



(12) 发明专利

(10) 授权公告号 CN 104306022 B

(45) 授权公告日 2016.05.25

(21) 申请号 201410578176.5

(22) 申请日 2014.10.24

(73) 专利权人 西安电子科技大学  
地址 710071 陕西省西安市太白南路2号

(72) 发明人 林杰 韩亭玉 贺玉高 石光明

(74) 专利代理机构 陕西电子工业专利中心  
61205

代理人 王品华 朱卫星

(51) Int. Cl.  
A61B 8/00(2006.01)

(56) 对比文件

- CN 103235298 A, 2013.08.07,
- CN 102058416 A, 2011.05.18,
- CN 103983968 A, 2014.08.13,
- US 2013/0310678 A1, 2013.11.21,
- US 2009/0076387 A1, 2009.03.19,
- WO 2006/078928 A2, 2006.07.27,
- Sergry voronin, etc. A iterative

thresholding algorithm for linear inverse problems with multi-constraints and its applications. 《ELSEVIER》.2012,  
Massimo Fornasier, etc. Iterative thresholding algorithms. 《ELSEVIER》.2008,  
石光明, 刘丹华, 等. 压缩感知理论及其研究进展. 《电子学报》.2009, 第37卷(第5期),  
Sergey Voronin, etc. A new iterative firm-thresholding algorithm for inverse problems with sparsity constraints. 《ELSEVIER》. 2013,

审查员 孙晓彤

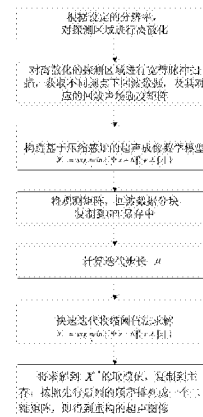
权利要求书3页 说明书7页 附图6页

(54) 发明名称

用 GPU 实现压缩感知超声成像的方法

(57) 摘要

本发明公开了一种用 GPU 实现压缩感知超声成像的方法, 主要解决压缩感知理论框架下成像重建时间较慢的问题。其实现步骤为: 1. 根据设定的分辨率, 对探测区域进行离散化并对该区域进行宽带脉冲扫描, 得到回波向量和观测矩阵, 进而建立超声成像数学模型; 2. 将回波向量与观测矩阵分块并复制到 GPU 显存中; 3. 在 GPU 中计算迭代步长; 4. 将迭代步长带入快速迭代收缩阈值算法求解出重建观测场景散射强度; 5. 将该散射强度复制到主存中取模值并排列成一个二维矩阵, 得到重建的超声图像。本发明相对传统的快速迭代收缩阈值算法, 重建时间从分钟级别降低到毫秒级别, 极大提高了超声成像的实时性, 可用于超声实时处理领域。



1. 一种用GPU实现压缩感知超声成像的方法,其特征在于,包括以下步骤:

(1)将超声探测区域二维离散化,得到N个离散化的像素点,其中 $N=T \times S$ ,T表示轴向像素的个数,S表示侧向像素的个数;

(2)将超声宽带脉冲发射信号在频域均匀采样得到W个频点,按频点顺序依次对已离散化的二维探测区域进行一次平面波扫描,每次扫描得到一个长度为A的局部观测回波向量 $b_t$ ,并将这W个局部观测回波向量按从上到下顺序合成长度为 $M=A \times W$ 的观测回波向量 $b=\{b_1, \dots, b_t, \dots, b_w\}$ ,同时保存由这W个频点产生的回波声场强度矩阵 $\Psi_1, \dots, \Psi_t, \dots, \Psi_w$ ,其中,A表示超声线阵的阵元个数,矩阵 $\Psi_t$ 的宽度为A,长度为N, $1 \leq t \leq W$ ;

(3)将回波声场强度矩阵 $\Psi_1, \dots, \Psi_t, \dots, \Psi_w$ ,按照从上到下顺序排列成一个大小为 $M \times N$ 的观测矩阵 $\Psi$ ;将离散化的二维探测区域按照行优先的顺序排列成一个目标向量 $x$ ;

(4)根据回波向量 $b$ 和观测矩阵 $\Psi$ 定义基于压缩感知的超声成像数学模型:

$$X^* = \underset{x}{\operatorname{arg\,min}} \{ \|\Psi x - b\|_2^2 + \lambda \|x\|_1 \} \quad (1)$$

其中 $X^*$ 为重建观测场景散射强度, $x$ 为目标向量, $\lambda$ 为正则化参数, $\|\Psi x - b\|_2^2$ 表示向量 $\Psi x - b$ 二范数的平方, $\|x\|_1$ 表示目标向量 $x$ 的一范数;

(5)在GPU中对上述数学模型进行求解,得到重建观测场景散射强度 $X^*$ :

(5a)初始化: $n=0, \varepsilon=10^{-3}$ ,其中 $n$ 表示第 $n$ 次迭代, $\varepsilon$ 表示迭代终止条件;

(5b)将观测矩阵 $\Psi$ 按行均匀等分成宽度为 $M/K$ ,长度为 $N$ 的 $K$ 个子矩阵 $\Psi = \{\Psi_{s1}, \dots, \Psi_{si}, \dots, \Psi_{sK}\}$ ,要保证每个子矩阵数据量小于显存的容量上限, $1 \leq i \leq K$ ;

(5c)将每个子矩阵 $\Psi_{si}$ 以列优先顺序向量化为一维行向量 $d\_MA_i$ 并采用流操作拷贝到显存中,此时观测矩阵 $\Psi$ 转换成一个宽度为 $K$ ,长度为 $M/K \times N$ 的向量化矩阵 $d\_MA = \{d\_MA_1, \dots, d\_MA_i, \dots, d\_MA_K\}$ ,其中观测矩阵 $\Psi$ 与向量化矩阵它们存在一一对应的关系;将回波向量 $b$ 等分成长度为 $M/K$ 的 $K$ 块, $b = \{b_1, \dots, b_i, \dots, b_K\}$ ,并拷贝到显存中;

(5d)根据向量化矩阵 $d\_MA$ 计算基于梯度下降算法的迭代步长 $\mu$ :

(5d1)在GPU中建立一个重复执行 $K$ 次的循环,每次循环创建一个 $k\_step$ 内核函数求出向量 $d\_MA_i$ 对应观测矩阵 $\Psi$ 每一列元素绝对值的和,具体步骤如下:

(5d1.a)在GPU中开辟 $N/V$ 个线程块和长度为 $N$ 的向量 $d\_sum$ ,每个线程块中设有 $M/K$ 个线程,其中向量 $d\_sum = [d\_sum_1, \dots, d\_sum_q, \dots, d\_sum_N] \in \mathbb{R}^N$ , $V$ 要满足 $V$ 对 $N$ 取余值为0, $V$ 取值为2~4可以达到理想的性能, $\mathbb{R}$ 表示实数域, $1 \leq q \leq N, 1 \leq V \leq N$ ;

(5d1.b)用第 $z$ 个线程块读取 $M/K \times V$ 个元素,即向量 $d\_MA_i$ 中第 $M/K \times V \times (z-1) + 1$ 到第 $M/K \times V \times z$ 的元素;在该线程块中分 $V$ 次取出这些元素,每次按顺序取 $M/K$ 个元素,依次分配给线程1~线程 $M/K$ ,每个线程计算各自元素的模值;将这 $M/K$ 个模值累加给向量 $d\_sum$ 的第 $d\_sum_{V \times (z-1) + p}$ 个元素,其中 $1 \leq z \leq N/V, 1 \leq p \leq V$ ;

(5d2)循环结束后,求出向量 $d\_sum$ 的二范数,取倒数即得到迭代步长 $\mu$ ;

(5e)将回波向量 $b$ 、向量化矩阵 $d\_MA$ 和迭代步长 $\mu$ 带入快速迭代收缩阈值算法中,经过多次梯度下降和快速阈值收缩过程,直到目标向量满足迭代终止条件,得到重建观测场景散射强度 $X^*$ ;

(6)将重建场景散射强度 $X^*$ 拷贝到主机端内存取模值,并按照先行后列的顺序排列成一

个二维矩阵,得到重建的超声图像。

2.如权利要求1所述的用GPU实现压缩感知超声成像方法,其特征在于,所述步骤(2)中,频点取 $k_t$ 时的回波声场强度矩阵 $\Psi_t$ ,通过下式计算:

$$\Psi_t = \begin{bmatrix} k_t^2 A_m(k_t) [e^{-jk_t \vec{e}_g \vec{r}_1} \cdot G(\vec{r}_{A1} - \vec{r}_1), \dots, e^{-jk_t \vec{e}_g \vec{r}_j} \cdot G(\vec{r}_{A1} - \vec{r}_j), \dots, e^{-jk_t \vec{e}_g \vec{r}_N} \cdot G(\vec{r}_{A1} - \vec{r}_N)] \\ \dots \\ k_t^2 A_m(k_t) [e^{-jk_t \vec{e}_g \vec{r}_1} \cdot G(\vec{r}_{Am} - \vec{r}_1), \dots, e^{-jk_t \vec{e}_g \vec{r}_j} \cdot G(\vec{r}_{Am} - \vec{r}_j), \dots, e^{-jk_t \vec{e}_g \vec{r}_N} \cdot G(\vec{r}_{Am} - \vec{r}_N)] \\ \dots \\ k_t^2 A_m(k_t) [e^{-jk_t \vec{e}_g \vec{r}_1} \cdot G(\vec{r}_{Aa} - \vec{r}_1), \dots, e^{-jk_t \vec{e}_g \vec{r}_j} \cdot G(\vec{r}_{Aa} - \vec{r}_j), \dots, e^{-jk_t \vec{e}_g \vec{r}_N} \cdot G(\vec{r}_{Aa} - \vec{r}_N)] \end{bmatrix} \quad (2)$$

其中, $A_{in}(k_t)$ 表示超声宽带脉冲发射信号在频点取值为 $k_t$ 时的幅度; $e^{-jk_t \vec{e}_g \vec{r}_j}$ 表示超声宽带脉冲发射信号在频点取值为 $k_t$ 时离散二维探测区域中各个像素点返回的相位, $\vec{e}_g$ 表示超声宽带脉冲发射信号方位角的单位向量,指定为轴向方向; $\vec{r}_j$ 表示超声宽带脉冲发射信号从超声线阵轴心位置到离散二维探测区域各个像素点距离的向量;

$G(\vec{r}_{Am} - \vec{r}_j)$ 表示格林函数, $\vec{r}_{A,m}$ 表示超声线阵轴心位置到到超声线阵各个阵元距离的向量, $1 \leq t \leq W, 1 \leq j \leq N, 1 \leq m \leq A$ 。

3.如权利要求1所述的用GPU实现压缩感知超声成像方法,其特征在于,所述步骤(5d1)中,常量 $V$ 的取值要满足 $V$ 对 $N$ 取余值为0, $V$ 取值为2~4。

4.如权利要求1所述的用GPU实现压缩感知超声成像方法,其特征在于,所述步骤(5e),按如下步骤进行:

(5e1)将回波向量 $b$ 、向量化矩阵 $d\_MA$ 和迭代步长 $\mu$ 带入下式,更新梯度下降序列 $u_n$ :

$$u_n = y_n - \mu \times \text{cublas}(d\_MA^H \cdot (\text{cublas}(d\_MA \cdot y_n) - b)) \quad (3)$$

其中 $y_n$ 是快速收缩向量,初始值为0,长度为 $N$ ;  $d\_MA^H$ 表示向量化矩阵 $d\_MA$ 的共轭转置; $\mu$ 为迭代步长; $\text{cublas}$ 是CUDA线性代数库CUBLAS Library中所支持的库函数,它可以执行向量化矩阵与向量的乘法运算;

(5e2)将梯度下降序列 $u_n$ 带入下式,更新目标场景散射强度 $x_n$ :

$$x_n = S_\Gamma(u_n) \quad (4)$$

其中 $S_\Gamma$ 为阈值函数:

$$S_\Gamma(u_n) = \begin{cases} \text{sign}(u_n)(|u_n| - \Gamma) & |u_n| > \Gamma \\ 0 & \text{其它} \end{cases}$$

其中 $\Gamma$ 为阈值, $\Gamma = \lambda \mu$ , $\lambda$ 取值在 $2e^4 \sim 4e^4$ 之间, $e$ 表示科学计数,取值为10; $\text{sign}()$ 表示取符号函数;

(5e3)判断收敛条件 $\|x_n - x_{n-1}\|_2 < \varepsilon$ 是否成立:

若成立,则停止计算,重建观测场景散射强度 $X^* = x_n$ ;

若不成立,令 $n = n + 1$ ,更新快速收缩向量 $y_n$ 为:

$$y_n = x_{n-1} + (x_{n-2} - x_{n-1}) \times (1 - t_1) / t_2 \quad (5)$$

其中 $\|x_n - x_{n-1}\|_2 = \sqrt{\sum_{j=1}^N (x_n[j] - x_{n-1}[j])^2}$ ,当 $n=1$ 时, $x_0=0$ ;  $j$ 表示系数向量 $x_n$ 和 $x_{n-1}$ 中的第 $j$

个元素,  $x_n[j]$ 表示向量 $x_n$ 的第 $j$ 个元素的值,  $x_{n-1}[j]$ 表示向量 $x_{n-1}$ 的第 $j$ 个元素值;  $t_1$ 、 $t_2$ 是两个数值不同的加速因子,  $t_1$ 的初始值为1,  $t_2 = 0.5 + 0.5 \times \sqrt{1 + 4 \times t_1 \times t_1}$ ; 将 $t_1$ 更新为 $t_1 = t_2$ , 返回步骤(5e1)。

5. 如权利要求4所述的用GPU实现压缩感知超声成像方法, 其特征在于, 所述步骤(5e2)中, 更新目标向量 $x_n$ 的计算是在内核函数kernel中完成的, 即 $x_n = S_{\Gamma}(u_n)$ 通过使用多线程进行并行计算得到。

## 用GPU实现压缩感知超声成像的方法

### 技术领域

[0001] 本发明属于图像处理技术领域,特别涉及一种用GPU实现快速成像的方法,可用于B超成像。

### 背景技术

[0002] 医学超声成像经过60余年的发展,它具有相对安全、实时性好、无创、便携、价格低廉等优点,其与X射线诊断技术、电子计算机断层扫描CT成像技术、核磁共振成像技术一起称为现代医学四大影像技术,已令亿万患者受益。

[0003] 但是超声成像仍然存在一些不足,如分辨率不高,多为毫米级;受噪声干扰严重,图像质量较差;实时性一般。

[0004] 近年来,在信号处理领域兴起的压缩感知理论吸引了诸多学者的关注,该理论指出,只要信号在某一个空间 $\Psi$ 上具有稀疏性,就可以利用观测矩阵对其以远低于奈奎斯特采样速率进行观测,进而利用优化手段高概率地从混叠观测中重构原信号,这使得传感器的采样成本大大降低。而且通过恰当选择空间 $\Psi$ ,信号的稀疏性越强,精确恢复原信号的可能性就越大,这样就会在提高图像分辨率、抑制噪声方面有出色的表现。从近几年国内外发表的文献来看,对压缩感知理论的研究已经涉及众多领域如压缩感知CS雷达成像、医学图像处理、光谱分析、遥感图像处理等,具有非常广阔的应用前景。

[0005] 由于病灶区域与正常组织的密度特征有明显差别,可认为超声图像在空间域是稀疏的,将压缩感知理论应用到超声成像可以较好的解决超声成像分辨率不高和噪声干扰严重的问题,但是压缩感知理论的问题在于重建过程中观测矩阵维度巨大、导致计算复杂度高,图像的重建时间过长。

[0006] 针对这个问题,以色列学者A.Beck等人在论文“A fast iterative shrinkage-thresholding algorithm for linear inverse problems”SIAM J.IMAGE SCIENCES, Vol.2.No.1,pp.183-202,2009中提出了快速迭代收缩阈值算法,简称FISTA。利用该算法,超声成像的基本框架可以表示为:

[0007]  $X^* = \arg \min_x \{ \|\Psi x - b\|_2^2 + \lambda \|x\|_1 \}$ ,其中, $X^*$ 为重建观测场景散射强度, $x$ 为目标向量, $b$

是采样后超声阵元接收的回波数据, $\lambda$ 为正则化参数, $\|\Psi x - b\|_2^2$ 表示向量 $\Psi x - b$ 二范数的平方, $\|x\|_1$ 表示向量 $x$ 的一范数。

[0008] 上述FISTA算法虽然复杂度低,适合求解大规模矩阵的重建问题,同时具备全局收敛性,收敛速度快,使得迭代时间大大缩短。但是,在面对数量级为千兆GB的数据时,仍然需要数分钟到半小时才能恢复一幅图像,这并不能满足对超声成像的实时性要求。

### 发明内容

[0009] 本发明的目的在于针对上述已有技术的不足,提出一种用GPU实现压缩感知超声成像的方法,以缩短压缩感知理论应用于超声成像时的重建时间,满足超声成像的实时性

要求。

[0010] 本发明的技术方案是这样实现的：

[0011] 一.技术原理

[0012] 2007年,英伟达NVIDIA®公司提出了统一计算设备架构CUDA,使图形处理器GPU具备了容易上手的快速并行计算能力,它采用单指令多线程SIMT的编程模式,具备数百倍乃至上千倍CPU核心数的CUDA核心数,如英伟达精视™系列显卡GTX770拥有1536个CUDA核心,单精度的计算峰值达到3.6TFLOPs。本发明首先建立起压缩感知超声成像的数学模型;再将CUDA技术应用到FISTA算法的重建,在重建之前改进了原FISTA算法中迭代步长的计算方式并实现了并行处理。通过采用矩阵分块思想和CUDA流技术减少数据交换时的延迟,采用指令级并行操作集ILP实现一个线程处理多组数据,调用线性代数函数库实现矩阵乘法运算的性能最优,最终实现了快速成像。

[0013] 二、技术方案

[0014] 根据上述原理,本发明的实现步骤如下:

[0015] (1)将超声探测区域二维离散化,得到N个离散化的像素点,其中 $N=T \times S$ ,T表示轴向像素的个数,S表示侧向像素的个数;

[0016] (2)将超声宽带脉冲发射信号在频域均匀采样得到W个频点,按频点顺序依次对已离散化的二维探测区域进行一次平面波扫描,每次扫描得到一个长度为A的局部观测回波向量 $b_t$ ,并将这W个局部观测回波向量按从上到下顺序合成长度为 $M=A \times W$ 的观测回波向量 $b=\{b_1, \dots, b_t, \dots, b_w\}$ ,同时保存由这W个频点产生的回波声场强度矩阵 $\Psi_1, \dots, \Psi_t, \dots, \Psi_w$ ,其中,A表示超声线阵的阵元个数,矩阵 $\Psi_t$ 的宽度为A,长度为N, $1 \leq t \leq W$ ;

[0017] (3)将回波声场强度矩阵 $\Psi_1, \dots, \Psi_t, \dots, \Psi_w$ ,按照从上到下顺序排列成一个大小为 $M \times N$ 的观测矩阵 $\Psi$ ;将离散化的二维探测区域按照行优先的顺序排列成一个目标向量x;

[0018] (4)根据回波向量b和观测矩阵 $\Psi$ 定义基于压缩感知的超声成像数学模型:

$$[0019] \quad X^* = \arg \min_x \{ \|\Psi x - b\|_2^2 + \lambda \|x\|_1 \} \quad (1)$$

[0020] 其中 $X^*$ 为重建观测场景散射强度,x为目标向量, $\lambda$ 为正则化参数, $\|\Psi x - b\|_2^2$ 表示向量 $\Psi x - b$ 二范数的平方, $\|x\|_1$ 表示目标向量x的一范数;

[0021] (5)在GPU中对上述数学模型进行求解,得到重建观测场景散射强度 $X^*$ ;

[0022] (5a)初始化: $n=0, \epsilon=10^{-3}$ ,其中n表示第n次迭代, $\epsilon$ 表示迭代终止条件;

[0023] (5b)将观测矩阵 $\Psi$ 按行均匀等分成宽度为 $M/K$ ,长度为N的K个子矩阵 $\Psi = \{\Psi_{s1}, \dots, \Psi_{si}, \dots, \Psi_{sK}\}$ ,要保证每个子矩阵数据量小于显存的容量上限, $1 \leq i \leq K$ ;

[0024] (5c)将每个子矩阵 $\Psi_{si}$ 以列优先顺序向量化为一维行向量 $d\_MA_i$ 并采用流操作拷贝到显存中,此时观测矩阵 $\Psi$ 转换成一个宽度为K,长度为 $M/K \times N$ 的向量化矩阵 $d\_MA = \{d\_MA_1, \dots, d\_MA_i, \dots, d\_MA_K\}$ ,其中观测矩阵 $\Psi$ 与向量化矩阵它们存在一一对应的关系;将回波向量b等分成长度为 $M/K$ 的K块, $b = \{b_1, \dots, b_i, \dots, b_K\}$ ,并拷贝到显存中;

[0025] (5d)根据向量化矩阵 $d\_MA$ 计算基于梯度下降算法的迭代步长 $\mu$ ;

[0026] (5d1)在GPU中建立一个重复执行K次的循环,每次循环创建一个 $k\_step$ 内核函数,求出向量 $d\_MA_i$ 对应观测矩阵 $\Psi$ 每一列元素模值的和;

- [0027] (5d2)循环结束后,求出向量 $d\_sum$ 的二范数,取倒数即得到迭代步长 $\mu$ ;
- [0028] (5e)将回波向量 $b$ 、向量化矩阵 $d\_MA$ 和迭代步长 $\mu$ 带入快速迭代收缩阈值算法中,经过多次梯度下降和快速阈值收缩过程,直到目标向量满足迭代终止条件,得到重建观测场景散射强度 $X^*$ ;
- [0029] (6)将重建场景散射强度 $X^*$ 拷贝到主机端内存取模值,并按照先行后列的顺序排列成一个二维矩阵,得到重建的超声图像。
- [0030] 本发明与现有技术相比有如下优点:
- [0031] 本发明是基于CUDA架构下的GPU并行计算超声成像过程,通过优化F1STA算法、对观测矩阵分块划分和充分利用GPU中线程块的性能,将成像速度提升了180倍,重建时间由现在的分钟级减少至毫秒级,达到了准实时的超声成像。

### 附图说明

- [0032] 图1是本发明的实现流程图;
- [0033] 图2是本发明对探测区域离散化的示意图;
- [0034] 图3是本发明使用的超声阵列与探测区域位置示意图;
- [0035] 图4是本发明将回波数据排列成列向量的示意图;
- [0036] 图5是本发明中的矩阵分块示意图;
- [0037] 图6是本发明中进行快速迭代收缩阈值算法的子流程图;
- [0038] 图7是本发明中计算核函数kernel的子流程图;
- [0039] 图8是用本发明仿真得到的重建图像与原始图像的对比图。

### 具体实施方式

- [0040] 本发明采用CUDA语言,可以在NVIDIA®任何一款支持CUDA架构的GPU设备上实现,建议选择计算能力3.0以上的显卡,计算能力越强,对显卡优化越好。为满足大规模矩阵向量乘法的要求,建议显存容量在2GB以上,可以搭载多块GPU实现显存容量高于观测矩阵的大小,这样可以减少数据交换带来的延迟。
- [0041] 以下结合附图对本发明作进一步详细描述。
- [0042] 参照图1,本发明的实现步骤如下:
- [0043] 步骤一:对探测区域进行离散化。
- [0044] 本发明对探测区域离散化采用等间隔采样,得到离散化的二维探测区域 $N=T \times S$ ,如图2所示,其中 $N$ 表示离散化总像素点数, $T$ 表示轴向像素点数, $S$ 表示侧向像素点数。
- [0045] 步骤二:对二维离散化的探测区域进行宽带脉冲平面波扫描,得到回波向量 $b$ 和观测矩阵 $\Psi$ 。
- [0046] 具体实现如下:
- [0047] (2a)根据图3所示的超声阵列与探测区域的相对位置建立直角坐标系 $(x, y)$ ,其中 $x$ 表示侧向, $y$ 表示轴向;
- [0048] (2b)在直角坐标系中,将超声阵列固定在轴向坐标为零的位置,即 $y=0$ ,且将阵列中心与探测区域中心对齐,超声阵列的长度为 $(A-1) \times d$ ,则第1个阵元的横坐标 $x_1$ 为:

$$[0049] \quad x_l = (1 - \frac{A+1}{2}) \times d,$$

[0050] 其中A为超声线阵的阵元个数,d为相邻两个阵元之间的间隔, $1 \leq l \leq A$ ;

[0051] (2c)将超声宽带脉冲发射信号在频域均匀采样得到W个频点,按频点顺序依次对已离散化的二维探测区域进行一次平面波扫描,每次扫描得到一个长度为A的局部观测回波向量 $b_t$ , $1 \leq t \leq W$ ,并将这W个局部观测回波向量按图4所示从上到下顺序合成长度为 $M=A \times W$ 的观测回波向量 $b = \{b_1, \dots, b_t, \dots, b_w\}$ ;

[0052] (2d)计算探测区域中各个像素点在第m个超声线阵阵元处产生的声场强度 $D(\vec{r}_j, k_t, \vec{r}_{A,m})$ :

$$[0053] \quad D(\vec{r}_j, k_t, \vec{r}_{A,m}) = k_t^2 A_m(k_t) e^{-jk_t \vec{e}_g \cdot \vec{r}_j} G(\vec{r}_{A,m} - \vec{r}_j),$$

[0054] 其中 $A_{in}(k_t)$ 表示超声宽带脉冲发射信号在频点取值为 $k_t$ 时的幅度; $e^{-jk_t \vec{e}_g \cdot \vec{r}_j}$ 表示超声宽带脉冲发射信号在频点取值为 $k_t$ 时离散二维探测区域中各个像素点返回的相位, $\vec{e}_g$ 表示超声宽带脉冲发射信号方位角的单位向量,指定为轴向方向; $\vec{r}_j$ 表示超声宽带脉冲发射信号从超声线阵轴心位置到离散二维探测区域各个像素点距离的向量; $G(\vec{r}_{A,m} - \vec{r}_j)$ 表示格林函数, $\vec{r}_{A,m}$ 表示超声线阵轴心位置到到超声线阵各个阵元距离的向量, $1 \leq m \leq A$ ;

[0055] (2e)由所述声场强度 $D(\vec{r}_j, k_t, \vec{r}_{A,m})$ 计算出当频点为 $k_t$ 时,宽度为A,长度为N回波声场强度矩阵 $\Psi_t$ :

$$[0056] \quad \Psi_t = \begin{bmatrix} k_t^2 A_m(k_t) [e^{-jk_t \vec{e}_g \cdot \vec{r}_1} \cdot G(\vec{r}_{A,1} - \vec{r}_1), \dots, e^{-jk_t \vec{e}_g \cdot \vec{r}_j} \cdot G(\vec{r}_{A,1} - \vec{r}_j), \dots, e^{-jk_t \vec{e}_g \cdot \vec{r}_N} \cdot G(\vec{r}_{A,1} - \vec{r}_N)] \\ \dots \\ k_t^2 A_m(k_t) [e^{-jk_t \vec{e}_g \cdot \vec{r}_1} \cdot G(\vec{r}_{A,m} - \vec{r}_1), \dots, e^{-jk_t \vec{e}_g \cdot \vec{r}_j} \cdot G(\vec{r}_{A,m} - \vec{r}_j), \dots, e^{-jk_t \vec{e}_g \cdot \vec{r}_N} \cdot G(\vec{r}_{A,m} - \vec{r}_N)] \\ \dots \\ k_t^2 A_m(k_t) [e^{-jk_t \vec{e}_g \cdot \vec{r}_1} \cdot G(\vec{r}_{A,A} - \vec{r}_1), \dots, e^{-jk_t \vec{e}_g \cdot \vec{r}_j} \cdot G(\vec{r}_{A,A} - \vec{r}_j), \dots, e^{-jk_t \vec{e}_g \cdot \vec{r}_N} \cdot G(\vec{r}_{A,A} - \vec{r}_N)] \end{bmatrix},$$

[0057] 其中 $1 \leq t \leq W$ , $1 \leq j \leq N$ , $1 \leq m \leq A$ ;

[0058] (2f)将回波声场强度矩阵 $\Psi_1, \dots, \Psi_t, \dots, \Psi_w$ 按照从上到下顺序排列成一个大大小为 $M \times N$ 的观测矩阵 $\Psi$ ;

[0059] (2g)将离散化的二维探测区域按照行优先顺序排列成一个长度为N的目标向量 $x$ 。

[0060] 步骤三:构建基于压缩感知的超声成像数学模型。

$$[0061] \quad X^* = \arg \min_x \{ \|\Psi x - b\|_2^2 + \lambda \|x\|_1 \},$$

[0062] 其中 $X^*$ 为重建观测场景散射强度, $x$ 为目标向量, $\lambda$ 为正则化参数, $\|\Psi x - b\|_2^2$ 表示向量 $\Psi x - b$ 二范数的平方, $\|x\|_1$ 表示目标向量 $x$ 的一范数。

[0063] 步骤四:在GPU中对上述数学模型进行求解,得到重建观测场景散射强度 $X^*$ 。

[0064] 具体实现如下:

[0065] (4a)初始化: $n=0$ , $\varepsilon=10^{-3}$ ,其中 $n$ 表示第 $n$ 次迭代, $\varepsilon$ 表示迭代终止条件;

[0066] (4b)如图5所示,将观测矩阵 $\Psi$ 按行均匀等分成宽度为 $M/K$ ,长度为 $N$ 的 $K$ 个子矩阵,

$\Psi = \{\Psi_{s1}, \dots, \Psi_{si}, \dots, \Psi_{sk}\}$ , 要保证每个子矩阵的数据量小于显存的容量上限,  $1 \leq i \leq K$ ;

[0067] (4c)将子矩阵  $\Psi_{si}$  以列优先顺序向量化为一维行向量  $d\_MA_i$  并拷贝到显存中, 此时观测矩阵  $\Psi$  转换成一个宽度为  $K$ , 长度为  $M/K \times N$  的向量化矩阵  $d\_MA = \{d\_MA_1, \dots, d\_MA_i, \dots, d\_MA_k\}$ , 它们存在一一对应的关系; 将回波向量  $b$  等分成长度为  $M/K$  的  $K$  块,  $b = \{b_1, \dots, b_i, \dots, b_k\}$ , 拷贝到显存中;

[0068] (4d)根据向量化矩阵  $d\_MA$  计算基于梯度下降算法的迭代步长  $\mu$ ;

[0069] (4d1)在GPU中建立一个重复执行  $K$  次的循环, 每次循环创建一个  $k\_step$  内核函数求出向量  $d\_MA_i$  对应观测矩阵  $\Psi$  每一列元素绝对值的和, 具体步骤如下:

[0070] (4d1.a)在GPU中开辟  $N/V$  个线程块和长度为  $N$  的向量  $d\_sum$ , 每个线程块中设有  $M/K$  个线程, 其中向量  $d\_sum = [d\_sum_1, \dots, d\_sum_q, \dots, d\_sum_N] \in \mathbb{R}^N$ ,  $V$  要满足  $V$  对  $N$  取余值为  $0$ ,  $V$  取值为  $2 \sim 4$  可以达到理想的性能,  $\mathbb{R}$  表示实数域,  $1 \leq q \leq N$ ,  $1 \leq V \leq N$ ;

[0071] (4d1.b)用第  $z$  个线程块读取  $M/K \times V$  个元素, 即向量  $d\_MA_i$  中第  $M/K \times V \times (z-1) + 1$  到第  $M/K \times V \times z$  的元素; 在该线程块中分  $V$  次取出这些元素, 每次按顺序取  $M/K$  个元素, 依次分配给线程  $1 \sim$  线程  $M/K$ , 每个线程计算各自元素的模值; 将这  $M/K$  个模值累加给向量  $d\_sum$  的第  $d\_sum \times (z-1) + p$  个元素, 其中  $1 \leq z \leq N/V$ ,  $1 \leq p \leq V$ ;

[0072] (4d2)循环结束后, 求出向量  $d\_sum$  的二范数, 取倒数即得到迭代步长  $\mu$ 。

[0073] (4e)将回波  $b$ 、观测矩阵  $\Psi$  和迭代步长  $\mu$  带入快速迭代收缩阈值算法中, 进行梯度下降和快速阈值收缩, 得到重建观测场景散射强度  $X^*$ 。

[0074] 参见图6, 本发明步骤具体步骤如下:

[0075] (4e1)将回波向量  $b$ 、向量化矩阵  $d\_MA$  和迭代步长  $\mu$  带入下式, 更新梯度下降序列  $u_n$ :

$$[0076] \quad u_n = y_n - \mu \times \text{cublas}(d\_MA^H \cdot (\text{cublas}(d\_MA \cdot y_n) - b)),$$

[0077] 其中  $u_n = [u_1, \dots, u_j, \dots, u_N] \in \mathbb{C}^N$ ,  $\mathbb{C}$  表示复数域;  $y_n$  是快速收缩变量, 初始值为  $0$ , 长度为  $N$ ;  $d\_MA^H$  表示向量化矩阵  $d\_MA$  的共轭转置;  $\mu$  为迭代步长;  $\text{cublas}$  是CUDA线性代数库 CUBLASLibrary 中所支持的库函数, 它可以执行矩阵与向量乘法运算,  $1 \leq j \leq N$ ;

[0078] (4e2)参见图7, 在kernel内核函数中开辟  $N$  个线程, 将梯度下降序列  $u_n$  带入kernel内核函数, 用第  $j$  个线程对梯度下降序列  $u_n$  的元素  $u_j$  进行阈值操作, 得到目标元素  $x_j$ :  $x_j = S_r(u_j)$ , 其中  $S_r$  为阈值函数:

$$[0079] \quad S_r(u_j) = \begin{cases} \text{sign}(u_j)(|u_j| - \Gamma) & |u_j| > \Gamma \\ 0 & \text{其它} \end{cases},$$

[0080] 其中  $\Gamma$  为阈值,  $\Gamma = \lambda\mu$ ,  $\lambda$  取值在  $2e^4 \sim 4e^4$  之间,  $e$  表示科学计数, 取值为  $10$ ;  $\text{sign}()$  表示取符号函数,  $1 \leq j \leq N$ ;

[0081] 将目标元素  $x_1, \dots, x_j, \dots, x_N$  组合为目标向量  $x_n$ ,  $x_n = [x_1, \dots, x_j, \dots, x_N] \in \mathbb{C}^N$ ;

[0082] (4e3)判断收敛条件  $\|x_n - x_{n-1}\|_2 < \varepsilon$  是否成立:

[0083] 若成立, 则停止计算, 重建观测场景散射强度  $X^* = x_n$ ;

[0084] 若不成立, 令  $n = n + 1$ , 更新快速收缩变量  $y_n$  为:

[0085]  $y_n = x_{n-1} + (x_{n-2} - x_{n-1}) \times (1 - t_1) / t_2$ ,

[0086] 其中  $\|x_n - x_{n-1}\|_2 = \sqrt{\sum_{j=1}^N (x_n[j] - x_{n-1}[j])^2}$ , 当  $n=1$  时,  $x_0=0$ ;  $j$  表示系数向量  $x_n$  和  $x_{n-1}$  中的第  $j$  个元素,  $x_n[j]$  表示向量  $x_n$  的第  $j$  个元素的值,  $x_{n-1}[j]$  表示向量  $x_{n-1}$  的第  $j$  个元素值;

[0087] 其中  $t_1$ 、 $t_2$  是两个数值不同的加速因子,  $t_1$  初始值为 1,  $t_2 = 0.5 + 0.5 \times \sqrt{1 + 4 \times t_1 \times t_1}$ ; 将  $t_1$  更新为  $t_1 = t_2$ , 返回步骤(4e1)。

[0088] 步骤五: 将重建场景散射强度  $X^*$  拷贝到主机端内存取模值, 并按照先行后列的顺序排列成一个二维矩阵, 即得到重建的超声图像。

[0089] 本发明的效果可以通过以下仿真实验说明:

[0090] 1. 仿真内容

[0091] 为了验证本发明在 CUDA 架构下并行计算的优越性, 通过一组仿真实验对本发明方法所需的时间与 CPU 串行所需的时间定量分析, 仿真测试平台为:

[0092] CPU: Intel Core i7 4770, 内存: 16GB

[0093] 显卡: NVIDIA GeForce GTX770, 显卡显存: 4GB, 计算能力 3.0

[0094] 软件平台: windows7 64位操作系统, VS2010 x64编译器, CUDA版本: 5.5

[0095] 2. 仿真结果及分析

[0096] 仿真实验用到两组数据, 第一组数据观测矩阵大小为 1GB, 图像大小为  $128 \times 128$ , 目标点个数为 20 个; 第二组数据观测矩阵大小为 8GB, 图像大小为  $256 \times 512$ , 目标点个数为 100 个。图像的分辨率为 100um, 要高于传统超声成像分辨率, 迭代次数设定为 30 次, 第一组仿真得到的重建图像与原始图像的对比如图 8 所示。

[0097] 图 8 表明, CUDA 架构下 GPU 的重建图像与原始图像在主观上评价基本一致, 点目标的位置与原始图像完全吻合, 从而达到了较高的重建效果。

[0098] 将 CUDA 架构下 GPU 并行计算时间和 CPU 串行时间进行对比, 如表 1 和表 2 所示:

[0099] 表 1 观测矩阵为 1GB 重建时间

[0100]

图像大小	计算方法	算法执行时间	GPU 加速比
128×128	CPU 串行	69.195s	179 倍
	GPU 并行	0.386s	

[0101] 从表 1 看出, 本发明方法对超声成像的重构速度提高明显, 达到了每秒 2.5 帧图像, 基本上解决了压缩感知框架下处理大规模矩阵耗时较长的问题, 实现了准实时性。

[0102] 表 2 观测矩阵为 8GB 重建时间

[0103]

图像大小	计算方法	算法执行时间	GPU 加速比
256×512	CPU 串行	551.983s	13 倍
	GPU 并行	44.139s	

[0104] 表2中,GPU的加速效果比表1中的加速比差,其主要原因在于显存不足导致每次迭代都需要进行数据交换,将数据从主机端内存交换到显存端。尽管如此,GPU加速处理后仍然比CPU串行快13倍,而且通过分析数据可以发现在CPU+GPU的方案中数据交换的时间占到总时间的92%,因此,若可以利用提升硬件解决显存不足的问题,加速比仍然可以达到180倍左右。

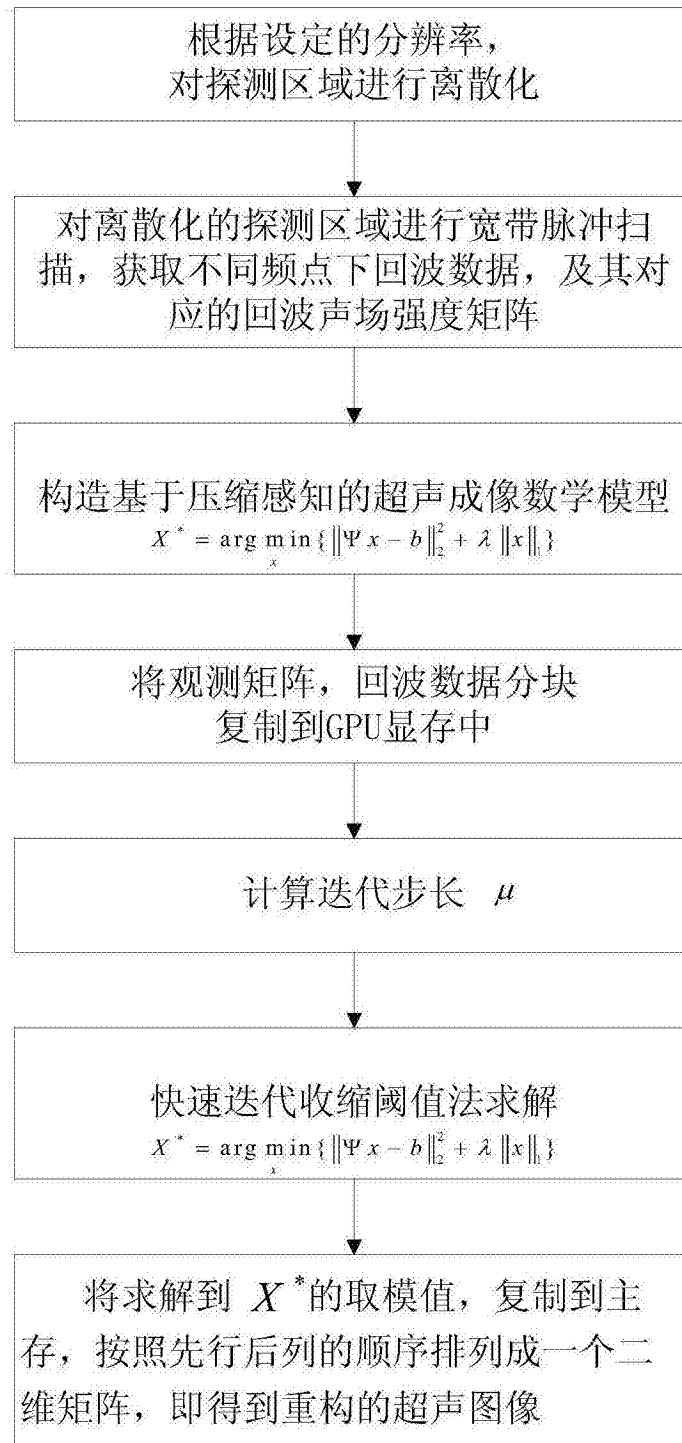


图1

空间采样

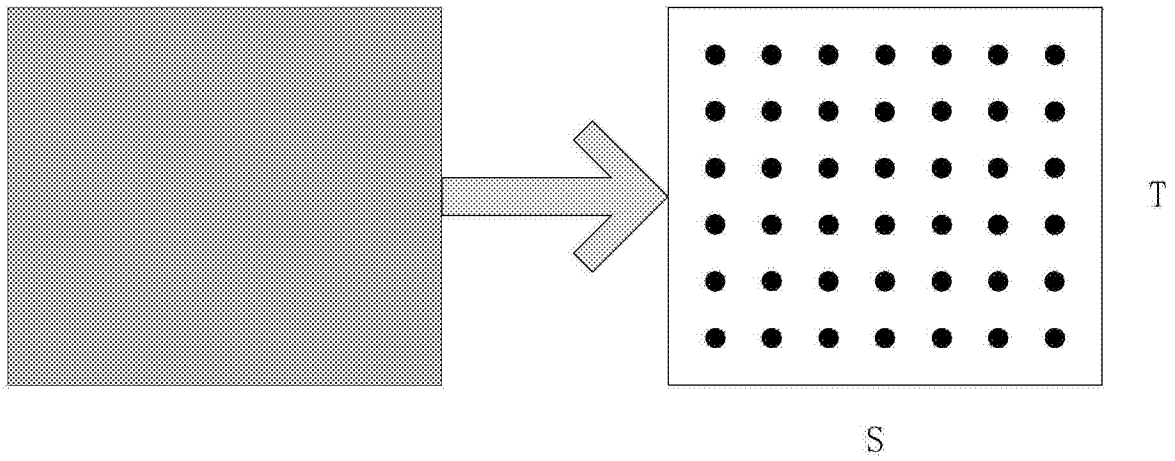


图2

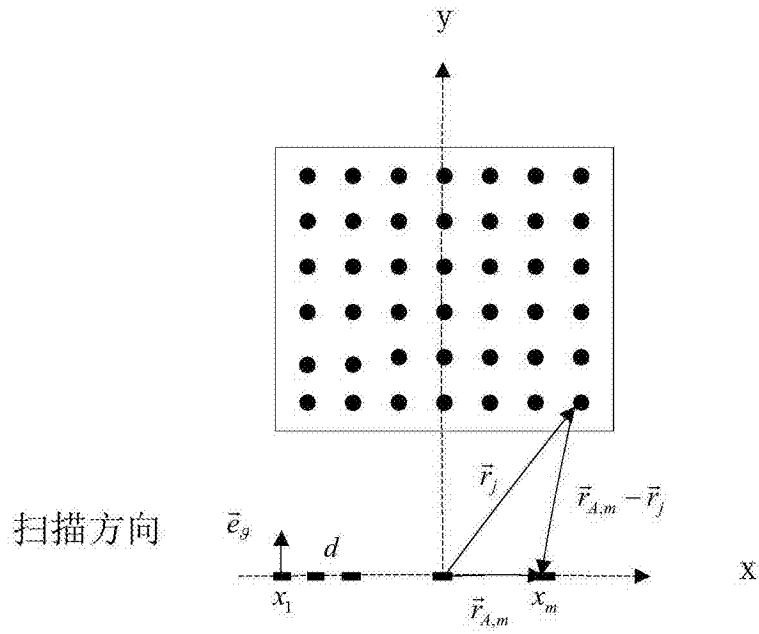


图3

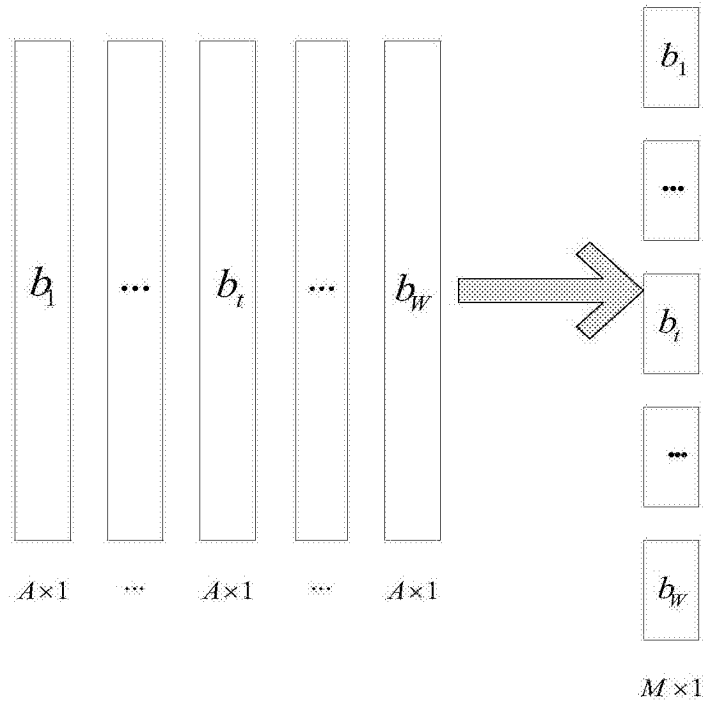


图4

矩阵分块

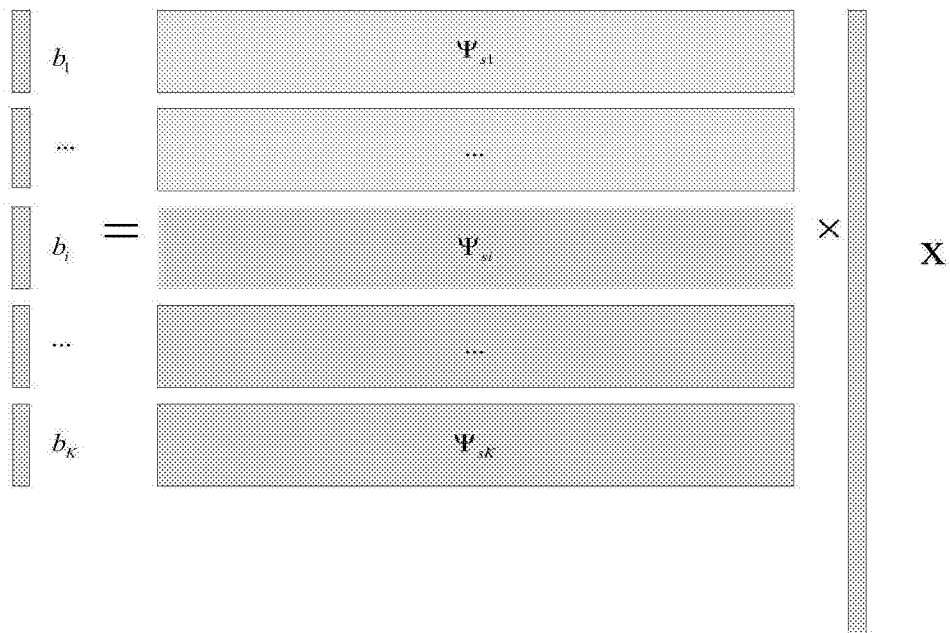


图5

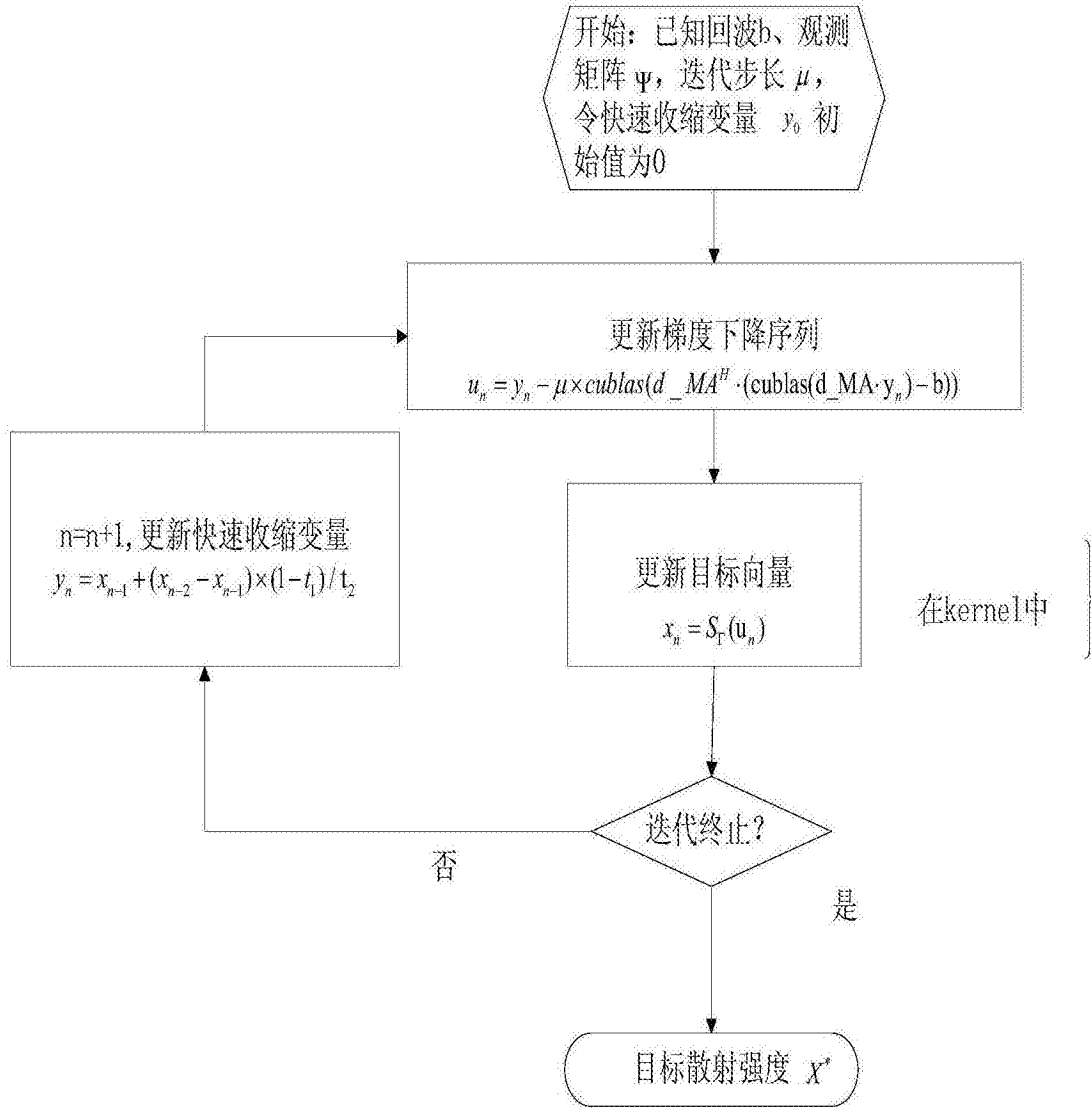


图6

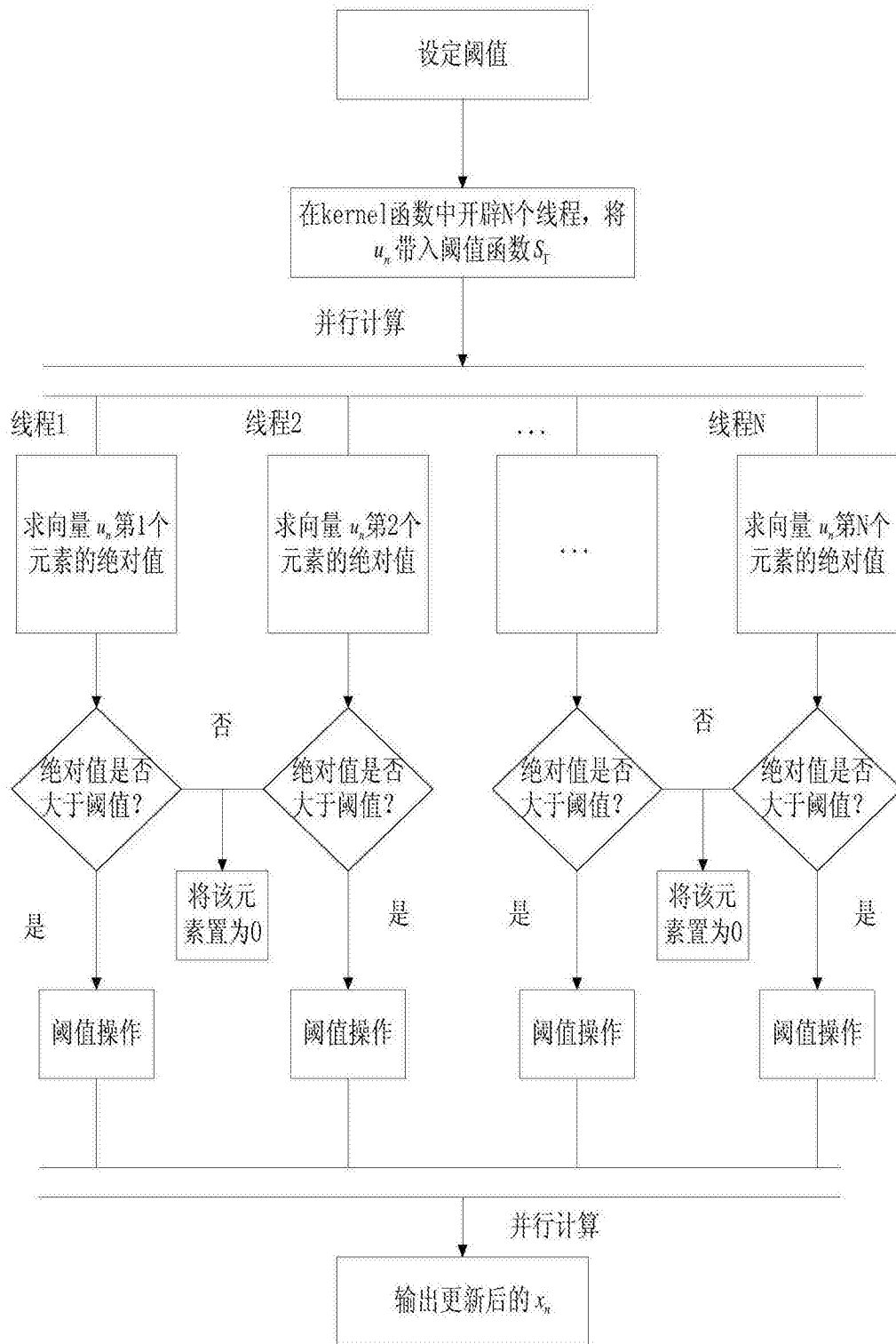


图7

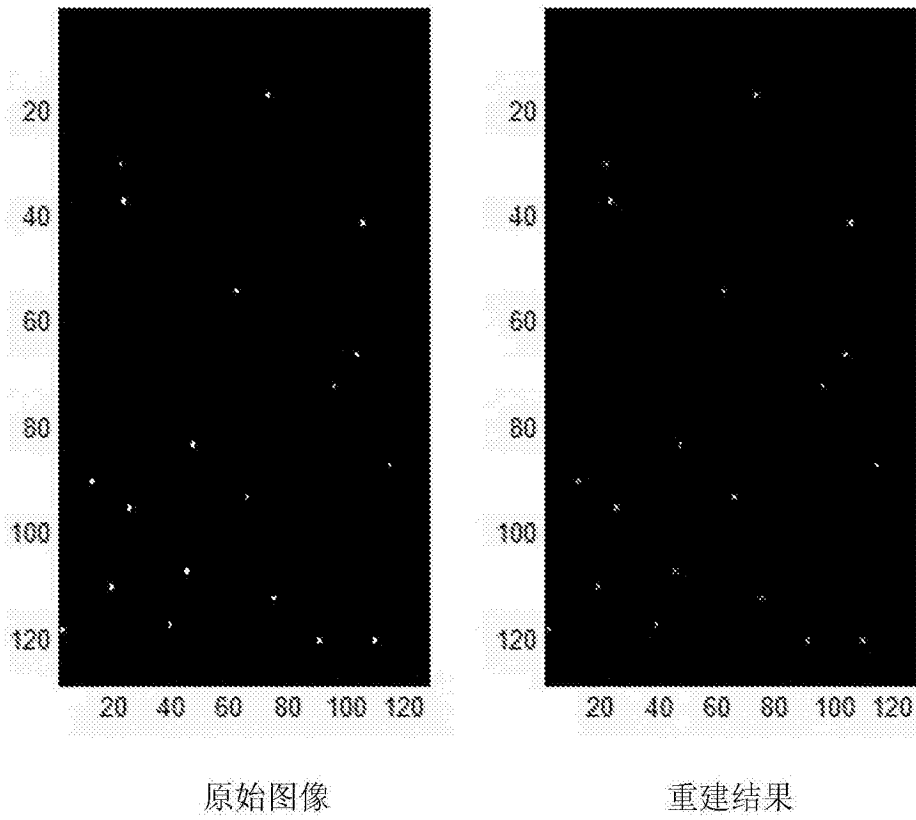


图8

专利名称(译)	用GPU实现压缩感知超声成像的方法		
公开(公告)号	<a href="#">CN104306022B</a>	公开(公告)日	2016-05-25
申请号	CN201410578176.5	申请日	2014-10-24
[标]申请(专利权)人(译)	西安电子科技大学		
申请(专利权)人(译)	西安电子科技大学		
当前申请(专利权)人(译)	西安电子科技大学		
[标]发明人	林杰 韩亭玉 贺玉高 石光明		
发明人	林杰 韩亭玉 贺玉高 石光明		
IPC分类号	A61B8/00		
CPC分类号	A61B8/14 A61B8/4488 A61B8/5207 A61B8/5215		
代理人(译)	朱卫星		
审查员(译)	孙晓彤		
其他公开文献	CN104306022A		
外部链接	<a href="#">Espacenet</a> <a href="#">SIPO</a>		

摘要(译)

本发明公开了一种用GPU实现压缩感知超声成像的方法，主要解决压缩感知理论框架下成像重建时间较慢的问题。其实现步骤为：1.根据设定的分辨率，对探测区域进行离散化并对该区域进行宽带脉冲扫描，得到回波向量和观测矩阵，进而建立超声成像数学模型；2.将回波向量与观测矩阵分块并复制到GPU显存中；3.在GPU中计算迭代步长；4.将迭代步长带入快速迭代收缩阈值算法求解出重建观测场景散射强度；5.将该散射强度复制到主存中取模值并排列成一个二维矩阵，得到重建的超声图像。本发明相对传统的快速迭代收缩阈值算法，重建时间从分钟级别降低到毫秒级别，极大提高了超声成像的实时性，可用于超声实时处理领域。

