行处理加速,最终使系统能够在不转动换能器的情况下,实现对人体组织的快速扫描,使其适用于穿刺等临床治疗。

实验系统分为硬件电路和处理软件两大部分,其原理图如图 4 所示。其中,现场可编程门阵列(FPGA)是整个超声内镜实验系统硬件部分的控制核心;脉冲发射电路发射的激励脉冲通过多阵元选通电路激励超声换能器阵元来产生超声波,对物体进行合成孔径扫描;换能器接收超声回波信号,通过模数转换电路将得到的回波数据传输给 FPGA,并在双倍速率同步动态随机存储器(DDR SDRAM)中进行缓存;通过 USB3.0

接口将回波数据传输至计算机(PC),进行高质量超声回波图像的合成。处理软件在PC机上运行,其中并行处理算法在GPU中执行,程序通过混合编译的方式生成可执行文件,图形用户界面(GUI)软件通过调用该文件对成像过程实现加速。

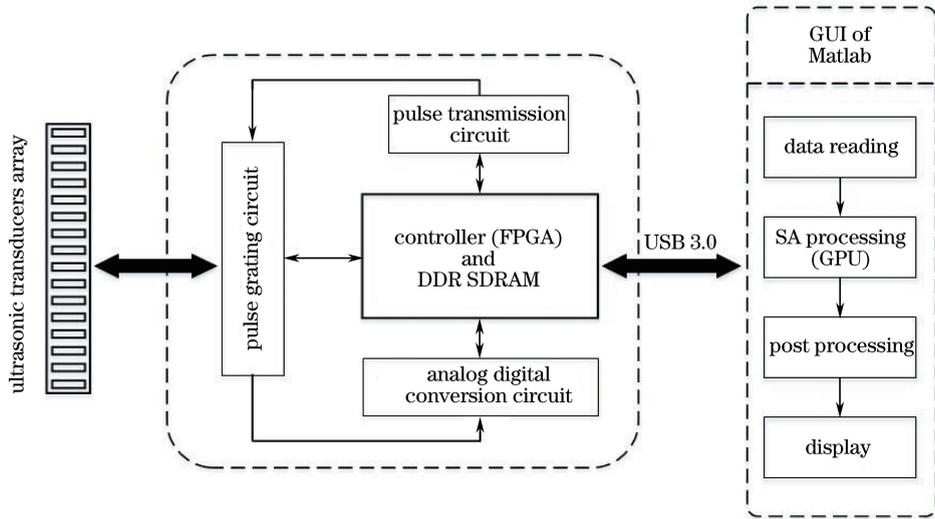


图4 超声内镜实验系统原理图

Fig. 4 Schematic diagram of endoscopic ultrasound experimental system

4.2 实验系统

实验搭建的实物系统如图5所示。系统外接15 V直流电压进行供电。该系统硬件控制核心使用的是Altera公司生产的Cyclone III系列的FPGA;换能器采用的是环阵相控阵超声换能器,阵元数为64;阵元中心频率为7 MHz;系统采样频率为180 MHz;阵元间距为0.145 mm;探头扫描角度为 $-30^{\circ} \sim 30^{\circ}$,对正前方深度5~50 mm的区域进行扇形扫描成像。

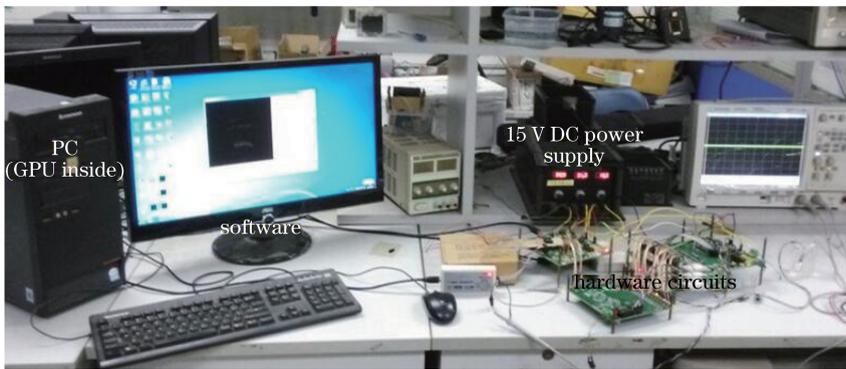


图5 超声内镜实验系统实物图

Fig. 5 Picture of endoscopic ultrasound experimental system

5 实验结果与分析

实验中所有的操作均是在Windows 7旗舰版32位操作系统下完成的。CPU为Intel(R) Core(TM) i3-2100,4核,主频为3.10 GHz,系统内存为4 GB;GPU为NVIDIA GeForce GT650,包含384个CUDA核心,主频为735 MHz,显卡内存为1 GB;CUDA驱动版本为8.0,Runtime版本为8.0。

5.1 仿真实验设计与结果分析

首先使用Field II^[14-15]软件进行仿真,建立医用超声内镜系统模型,该仿真系统的基本参数如表1所示,仿真系统参数与实际搭建的实验系统参数相符合。通过仿真实验得到理想的超声内镜系统的回波数据。

表 1 仿真系统基本参数

Table 1 Basic parameters of simulation system

Parameter	Value
Ultrasonic velocity in human body /($\text{m}\cdot\text{s}^{-1}$)	1540
Elements center frequency /MHz	7
Sampling frequency /MHz	180
Element width /mm	0.2400
Adjacent element spacing /mm	0.145

在实验中设置了 7 个成像散射点, 它们的空间位置分别为 $(0, 0, 25)$, $(0, 0, 30)$, $(0, 0, 35)$, $(0, 0, 40)$, $(0, 0, 45)$, $(0, 0, 50)$, $(0, 0, 55)$ (单位为 mm)。这些散射点在 25~55 mm 的轴向距离内, 以 5 mm 的间隔等距离分布。

设置医用超声内镜系统的阵元数为 64, 运算数据量为 $5305 \times 581 \times 64 \times 8$ byte, 成像算法在 CPU 平台上的执行结果如图 6(a) 所示。在 GPU 平台上设置每个线程块内开启的线程数 $V_{\text{threads}} = 128$, 所提出方法的执行结果如图 6(b) 所示。

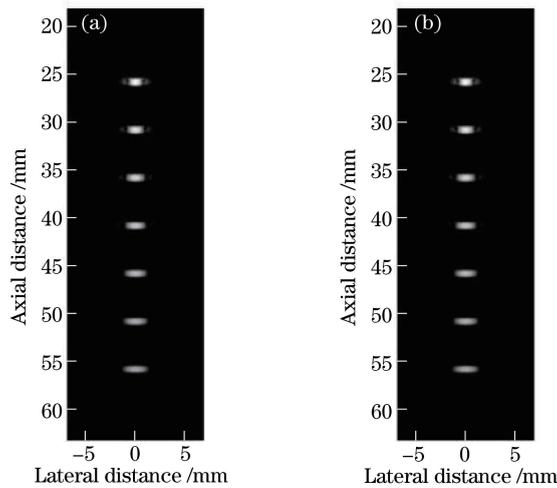


图 6 模拟散射点成像的 CPU 与 GPU 执行结果。(a) CPU 成像结果; (b) GPU 成像结果

Fig. 6 Implementation results of CPU and GPU of simulated scattering points imaging.

(a) CPU imaging result; (b) GPU imaging result

在上述实验的基础上, 改变系统阵元数目以及运算数据量, 进行 CPU 和 GPU 的程序执行耗时统计与加速比计算。以 20 次程序运行时间的均值作为每组实验的计时结果, 最终统计结果如表 2 所示。

表 2 CPU 和 GPU 上的程序耗时对比

Table 2 Comparison of program time-consuming of CPU and GPU

Data size /byte	CPU /s	GPU /s	Speedup ratio
$5305 \times 59 \times 32 \times 8$	53.66	1.30	41.28
$5305 \times 59 \times 64 \times 8$	198.82	4.46	44.58
$5305 \times 581 \times 32 \times 8$	516.33	10.86	47.54
$5305 \times 581 \times 64 \times 8$	2004.18	39.35	50.93

对上述实验的成像结果进行分析, 引入成像误差率和归一化相关系数这两个值进行成像结果相似度的判定。其中, 成像误差率表示两幅图中位置相同、像素值不相等的点数与总点数的比率, 其值越小表明, 两者相似度越高; 归一化相关系数表示两幅图的相似度, 取值在 0~1 之间, 越接近 1, 表明两者相似度越高。以上述模拟散射点的成像结果为例, 进行成像效果一致性评估, 具体的实验统计结果如表 3 所示。

结合上述实验统计结果对成像效果的一致性进行分析, 可以看出, 由于 GPU 对于双精度浮点数据的运算能力不及 CPU^[16], 因此造成计算结果上的差异。但是这种差异极小, 对于成像系统而言是可以忽略不计的, 并不影响最终的成像质量, 因此可以认为两者成像质量相同。

表3 CPU和GPU上的成像误差率和归一化相关系数

Table 3 Imaging error rate and normalized correlation coefficient of CPU and GPU

Data size /byte	Number of different points	Number of total points	Error rate /%	Normalized correlation coefficient
5305×59×32×8	1917	413×561×3	0.276	0.999998870619135
5305×59×64×8	816	413×561×3	0.117	0.999999989915709
5305×581×32×8	2193	413×561×3	0.316	0.999999727733754
5305×581×64×8	1212	413×561×3	0.174	0.999999677298417

实验结果表明,成像算法在CPU平台和GPU平台上具有近乎相同的成像效果,验证了在GPU平台上算法实现结果的准确性。在保证成像效果一致的前提下,所提出的方法耗时较短,加速效果明显。在数据规模为5305×581×32×8 byte时,加速比为47.5。而且随着数据规模的增大,加速比不断增加,验证了所提方法的高效性。

结合上述实验,改变 V_{threads} 取值,结合GT650的开普勒架构^[17],设置 V_{threads} 值分别为64,128,256,512,以20次程序运行时间的均值作为每组实验的计时结果,最终统计结果如表4、表5所示。

表4 V_{threads} 分别为64和128时程序运行时间的比较Table 4 Comparison of program running time when V_{threads} is 64 and 128, respectively

Data size /byte	$V_{\text{threads}} = 64$			$V_{\text{threads}} = 128$		
	CPU/s	GPU/s	Speedup ratio	CPU/s	GPU/s	Speedup ratio
5305×59×32×8	53.66	1.44	37.26	53.66	1.30	41.28
5305×59×64×8	198.82	4.76	41.77	198.82	4.46	44.58
5305×581×32×8	516.33	10.98	47.02	516.33	10.86	47.54
5305×581×64×8	2004.18	40.42	49.58	2004.18	39.35	50.93

表5 V_{threads} 分别为256和512时程序运行时间的比较Table 5 Comparison of program running time when V_{threads} is 256 and 512, respectively

Data size /byte	$V_{\text{threads}} = 256$			$V_{\text{threads}} = 512$		
	CPU /s	GPU /s	Speedup ratio	CPU /s	GPU /s	Speedup ratio
5305×59×32×8	53.66	1.37	39.17	53.66	1.64	32.72
5305×59×64×8	198.82	4.73	42.03	198.82	5.55	35.82
5305×581×32×8	516.33	11.19	46.14	516.33	13.69	37.72
5305×581×64×8	2004.18	40.19	49.87	2004.18	49.55	40.45

实验结果表明, V_{threads} 取值分别为64,128,256,512时,所提方法与CPU上的SA算法,均能起到加速效果。在 $V_{\text{threads}} = 128$ 时,所提方法加速效果最好,在数据规模为5305×581×64×8 byte时,可达50.93倍的最大加速比。

5.2 超声成像实验与结果分析

利用自行搭建的超声内镜实验系统对铁丝进行扫描成像。铁丝按照仿真实验的设置,以5 mm的间隔等距离固定放置在装满水的水槽中,铁丝直径约为0.6 mm。超声成像系统将回波数据上传给PC,在相同实验条件下,CPU平台与GPU平台的成像结果分别如图7(a)和图7(b)所示。

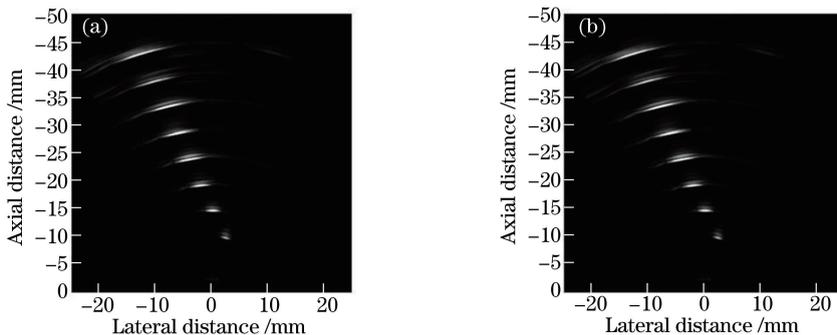


图7 铁丝扫描成像的CPU与GPU执行结果。(a) CPU成像结果;(b) GPU成像结果

Fig. 7 Implementation results of CPU and GPU of iron wire scanning imaging. (a) CPU imaging result; (b) GPU imaging result

在此基础上,对肿瘤假体以及猪皮组织进行了对比实验。成像实验基于 CPU 平台与 GPU 平台进行,实验结果如图 8、图 9 所示。

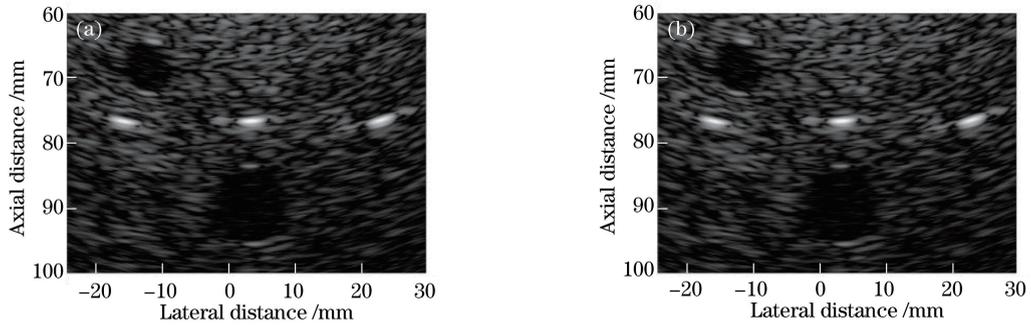


图 8 肿瘤假体成像的 CPU 与 GPU 执行结果。(a) CPU 成像结果;(b) GPU 成像结果

Fig. 8 Implementation results of CPU and GPU of cyst prosthesis imaging.

(a) CPU imaging result; (b) GPU imaging result

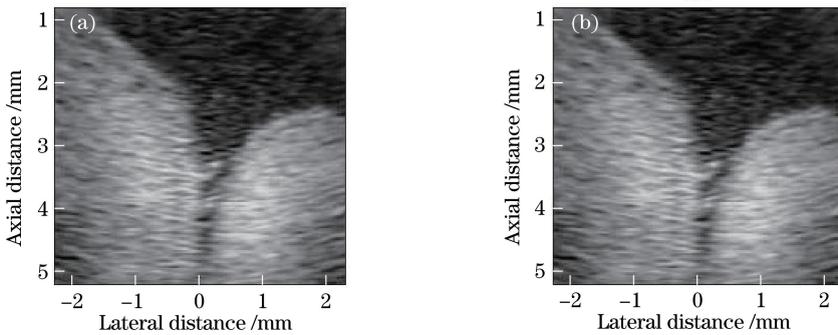


图 9 猪皮组织成像的 CPU 与 GPU 执行结果。(a) CPU 成像结果;(b) GPU 成像结果

Fig. 9 Implementation results of CPU and GPU of pigskin tissue imaging.

(a) CPU imaging result; (b) GPU imaging result

通过上述多组实验对比发现,成像算法在 CPU 平台与 GPU 平台上具有相同的成像效果,验证了基于 GPU 的超声内镜合成孔径成像方法和实验系统的可行性。

6 结 论

虽然 SA 成像质量高,但是存在计算耗时长的问题。因此,本文提出一种在 GPU 上并行实现的适用于医学超声内镜系统的合成孔径成像方法,利用 GPU 的多线程处理能力,实现高质量超声图像求解过程的并行化处理。对多组仿真实验进行对比分析,并采用自行搭建的超声内镜成像系统进行实验验证。结果表明,所提方法与基于 CPU 实现的方法相比,成像效果一致,加速效果明显,验证了 GPU 并行处理过程的准确性与高效性,具有重要的实际意义。后续的工作,可在此基础上利用 GPU 实现合成孔径连续声速合成算法(SASB)、多阵元合成发射孔径算法(MSA)的并行加速优化,以提高算法的计算效率。

参 考 文 献

- [1] Chen Xiaodong, Li Ming, Yu Feng, *et al.* Technology of digital endoscopic ultrasonic imaging[J]. Chinese J Lasers, 2009, 36(10): 2557-2560.
陈晓冬, 李明, 俞锋, 等. 超声内窥数字成像技术[J]. 中国激光, 2009, 36(10): 2557-2560.
- [2] Yu Daoyin, Li Ming, Li Yan, *et al.* Research on synthetic aperture technique for endoscopic ultrasound imaging[J]. Chinese J Lasers, 2010, 37(11): 2693-2697.
郁道银, 李明, 李妍, 等. 超声内窥合成孔径成像技术的研究[J]. 中国激光, 2010, 37(11): 2693-2697.
- [3] Tasinkevych Y, Trots I, Nowicki A, *et al.* Modified synthetic transmit aperture algorithm for ultrasound imaging[J].

- Ultrasonics, 2012, 52(2): 333-342.
- [4] Jensen J A, Nikolov S I, Gammelmark K L, *et al.* Synthetic aperture ultrasound imaging[J]. Ultrasonics, 2006, 44(s1): 5-15.
- [5] Zhang Zhilu, Cai Dongmei, Jia Peng, *et al.* Fast simulation for high precision atmospheric turbulence phase screen based on power spectrum[J]. Laser & Optoelectronics Progress, 2017, 54(2): 020101.
张智露, 蔡冬梅, 贾鹏, 等. 基于功率谱的高精度大气湍流相位屏的快速模拟[J]. 激光与光电子学进展, 2017, 54(2): 020101.
- [6] Zhao Chunhui, Yao Xifeng, Zhang Lili. Target detection sparse algorithm by recursive dictionary updating and GPU implementation[J]. Acta Optica Sinica, 2016, 36(8): 0828002.
赵春晖, 姚浙峰, 张丽丽. 采用字典递归更新的目标检测稀疏算法及 GPU 实现[J]. 光学学报, 2016, 36(8): 0828002.
- [7] Li Luyao, Yan Lianshan, Ye Jia, *et al.* Mode conversion technology based on adaptive simulated annealing algorithm[J]. Acta Optica Sinica, 2016, 36(12): 1206004.
李路遥, 闫连山, 叶佳, 等. 基于自适应模拟退火算法的光学模式转换技术[J]. 光学学报, 2016, 36(12): 1206004.
- [8] Ding Ke, Tan Ying. A review on general purpose computing on GPUs and its applications in computational intelligence[J]. CAAI Transactions on Intelligent Systems, 2015, 10(1): 1-11.
丁科, 谭莹. GPU 通用计算及其在计算智能领域的应用[J]. 智能系统学报, 2015, 10(1): 1-11.
- [9] Tao Ziqiang, Lü Liushuai, Wu Jian, *et al.* CUDA accelerate SRAD algorithm in medical ultrasonic image[J]. Chinese Journal of Medicinal Guide, 2015, 17(s1): 44-48.
陶自强, 吕留帅, 吴剑, 等. CUDA 加速医学超声图像 SRAD 去噪[J]. 中国医药导刊, 2015, 17(s1): 44-48.
- [10] Karaman M, Li P C, O'Donnell M. Synthetic aperture imaging for small scale systems[J]. IEEE Transactions on Ultrasonics Ferroelectrics & Frequency Control, 1995, 42(3): 429-442.
- [11] Zhang Chuang, Chen Xiaodong, Wang Yi, *et al.* Phased array imaging algorithm for endoscopic ultrasound based on synthetic aperture imaging technique[J]. Acta Optica Sinica, 2014, 34(12): 1211004.
张闯, 陈晓冬, 汪毅, 等. 基于合成孔径技术的内镜超声相控阵成像算法[J]. 光学学报, 2014, 34(12): 1211004.
- [12] Xiong Bangshu, Ye Yijia. Ou Qiaofeng, *et al.* Fast implementation method of protein spots detection based on CUDA[J]. Journal of Biomedical Engineering, 2016, 33(1): 83-88.
熊邦书, 叶毅嘉, 欧巧凤, 等. 基于 CUDA 的蛋白质点检测快速实现方法[J]. 生物医学工程学杂志, 2016, 33(1): 83-88.
- [13] Cheng Feng, Li Dehua. CUDA-based parallel implementation of the Adaboost algorithm[J]. Computer Engineering & Science, 2011, 33(2): 118-123.
程峰, 李德华. 基于 CUDA 的 Adaboost 算法并行实现[J]. 计算机工程与科学, 2011, 33(2): 118-123.
- [14] Jensen J A. FIELD: a program for simulating ultrasound systems[J]. Medical & Biological Engineering & Computing, 1999, 34: 351-353.
- [15] Jensen J A. Users' guide for the Field II program[M]. Denmark: Technical University of Denmark, 2001.
- [16] Jiang Xiaoyu, Cong Bin, Pei Chuang, *et al.* A new look-up table method of holographic algorithms based on compute unified device architecture parallel computing[J]. Acta Optica Sinica, 2015, 35(2): 0209001.
蒋晓瑜, 丛彬, 裴闯, 等. 一种基于新型查表方法的统一计算设备架构并行计算全息算法[J]. 光学学报, 2015, 35(2): 0209001.
- [17] Liu Jinshuo, Liu Tianxiao, Wu Hui, *et al.* From graphic processing unit to general purpose graphic processing unit[J]. Journal of Wuhan University (Natural Science Edition), 2013, 59(2): 198-206.
刘金硕, 刘天晓, 吴慧, 等. 从图形处理器到基于 GPU 的通用计算[J]. 武汉大学学报(理学版), 2013, 59(2): 198-206.