



(12)发明专利申请

(10)申请公布号 CN 108354626 A  
(43)申请公布日 2018.08.03

(21)申请号 201810278570.5

(22)申请日 2018.03.31

(71)申请人 华南理工大学

地址 510640 广东省广州市天河区五山路  
381号

(72)发明人 陈俊颖 陈锦辉 闵华清

(74)专利代理机构 广州粤高专利商标代理有限  
公司 44102

代理人 何淑珍

(51)Int.Cl.

A61B 8/00(2006.01)

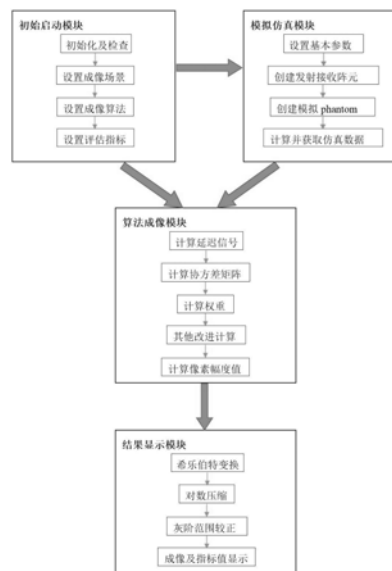
权利要求书1页 说明书10页 附图4页

(54)发明名称

基于GPU的多种MV高清算法快速医学超声影  
像系统

(57)摘要

本发明公开基于GPU的多种MV高清算法快速  
医学超声影像系统。本发明的系统包括初始启动  
模块、模拟仿真模块、算法成像模块和结果显示  
模块；其中，初始启动模块完成系统初始化设置  
后调用算法成像模块进行成像，模拟仿真模块作  
为一个辅助模块产生模拟数据供其他模块使用，  
算法成像模块是整个系统的核心模块，接收来自  
初始启动模块的参数设置与模块仿真模块或数  
据采集设备的数据进行成像计算，而结果显示模  
块则对其他模块的结果进行最后的处理与显示，  
最终实现多种MV算法的快速成像功能。本发明可  
在极短的时间内完成高清医学超声图像算法的  
复杂计算，使用的各MV高清算法的GPU编程实现  
方案可方便地部署于带有GPU的计算机上，可有  
效满足使用需求，实用性强。



1. 基于GPU的多种MV高清算法快速医学超声影像系统,其特征在于包括初始启动模块、模拟仿真模块、算法成像模块和结果显示模块;其中,初始启动模块完成系统初始化设置后调用算法成像模块进行成像,模拟仿真模块作为一个辅助模块产生模拟数据供其他模块使用,算法成像模块是整个系统的核心模块,接收来自初始启动模块的参数设置与模块仿真模块或数据采集设备的数据进行成像计算,而结果显示模块则对其他模块的结果进行最后的处理与显示,各个模块之间既独立工作又相互协作,最终实现多种MV算法的快速成像功能。

2. 根据权利要求1所述的基于GPU的多种MV高清算法快速医学超声影像系统,其特征在于初始启动模块主要进行系统启动时需要的初始化工作,包括初始化参数、设置成像场景、设置启用的算法、评估指标及图像显示参数,其中大多数初始化设置提供了与传统超声成像系统一致的选项,而对于成像算法的设置,本系统则提供了多种高清成像算法的选择,包括常规MV算法、迭代MV算法、融合对角加载的MV算法、前后向平滑MV算法、融合广义相干因子的MV算法、基于特征空间处理的MV算法和特征空间与符号相干系数融合的MV算法。

3. 根据权利要求1所述的基于GPU的多种MV高清算法快速医学超声影像系统,其特征在于模拟仿真模块采用Field II模拟超声成像过程取得仿真数据,首先模拟现实的超声成像设备设置需要的物理数据,创建发射和接收阵元,另外需根据实际应用场景需要创建虚拟的物体,然后,按扫描线逐条计算并收集回波数据。

4. 根据权利要求1所述的基于GPU的多种MV高清算法快速医学超声影像系统,其特征在于算法成像模块包含常规MV算法、迭代MV算法、融合对角加载的MV算法、前后向平滑MV算法、融合广义相干因子的MV算法、基于特征空间处理的MV算法和特征空间与符号相干系数融合的MV算法。

算法成像模块由初始模块进行调用启动,通过数据采集设备或模拟仿真模块获得成像所需的原始信号数据,然后根据MV算法进行像素幅度值计算,取得成像,并将成像结果交给结果显示模块以显示最终的超声图像。

5. 根据权利要求4所述的基于GPU的多种MV高清算法快速医学超声影像系统,其特征在于将算法成像模块主要计算过程分为计算信号延迟,计算协方差矩阵,计算自适应权重,计算优化算子和计算像素幅度值5个部分,这些计算过程在GPU线程块中通过多线程协作完成。

算法成像模块可根据实际诊断场景需求经由初始启动模块进行设定后调用,采用策略模式对各MV高清成像算法进行封装,并实现可插拔式调用,可方便地进行增加、删除或升级操作。

6. 根据权利要求4所述的基于GPU的多种MV高清算法快速医学超声影像系统,其特征在于结果显示模块对算法成像模块的计算结果进行进一步的处理后将成像结果输出到显示器中进行显示;超声成像在经过波束形成后,通常还会进行包络检波、对数压缩和都灰阶范围校正,结果显示模块通过调用Matlab相应函数对数据进行希尔伯特变换、对数压缩、灰阶范围校正操作后,将图像相关数据输出到对应的横纵坐标轴内,最终在屏幕上显示出超声成像结果;同时,结果显示模块也会收集实际诊断场景的相关信息,并将其显示在显示器的合适区域中,为使用人员提供全面详尽的信息供诊断参考。

## 基于GPU的多种MV高清算法快速医学超声影像系统

### 技术领域

[0001] 本发明属于医学超声成像领域,具体涉及基于GPU的多种MV高清算法快速医学超声影像系统。

### 背景技术

[0002] 近年来,医学超声成像技术被广泛地应用并得到快速发展。与其它医学成像方法相比,超声成像技术具有可实时输出图像,高安全性和低花费等优势。作为一种常见的医学诊断技术,它通常用于观察胎儿发育,心脏运动和血液流动等。其中,波束成形算法是医学超声成像技术的关键部分,对图像质量具有关键影响。

[0003] 现在广泛流通使用的超声成像算法是延迟叠加波束形成算法,这是一种容易实现在不同计算平台且能快捷实现实时要求的算法。虽然延迟叠加波束形成算法有效应用于很多诊断场景,但是在一些复杂的诊断场景中它就不能提供足够的解剖结构细节。因此,本发明应用了多种更高清的自适应最小方差波束形成(MV)算法。自适应波束形成算法自适应于输入的回波数据,在运行时动态地计算变迹权重。由于利用了输入回波信号的实时统计特性,这些算法有效提高了成像质量,能提供更多的解剖结构细节,从而提高诊断正确率。

[0004] MV算法得到高质量成像的代价是更高的计算复杂度,所以,它的计算需要耗费更多的时间。这样的计算复杂度阻碍了最小方差波束形成算法在实际临床上的应用。传统的医学超声成像系统的核心计算通常在中央处理器CPU上实现,但高清成像算法的庞大成像数据及复杂的运算过程,使得传统CPU已经无法满足其快速成像的需求,而现在的GPU(图形处理器)发展迅猛,一个GPU可集成上百乃至上千个运算核心,具有极其强大的高性能计算能力。因此,本发明将多种MV算法在高性能GPU上的实现,大大提高了算法的成像速度,以此完成高清医学成像设备的核心处理模块,为临床诊断提供快速成像依据。

[0005] 目前便携式医学超声检测仪中使用的成像算法大多是传统的延时叠加成像算法,该算法运算简单,能满足医学超声的实时成像要求,但图像质量相对较低。或者也有采用较高清算法的,但因其较大的计算量导致其成像速度很不理想。

### 发明内容

[0006] 本发明的主要目的是解决目前医学超声成像设备成像质量低及速度慢的问题。本发明为了实现高清成像算法在医学超声检测仪中的应用,提供基于GPU的多种MV高清算法快速医学超声影像系统,基于GPU的高速并行计算能力完成多种MV高清成像算法的成像实现,达到快速的高清成像要求。

[0007] 本发明的目的通过以下技术方案实现。

[0008] 基于GPU的多种MV高清算法快速医学超声影像系统,包括初始启动模块、模拟仿真模块、算法成像模块和结果显示模块;其中,初始启动模块完成系统初始化设置后调用算法成像模块进行成像,模拟仿真模块作为一个辅助模块产生模拟数据供其他模块使用,算法成像模块是整个系统的核心模块,接收来自初始启动模块的参数设置与模块仿真模块或数

据采集设备的数据进行成像计算,而结果显示模块则对其他模块的结果进行最后的处理与显示,各个模块之间既独立工作又相互协作,最终实现多种MV算法的快速成像功能。

[0009] 进一步的,初始启动模块主要进行系统启动时需要的初始化工作,包括初始化参数、设置成像场景、设置启用的算法、评估指标及图像显示参数,其中大多数初始化设置提供了与传统超声成像系统一致的选项,而对于成像算法的设置,本系统则提供了多种高清成像算法的选择,包括常规MV算法、迭代MV算法、融合对角加载的MV算法、前后向平滑MV算法、融合广义相干因子的MV算法、基于特征空间处理的MV算法和特征空间与符号相干系数融合的MV算法。此模块作为初始启动,一方面进行系统检查,防止因设备故障而导致错误的诊断结果,一方面除了系统内置的比较全面的初始设置外,还可供专业人员根据实际使用场景对初始参数、使用的算法以及显示范围等进行设置,这可有效缩短运行时间并且增加准确性。初始启动模块在基本设置完毕后可通过模拟仿真模块或成像数据采集设备取得成像所需数据启用算法成像模块进行成像。

[0010] 进一步的,模拟仿真模块采用Field II模拟超声成像过程取得仿真数据,首先模拟现实的超声成像设备设置需要的物理数据,创建发射和接收阵元,另外需根据实际应用场景需要创建虚拟的物体,然后,按扫描线逐条计算并收集回波数据。此模块获得的模拟数据可作为算法成像模块的初始输入数据,一方面可用于系统软硬件的预热启动,一方面可作为一个判断的对比基准,以评估或预览不同场景下各算法的成像性能,为实际诊断场景的算法选择提供参考。此模块通常由初始启动模块进行调用启动并将产生的数据传给算法成像模块作进一步的成像计算。

[0011] 进一步的,算法成像模块是本系统的核心部分。超声系统在成像过程中需要实现的一个基本功能是在成像目标区域产生特定的声场分布,即波束形成,因而波束形成算法是医学超声成像技术的关键部分,对图像质量具有关键影响。本系统的算法成像模块包含常规MV算法、迭代MV算法、融合对角加载的MV算法、前后向平滑MV算法、融合广义相干因子的MV算法、基于特征空间处理的MV算法和特征空间与符号相干系数融合的MV算法。这些算法都是高清的自适应波束形成算法,相比于传统预固定权重的DAS算法,此模块中的算法可得到更高质量的成像,同时,本系统基于GPU实现了这7种算法的快速成像,利用GPU的高性能计算能力解决了高质量成像带来的计算量过大问题。

[0012] 算法成像模块由初始模块进行调用启动,通过数据采集设备或模拟仿真模块获得成像所需的原始信号数据,然后根据MV算法进行像素幅度值计算,取得成像,并将成像结果交给结果显示模块以显示最终的超声图像。

[0013] 进一步的,对各MV算法进行比较分析等综合研究后,可将算法成像模块主要计算过程分为计算信号延迟,计算协方差矩阵,计算自适应权重,计算优化算子和计算像素幅度值5个部分,这些计算过程在GPU线程块中通过多线程协作完成;其中,优化算子的计算部分为各个MV算法的不同之处,体现了各个算法的核心原理,各个算法的更具体计算原理及其在GPU中的实现将在实例中进行阐述。

[0014] 算法成像模块可根据实际诊断场景需求经由初始启动模块进行设定后调用,采用策略模式对各MV高清成像算法进行封装,并实现可插拔式调用,可方便地进行增加、删除或升级操作。同时,各个MV算法均可组合不同的M和L(接收信号阵元数量及子阵列阵元数量)等不同参数进行成像,为临床诊断提供更多维度的选择。

[0015] 进一步的,结果显示模块对算法成像模块的计算结果进行进一步的处理后将成像结果输出到显示器中进行显示;超声成像在经过波束形成后,通常还会进行包络检波(采用希尔伯特变换)、对数压缩和都灰阶范围校正,结果显示模块通过调用Maltab相应函数对数据进行希尔伯特变换、对数压缩、灰阶范围校正操作后,将图像相关数据输出到对应的纵横坐标轴内,最终在屏幕上显示出超声成像结果;同时,结果显示模块也会收集实际诊断场景的相关信息如硬件状态、成像参数、评估指标等,并将其显示在显示器的合适区域中,为使用人员提供全面详尽的信息供诊断参考。

[0016] 本发明采用基于GPU实现的多种MV高清算法来实现快速的医学超声成像,并且成像算法本身针对GPU架构做出了相应的适配和改进,使其计算流程适用于GPU计算环境,从而提高医学超声成像的图像质量与成像速度。系统采用的编程实现方案基于GPU的编程架构及内存层次对多种MV高清算法进行针对性的并行化编程,以实现各高清算法快速成像的功能。

[0017] 本发明依据现有科学技术现状,在带有GPU硬件的计算机系统中实现一个快速的医学超声高清成像系统。其特征在于基于GPU实现多种MV算法来达到快速的医学超声高清成像目的,其重点是对多种高清MV算法的计算过程进行并行化处理,使其运算流程适用于GPU的运算环境,从而充分发挥GPU的计算能力,提高医学超声成像的图像质量与成像帧率。

[0018] 与现有技术相比,本发明的优点主要体现在两个方面:一方面,本发明综合实现了多种高清MV成像算法,可有效提高医学超声成像的图像质量,其成像效果比传统的延迟叠加算法好很多;另一方面,通过GPU强大的计算能力妥善解决各高清MV成像算法庞大的计算需求,从而使成像系统的输出成像帧率大幅提高,可以快速地输出高清的医学超声图像。

## 附图说明

[0019] 图1为基于GPU实现的多种MV高清算法快速医学超声影像系统工作流程示意图。

[0020] 图2(a)为实例中模拟的实验方案示意图。

[0021] 图2(b)~(h)为实例中各算法输出的超声成像示例图。

## 具体实施方式

[0022] 以下结合附图和实例对本发明的具体实施作进一步说明,但本发明的实施和保护不限于此。需指出的是,若有未特别详细说明之处或现有算法公式符号等,均是本领域技术人员可参考现有技术实现或理解的,在此不再赘述。

[0023] 本系统GPU编程架构分为三个层次:网格、块和线程。当内核调用GPU计算时,所有的计算状态均在一个GPU计算网格中完成。一个GPU计算网格计算一幅完整的医学超声图像,而输出的目标图像则被分解为扫描线和像素。因此,一个GPU计算网格包含一个矩阵的计算块,每个块根据各MV算法的原理进行像素幅度值的计算。为了获得一个像素的最终幅度估算值,各MV算法的计算细节通过一些并行的计算线程实现(更具体的算法原理及其实现已在发明内容中进行了说明)。这些同时运行的线程利用共享内存进行合作并通过计算独立元素来进行数组或矩阵运算。此处计算层次中块和线程的最佳数值设置依赖于计算平台的资源和计算问题的大小。

[0024] 在GPU的实现中,数据从CPU内存中拷贝到GPU显存中,经GPU计算后,将成像数据拷

贝回CPU内存中进行显示。在这个过程中,内存的利用率是非常重要的。GPU有三个基本的内存层次:全局内存,共享内存和寄存器内存。这三个层次的内存访问时间长短关系是:全局存储器<共享内存<寄存器内存。所以,为了充分发挥GPU的计算性能,在计算过程中应尽量少地使用全局内存,而更多地使用寄存器内存进行计算。然而,三种内存类型的容量大小关系为:全局存储器>共享内存>寄存器内存。因此,本系统在进行资源分配的时候,尽量考虑小型的变量存储在寄存器中,而大型的数据则存储在全局内存中。。

[0025] 本系统根据实际使用场景进行模块设计,其总体模块设计如图1所示,具体可分为初始启动模块,模拟仿真模块,算法成像模块和结果显示模块等4个模块。每个模块之间既独立工作又相互合作,最终实现多种MV算法的快速成像功能,以下对各个模块的具体设计进行阐述。

#### [0026] (1) 初始启动模块

[0027] 初始启动模块主要进行系统启动时需要的初始化工作,包括初始化参数,设置成像场景,设置启用的算法及图像显示参数等。此模块作为初始启动,一方面进行系统检查,防止因设备故障而导致错误的诊断结果,一方面除了系统内置的比较全面的初始设置外,还可供专业人员根据实际使用场景对初始参数、使用的算法以及显示范围等进行设置,这可有效缩短运行时间并且增加准确性。初始启动模块在基本设置完毕后通过模拟仿真模块取得成像所需数据启用算法成像模块进行成像。

#### [0028] (2) 模拟仿真模块

[0029] 模拟仿真模块采用Field ii模拟超声成像过程取得仿真数据,首先模拟现实的超声成像设备设置需要的物理数据,创建发射和接收阵元,另外需根据实际应用场景需要创建虚拟的物体,然后,按扫描线逐条计算并收集回波数据。此模块获得的模拟数据可作为算法成像模块的初始输入数据,以评估此场景下各算法的成像性能,为选择算法提供参考。

#### [0030] (3) 算法成像模块

[0031] 算法成像模块包含了多种医学超声MV成像算法,是本系统的核心部分。此模块由初始模块进行调用启动,结合模拟仿真模块获得成像所需的原始信号数据,然后根据MV算法进行像素幅度值计算,取得成像,并将成像结果交给结果显示模块以显示最终的超声图像。对各MV算法进行比较分析等综合研究后,可将此模块主要计算过程分为5个部分,即计算信号延迟,计算协方差矩阵,计算自适应权重,其他改进计算以及计算像素幅度值,这些计算过程在GPU线程块中通过多线程协作完成。其中,改进计算部分为各个MV算法的不同之处,体现了各个算法的核心原理,各个算法的更具体计算原理及其在GPU中的实现将在下面进行阐述。

[0032] 为了方便系统算法的升级及拓展,本模块实现采用策略模式,由初始启动模块根据需要调用,各MV算法独立实现并且可插拔,方便增加、删除或者进行升级改造。并且各个MV算法均可组合不同的M和L(接收信号阵元数量及子阵列阵元数量)等不同参数进行成像,为临床诊断提供更多选择。

#### [0033] (4) 结果显示模块

[0034] 算法成像模块经计算得到成像数据后,由结果显示模块进行结果呈现,此模块通过调用Matlab相应函数对数据进行希尔伯特变换、对数压缩、灰阶范围校正等操作后,将图像相关数据输出到对应的横纵坐标轴内,最终,在屏幕上显示出超声成像结果。

[0035] 以下对本实例的多种医学超声MV成像算法进行说明,以下只是举例,鉴于篇幅,部分内容如公式的含义并未详述,本领域技术人员可参照现有技术实现或者理解。

[0036] 1、常规MV算法

[0037] (1) 算法介绍

[0038] 根据医学超声成像中最小方差小波束形成算法的发展,最常见的自适应波束形成算法是基于Capon在1969年设计的自适应最小方差(MV)波束形成器。常规MV算法是基于延迟叠加波束形成算法发展而来的,它们有相同的输入输出数据流和相同的延迟叠加处理。最主要的不同是最小方差波束形成算法使用变迹权重自适应输入的超声数据,而延迟叠加波束形成算法不能自适应输入数据的固定的变迹权重。正是这个最主要的区别使最小方差波束形成算法输出图像的质量比延迟叠加算法的高。

[0039] MV波束形成算法使用了子孔径平均法。一个接收孔径由M个连续的输入数据通道构成并被分成一组由L个连续输入通道组成的子孔径。因此,一个接收孔径由(M-L+1)个子孔径组成。通过子孔径平均法,可以用以下式子计算像素点p的协方差矩阵 $R_{MV}(p)$ 。

$$[0040] \quad R_{MV}(p) = \frac{1}{M-L+1} \sum_{i=1}^{M-L+1} x_i(p)x_i^H(p), \quad (3-1)$$

[0041] 其中 $x_i(p)$ 是一个(L×1)的向量,由第 $i^{th}$ 子阵列中输入的回波信号数据构成,如, $x_i(p)$ 代表 $x(p)$ 中第 $i^{th}$ 到 $(i+L-1)^{th}$ 个元素的集合,而 $x(p)$ 则是一个由与像素点对应的经过延迟之后的信号值构成的一个(L×1)的向量。协方差矩阵 $R_{MV}(p)$ 的计算完成后,自适应的变迹权重便可通过下式进行估计:

$$[0042] \quad w_{MV}(p) = \frac{R_{MV}^{-1}(p) a}{a^H R_{MV}^{-1}(p) a}, \quad (3-2)$$

[0043] 因为输入通道中的数据已经过延迟,所以此处 $a$ 是一个均为1的简单方向向量。最后,像素点p的幅度值可通过下式进行估算:

$$[0044] \quad v_{MV}(p) = \frac{1}{M-L+1} \sum_{i=1}^{M-L+1} w^H(p)x_i(p). \quad (3-3)$$

[0045] (2) GPU实现

[0046] 常规MV算法的GPU实现是其他改进MV算法的基础。其计算网格和线程块大小以二维下标组织,其中,计算网格的X,Y大小正好与成像的扫描线数及每条扫描线上的像素点数对应;线程块中 $X \geq L$ ,Y根据设置的每线程块总线程数与X的大小取得。在启动GPU计算之前,在共享内存中分配存放延迟之后的信号数据、协方差矩阵、权重向量数值及中间值的空间,而原始信号数据通过CPU拷贝存进GPU全局内存中。

[0047] 延迟计算部分根据扫描线位置获取有效的起始和结束阵元,然后通过线程块中的线程并行地计算延迟取得相应信号存放在共享内存中的相应地址,这个过程中,由于阵元位置具有对称性,部分信号通过判断只需进行一次计算过程便可取得相应信号,以此可实现计算量的减少。

[0048] 因MV算法的协方差矩阵具有对称性,可只计算对角线上及上三角部分的值,以此也达到节省计算量的目的。计算过程在一个线程块中通过多线程进行并行计算。

[0049] 权重向量计算部分,本系统通过Cholesky分解直接计算出协方差矩阵的逆与方向向量的乘积,以此省去求逆过程,并在一个线程块中通过多线程协作并行计算。

[0050] 最后的叠加部分因实际阵元数量级较小,因此能过特定线程直接计算即可得。通

过以上步骤后,可充分发挥GPU强大的计算性能,而实现高清MV算法的快速成像过程。

[0051] 2、迭代MV算法

[0052] (1) 算法介绍

[0053] 常规MV波束形成算法的核心是噪声协方差矩阵的估计,而造成MV误差的主要是使用样本协方差矩阵代替噪声协方差矩阵的结果。因此,Eng Nai Wee, Ser等人在2011年提出了一种迭代最小方差波束形成(IMV)算法,希望通过多次迭代提高噪声协方差矩阵的估计精度。

[0054] 在该算法中,常规MV波束形成算法的输出被作为期望信号幅度的预估值:

$$[0055] \hat{s}_{IMV} = v_{MV}(p). \quad (3-4)$$

[0056] 期望信号被从接收信号中分离出来,剩下的是干扰和噪声分量,如式(2-5)所示:

$$[0057] \hat{x}_{IMV}(p) = x(p) - \hat{s}_{IMV} a. \quad (3-5)$$

[0058] 然后,只需用 $\hat{x}_{IMV}(p)$ 代替 $x_i(p)$ ,按常规MV算法的公式计算一个新的波束形成权重矢量,并代入式(3-3)中最终获得像素点的幅度估计。由于该步骤相当于对干扰噪声分量重复一遍MV优化过程,因此我们称之为二步迭代最小方差(2StgMV)算法。而且这个步骤可以根据需要重复进行。

[0059] IMV算法可以显著提高MV算法的对比度,但是迭代过程中的求逆计算会造成很大的计算开销,使之不利于实现算法的实时性。

[0060] (2) GPU实现

[0061] 迭代MV算法的计算过程实质上与常规MV算法的一致,只是重复了常规MV算法中求权重的过程,并在计算过程中减去上一次迭代的幅度估值。因此,实现时通过增加一个变量记录上一次计算得到的幅度值,并在每次读原始信号时减去,其余过程采用常规MV算法的并行计算方案,以此通过进一步的迭代计算便可获得迭代MV算法的像素幅度值。其成像速度相比于常规MV算法取决于希望迭代的次数。

[0062] 3、融合对角加载的MV算法

[0063] (1) 算法介绍

[0064] 由于频率或到达方向的不准确估计,常规MV波束成形算法的性能会迅速下降。Li等人使用对角加载(DL)的方法来提高MV波束形成算法的鲁棒性。在2005年,Sasso等人将其应用于医学超声成像领域,这也在一定程度上提高了成像分辨率。

[0065] 对角加载过程是在原矩阵上叠加一定比例的单位阵,如下式所示:

$$[0066] \hat{R}_{DL}(p) = R_{MV}(p) + \delta \cdot \text{trace}\{R_{MV}(p)\} \cdot I, \quad (3-6)$$

[0067] 其中 $\delta$ 为对角加载量,  $\text{trace}\{\}$ 表示矩阵的迹,  $I$ 表示单位阵。然后,用 $\hat{R}_{DL}(p)$ 代替 $R_{MV}(p)$ 按式(3-2)和式(3-3)一样进行计算来获得像素点的幅度估计。

[0068] 对角加载的含义是在原信号上叠加与信号功率成一定比例的高斯白噪声,增加信号间的独立性。但它在改善鲁棒性的同时,会降低算法的自适应性,造成分辨率的降低,使成像性能趋向于传统延时叠加算法,因此对角加载量不宜过大,应满足: $\delta \ll 1/L$ ,且通常取: $\delta = 1/(100L)$ 。

[0069] (2) GPU实现

[0070] 融合对角加载的MV算法主要是在常规MV算法得出协方差矩阵后在其上加上一

比例的高斯白噪声,因此依然可沿用前述的常规MV算法GPU并行实现方案,然后,在此基础上,于协方差矩阵计算完成后组织部分线程计算出需要加上的数值,然后加到矩阵对角线上。因叠加过程无访问冲突,因此也可通过线程块中的L个线程进行并行操作以更快完成对角加载的过程,实现快速成像的目的。

[0071] 4、前后向平滑MV算法

[0072] (1) 算法介绍

[0073] 前后向平滑最小方差 (FBMV) 波束形成算法是Asl等人针对提高对比度提出的MV算法的改良算法。该算法的核心思想是估计更准确的计算变迹权重所需的协方差矩阵。它使用前向-后向平滑技术,而不是传统的仅前向空间平滑,从而获得更精确的背景斑点统计特性,并因此提高对比度。

[0074] 式 (3-1) 给出的估计方法可认为是前向空间平滑的估计,记为 $\hat{\mathbf{R}}_F(p)$ :

$$[0075] \quad \hat{\mathbf{R}}_F(p) = \mathbf{R}_{MV}(p). \quad (3-7)$$

[0076] 通过式 (3-8) 进行计算获得后向空间平滑的估计 $\hat{\mathbf{R}}_B(p)$ :

$$[0077] \quad \hat{\mathbf{R}}_B(p) = \mathbf{J} \cdot \hat{\mathbf{R}}_F(p) \cdot \mathbf{J}, \quad (3-8)$$

[0078] 其中,J表示交换矩阵,即将单位矩阵左右翻转。

[0079] 进而通过联合前后向空间平滑估计可以获得一个更准确的估计,其表达式如下:

$$[0080] \quad \hat{\mathbf{R}}_{FB}(p) = \frac{1}{2}(\hat{\mathbf{R}}_F(p) + \hat{\mathbf{R}}_B(p)). \quad (3-9)$$

[0081] 然后,用 $\hat{\mathbf{R}}_{FB}(p)$ 代替 $\mathbf{R}_{MV}(p)$ 按式 (3-2) 和式 (3-3) 一样进行计算来获得像素点的幅度估计。这样的平滑过程也使其比常规MV算法具有更强的鲁棒性。

[0082] (2) GPU实现

[0083] 从算法公式上可看出,前后向平滑MV算法将常规MV算法得出的协方差矩阵后对应于前向空间平滑的估计,而后向空间平滑的数值可在前向估计的相对位置上取得,因此,可在常规MV算法完成其协方差矩阵计算后,再采用一次计算协方差矩阵时的并行方案取得后向平滑的值存储于临时变量中,与此同时,因联合估计的数值为前后向估计的平均,因而数值上是相等的,可在得到两者数值后直接进行平均计算存于矩阵中相应位置。采用此方案可在不增加内存开销的基础上实现很好的并行计算,大大节省了联合协方差矩阵的计算时间。

[0084] 6、融合广义相干因子的MV算法

[0085] (1) 算法介绍

[0086] 为了获得高分辨率并校正由于组织传播中的声速不均匀引起的相位误差,吴文焘等人在2010年提出了融合广义相干因子 (GCF) 与MV波束形成算法的高分辨率医学超声成像方法。

[0087] GCF的计算基于聚焦延迟后的多通道数据。首先,将阵元域数据通过傅里叶变换从阵元域变换到波束域:

$$[0088] \quad \mathbf{y}(m) = e^{j\pi m} \sum_{i=0}^{M-1} \mathbf{x}_i(p) e^{-j2\pi p \frac{m}{M}}, \quad (3-10)$$

[0089] 其中 $\mathbf{y}(m)$ 为变换到波束域后的数据,然后计算各个波束方向的能量,得到相干方

向的能量与总能量的比作为GCF的值:

$$[0090] \quad c_{GCF}(p) = \frac{\sum_{m \in (0.1 \dots K)} |y(m)|^2}{\sum_{m=0}^{M-1} |y(m)|^2}, \quad (3-11)$$

[0091] 其中K为控制GCF的低频成分的能量比,通过改变K的数值,可以改变算法的性能。

[0092] 为了计算得像素点的幅度值,常规MV算法的计算过程应被完整进行一遍,然后,通过式(3-12)与GCF结合,从而获得像素点最后的幅度估计。

$$[0093] \quad v_{MV\_GCF}(p) = c_{GCF}(p) \cdot v_{MV}(p). \quad (3-12)$$

[0094] 这个方法把广义相干系数在相位存在误差时的鲁棒性与MV算法的高分辨率能力结合了起来。

[0095] (2) GPU实现

[0096] 融合广义相干因子的MV算法具有与常规MV算法一致的计算像素幅度值的计算过程,因此,常规MV算法的GPU实现方案依然应用于此。然后,在得到像素幅度值之后,对存储于共享内存中的阵元数据进行傅里叶变换,并根据如上算法所述计算出相干因子,最后乘上常规MV算法得到的幅度值。在算法改进的计算部分对于没有读写交联的部分依然可在同一个线程块中通过多个线程并行进行,以此加快完成计算过程。此外,本系统在实现过程中,也尽量采用更少计算量的等效替换计算或省去不必要的计算,以进一步加速成像速度。

[0097] 7、基于特征空间处理的MV算法

[0098] (1) 算法介绍

[0099] 最小方差波束形成算法能在一定程度上改善图像的分辨率,但不能显著提高图像的对比度。为此,Asl等人提出了基于特征空间处理的最小方差(ESBMV)波束形成算法,ESBMV算法利用协方差矩阵的特征结构来增强常规MV算法的性能,它可以在保持主瓣信号的同时降低旁瓣幅度,从而提高成像质量。

[0100] ESBMV算法在计算变迹权重之前的步骤与常规MV算法一样。然后,协方差矩阵的特征分解可以表示为:

$$[0101] \quad \mathbf{R}_{MV}(p) = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H = \mathbf{U}_S\mathbf{\Lambda}_S\mathbf{U}_S^H + \mathbf{U}_N\mathbf{\Lambda}_N\mathbf{U}_N^H, \quad (3-13)$$

[0102] 此处 $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_L)$ , 其中 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L$ 为协方差矩阵的特征值; $\mathbf{U} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L]$ ,  $\mathbf{v}_i$ 为与 $\lambda_i$ 对应的特征向量。 $\mathbf{U}_S = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{N_{sig}}]$ 为信号子空间,由 $N_{sig}$ 个较大特征值对应的特征向量组成的,而 $\mathbf{U}_N = [\mathbf{v}_{N_{sig}+1}, \mathbf{v}_{N_{sig}+2}, \dots, \mathbf{v}_L]$ 则为噪声子空间。 $N_{sig}$ 的取值决定了波束形成方法保持主瓣信号和降低旁瓣信号的能力,通常用大于最大特征值 $\alpha$  ( $0.1 \geq \alpha \geq .5$ ) 倍或大于最小特征值 $\beta$  ( $0.1 \geq \beta \geq .5$ ) 倍的特征向量个数决定。

[0103] 将常规MV算法得到的加权值投影到特征空间法得到的信号子空间,即可得到ESBMV的加权值为:

$$[0104] \quad \mathbf{w}_{ESBMV}(p) = \mathbf{U}_S\mathbf{U}_S^H\mathbf{w}_{MV}(p). \quad (3-14)$$

[0105] 由此得到的ESBMV波束形成的最终输出为:

[0106]

$$v_{ESBMV}(p) = \frac{1}{M-L+1} \sum_{i=1}^{M-L+1} \mathbf{w}_{ESBMV}^H(p) \mathbf{x}_i(p). \quad (3-15)$$

[0107] 与常规MV波束形成相比,ESBMV能够有效对比度和信噪比。然而,它的缺陷在于当

噪声超过期望信号的强度时,该算法会将噪声认作期望信号,从而可能导致完全错误的结果。

#### [0108] (2) GPU实现

[0109] 基于特征空间处理的MV算法比常规MV算法多了一步计算量较大的特征空间分解过程。基于精度和并行性的考虑,这里采用Jacobi分解方法进行特征分解,实现依然对无访问冲突的部分尽量在线程块中通过多线程并行进行。为了进行线程间的并行协作,需要在原有MV算法的基础上增加用于存储特征分解过程中间值的共享内存,这里主要用于存储特征向量,特征值可存储于最后计算已无需用到的已经分配的共享内存中。因设置共享内存的大小会限制实际GPU中可并行运行的线程块数,因此为了节省内存,实现过程中,根据计算过程的对称性,实际只需增加一半矩阵的容量,需用到其数据时再通过下标变换计算取得相应的数据,这样不仅可节省共享内存的使用,增加并行的线程块数,还可减少将近一半的计算量,从而达到尽量快速成像的目的。

### [0110] 8、特征空间与符号相干系数融合的MV算法

#### [0111] (1) 算法介绍

[0112] 图像分辨率主要由主瓣宽度决定,而ESBMV波束形成得到图像的分辨率几乎与常规MV算法近似相同,为了在提高图像对比度的同时进一步提高图像分辨率,刘廷平等人在2015年把符号相干系数引入到ESBMV算法中,提出了特征空间与符号相干系数(SCF)融合的波束形成算法。

[0113] SCF可以被认为是波束形成设备的一个非线性函数。它只需较少的硬件资源并且能被快速集成到波束形成设备中,可以通过式(3-16)计算其值:

$$[0114] \quad c_{SCF}(p) = \left| 1 - \sqrt{1 - \left( \frac{1}{N} \sum_{i=1}^N b_i(p) \right)^2} \right|^q, \quad (3-16)$$

[0115] 其中,  $b_i(p)$  可以通过式(3-17)获得:

$$[0116] \quad b_i(p) = \begin{cases} +1, & x_i(p) \geq 0 \\ -1, & x_i(p) < 0 \end{cases}, \quad (3-17)$$

[0117] 而其他的计算过程与ESBMV算法一致,只是其最后的输出通过式(2-18)获得:

$$[0118] \quad v_{ESBMV\_SCF}(p) = c_{SCF}(p) \cdot v_{ESBMV}(p). \quad (3-18)$$

[0119] 这个方法把相干系数相对于相位误差的稳健性与ESBMV的高分辨率特性结合了起来。

#### [0120] (2) GPU实现

[0121] 特征空间与符号相干系数融合的MV算法的计算过程主要是在基于特征空间处理的MV算法得到幅度值后再乘上一个符号相干系数,因此,其GPU实现方案与基于特征空间处理的MV算法一致,只需在最后根据如前所述公式多进行一步符号相干系数的计算,此过程与融合广义相干因子的MV算法一样,通过线程并行、替换计算等尽量省去不必要的计算,以加速成像速度。

[0122] 该系统通过使用Field II仿真器来模拟超声通道数据样本,从而进行一系列相关实验,得到此系统的成像性能。以下仿真模拟了一个如图2(a)所示的处于散斑中的囊泡虚像作为成像场景。实验采用Field II模拟器来模拟超声信道的回波数据样本,其仿真参数

设置见表1。

[0123] 表1实验中Field II模拟参数的设置

参数	取值
人体组织中的超声传播速度	1540 m/s
传感器阵元数量	128
[0124] 阵元间隔	0.3048 mm
脉冲重复频率	5 KHz
采样频率	40 MHz

[0125] 前述各MV算法都通过实验进行分析,其中每个算法均作为一个波束形成器进行编程实现,并且将像素计算过程并行化以节省程序运行时间。而为了获得具有比较意义的结果,每个算法的参数均被设置为其常用值,例如,子孔径通道的数目被设置为32。测试结果显示在图2 (b) ~ (h) 中,所有图像显示的动态范围均为60dB。对比这些图可看出,各种波束形成器的成像清晰度都足以辨别出目标,体现了这些MV算法高清成像的能力,但不同算法的对比度在此成像场景中还是有一定差别的,因此,临床使用时需要根据实际需求选择不同的算法进行成像。波束成形算法是医疗超声成像技术中的关键部分,MV自适应波束成形算法由于利用了输入回波数据的实时统计特性,因此,与传统的DAS波束成形算法相比,其具有更高的输出成像质量。而本系统通过GPU强大的高性能计算能力实现多种高清MV算法的快速成像,可为临床诊断提供快速有效的成像依据。

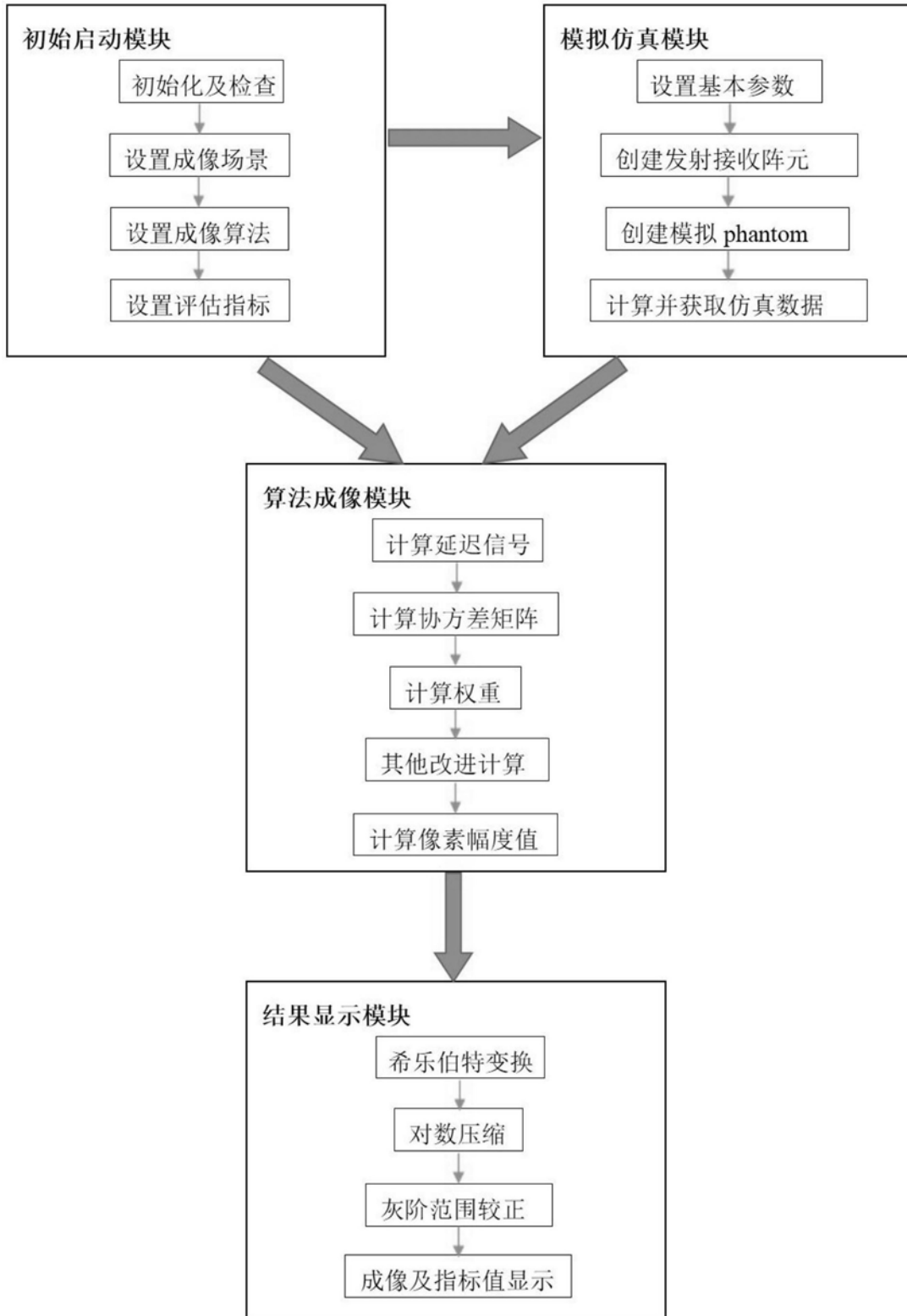


图1

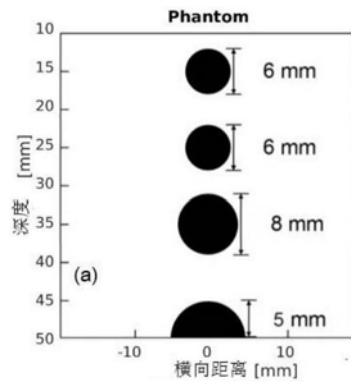


图2 (a)

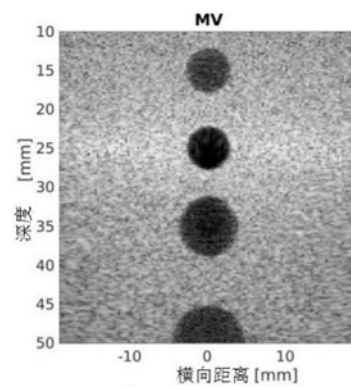


图2 (b)

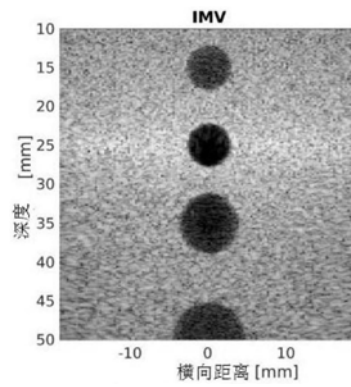


图2 (c)

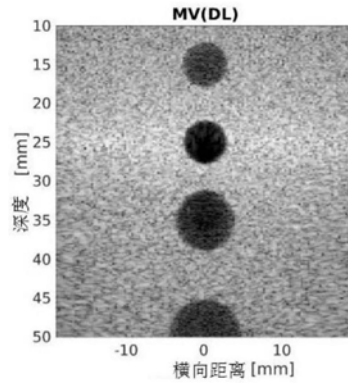


图2 (d)

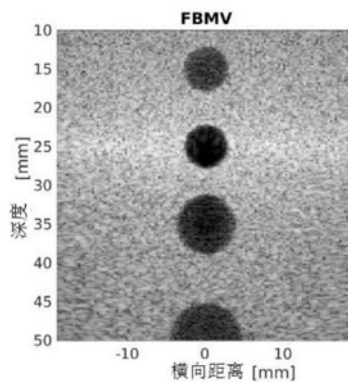


图2 (e)

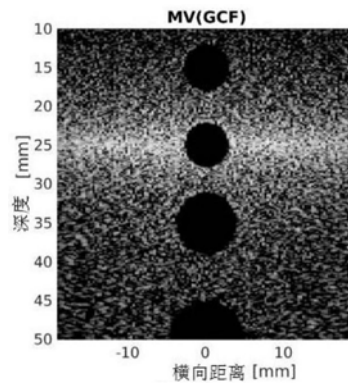


图2 (f)

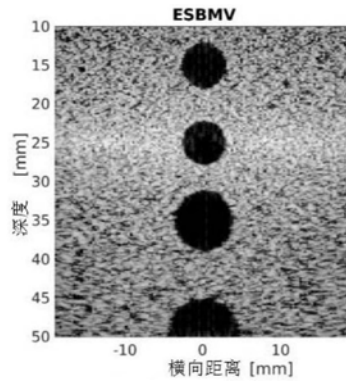


图2 (g)

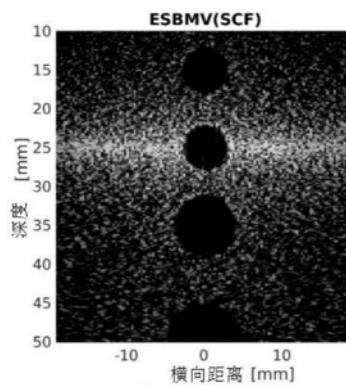


图2 (h)

专利名称(译)	基于GPU的多种MV高清算法快速医学超声影像系统		
公开(公告)号	<a href="#">CN108354626A</a>	公开(公告)日	2018-08-03
申请号	CN201810278570.5	申请日	2018-03-31
[标]申请(专利权)人(译)	华南理工大学		
申请(专利权)人(译)	华南理工大学		
当前申请(专利权)人(译)	华南理工大学		
[标]发明人	陈俊颖 陈锦辉 闵华清		
发明人	陈俊颖 陈锦辉 闵华清		
IPC分类号	A61B8/00		
CPC分类号	A61B8/4411 A61B8/5207 A61B8/5269		
代理人(译)	何淑珍		
外部链接	<a href="#">Espacenet</a> <a href="#">SIPO</a>		

摘要(译)

本发明公开基于GPU的多种MV高清算法快速医学超声影像系统。本发明的系统包括初始启动模块、模拟仿真模块、算法成像模块和结果显示模块；其中，初始启动模块完成系统初始化设置后调用算法成像模块进行成像，模拟仿真模块作为一个辅助模块产生模拟数据供其他模块使用，算法成像模块是整个系统的核心模块，接收来自初始启动模块的参数设置与模块仿真模块或数据采集设备的数据进行成像计算，而结果显示模块则对其他模块的结果进行最后的处理与显示，最终实现多种MV算法的快速成像功能。本发明可在极短的时间内完成高清医学超声图像算法的复杂计算，使用的各MV高清算法的GPU编程实现方案可方便地部署于带有GPU的计算机上，可有效满足使用需求，实用性强。

