

# 一种新的基于 GPU 实现的锥束 CT 正投影算法

赵 星<sup>1</sup>, 胡晶晶<sup>2</sup>, 潘晓川<sup>3</sup>, 张 朋<sup>1</sup>

(1. 首都师范大学数学学院, 北京 100037; 2. 北京理工大学软件学院, 北京 100081;  
3. 芝加哥大学放射学系, 美国芝加哥 60637)

**摘 要:** 锥束 CT 成像算法的正投影计算量大, 消耗时间长. 为此, 本文提出了一种基于 GPU 实现的正投影加速算法. 该算法在 GPU 的可编程管线中, 通过纹理映射方法实现了圆轨迹锥束扫描模式下正投影计算. 由于投影逐片并行计算, 因此该算法具有较高的计算速率, 且支持全浮点运算精度. 在该正投影算法的基础上, 本文还给出了三种算法优化方法. 这些优化方法进一步提高了算法的执行效率. 通过对 Shepp-logan 模型的正投影计算以及应用于迭代法进行三维图像重建等实验验证了本文算法的优点.

**关键词:** 计算机断层成像; 图形处理器; 图像重建; 正投影

**中图分类号:** TP391 **文献标识码:** A **文章编号:** 0372-2112 (2009) 06-1165-05

## A Novel GPU Based Cone Beam CT Forward Projection Method

ZHAO Xing<sup>1</sup>, HU Jing-jing<sup>2</sup>, PAN Xiao-chuan<sup>3</sup>, ZHANG Peng<sup>1</sup>

(1. School of Mathematical Sciences, Capital Normal University, Beijing 100037, China;  
2. School of Software, Beijing Institute of Technology, Beijing 100081, China;  
3. Department of Radiology, the University of Chicago, Chicago, IL 60637, USA)

**Abstract:** The forward projection of cone-beam CT image reconstruction algorithms requires both excessive computation and time consuming. A novel CT forward projection method based on graphics processing unit is proposed for accelerating the forward projection calculation. The method achieves the forward projection calculation in circular cone-beam scanning mode by using texture mapping techniques in GPU programmable pipeline. Because projections are calculated slice by slice in parallel, this method has the advantages of higher computational efficiency and calculation at full floating point precision. Three optimization methods are also presented for further improving the execution efficiency of the method. The advantages of this method are verified by the experiments of the projection calculation of Shepp-logan phantom and 3D image reconstruction with iterative image reconstruction algorithm.

**Key words:** computerized tomography; graphics processing unit; image reconstruction; forward projection

## 1 引言

如何提高图像重建速度是当前锥束 CT 研制中亟待解决的问题. 研究人员一方面通过改进原 CT 重建算法的执行效率来提高图像重建速度; 另一方面着力研究硬件加速的图像重建算法. 如 Cabral 等<sup>[1]</sup>和 Mueller 等<sup>[2]</sup>先后分别利用 SGI 图形工作站的纹理映射硬件实现了 CT 重建算法的加速. Li 等<sup>[3]</sup>基于 PC 机集群实现了锥束 CT 并行迭代重建算法. 商业 CT 机中多采用专用集成电路和可编程逻辑阵列来加速 CT 算法实现. 这些硬件设备虽然具有惊人的计算速度, 但都具有一定的局限性<sup>[4]</sup>. 相比于这些硬件设备, PC 机的图形处理器 (Graphics Processing Unit, GPU) 具有价格低廉、运算快捷、操作简单、更新换代快等优点<sup>[5]</sup>. 由于 CT 的数据扫描过程与

计算机图形学的投影过程本质上是一致的, 因此 GPU 非常适合用于 CT 图像加速重建. 基于 GPU 加速的 CT 重建算法已成为目前 CT 领域的一个研究热点<sup>[4~6]</sup>.

CT 图像重建算法从数学上可分为解析算法和迭代算法两类. 无论解析算法还是迭代算法, 其中的投影运算 (解析算法仅包含反投影, 迭代算法同时包含正投影和反投影) 都是最耗时的运算部分, 所耗时间远超过其他计算步骤. 因此提高 CT 重建算法的运算效率, 关键在于提高投影的计算速度. Mueller 等<sup>[2,7]</sup>基于 GPU 固定管线 (fixed pipeline) 用二维纹理映射技术实现了正投影过程的并行计算; 基于 GPU 可编程管线 (programmable pipeline) 用投射纹理技术实现了反投影过程的并行计算. 由此而实现了 CT 迭代算法 (如 SART 和 OS-EM 算法等) 和解析算法 (如 FDK 算法) 的加速, 取得了很高的图

像重建加速比.

尽管在 GPU 加速 CT 重建算法方面已取得以上研究成果,但受到当前 GPU 固定管线性能的局限,基于固定管线实现的正投影算法仍然有如下不足:

(1) 由于 GPU 固定管线使用不灵活,用户无法通过编写 GPU 程序调整顶点和像素的运算过程,因此在进行诸如“投影累加”、“权值计算”等计算时,还需要增加额外的计算步骤<sup>[2,7]</sup>.

(2) 目前普通 PC 机显卡上的 GPU 固定管线普遍仅支持 8bit 整型计算.因此基于固定管线实现的正投影过程的计算精度低,难以满足用于低衬度诊断的图像重建要求<sup>[5]</sup>.

针对以上问题,本文提出了一种新的正投影实现算法.该算法在 GPU 的可编程管线中实现,不仅支持 32bit 浮点运算精度,而且需要的渲染次数少,运算速度快.在该正投影算法的基础上,本文还给出了一些进一步提高本文正投影算法执行效率的优化方法.

## 2 基于 CPU 实现的 CT 正投影算法简介

在 CT 中,入射 X 射线强度与出射 X 射线强度之比经过对数运算后,表示沿着 X 射线路径上被测物体线性衰减系数的线积分. CT 的正投影计算就是求沿 X 射线路径上线性衰减系数的线积分.由于物体对给定能量光子的线性衰减系数正比于物体的密度值,因此在不混淆的情况下,在本文以下部分中称线性衰减系数为密度值.在传统的基于 CPU 实现的 CT 重建算法中,物体被离散化成规则的网格,每个网格单元称为一个体素(voxel).设对一个包含  $N \times N \times N$  个体素的被检测物体作正投影,在每个投影角度下得到  $M \times M$  个投影数据,在一个角度共得到  $M \times M \times$  个投影数据.由被检测物体的体素求投影数据的正投影计算可用公式表达如下:

$$p_i = \sum_{j=0}^{N-1} (w_{ij} \cdot v_j) \quad (1)$$

公式(1)中  $v_j$  表示被检测物体的第  $j$  个体素的密度值,  $0 < j < N \times N \times N$ ;  $p_i$  表示第  $i$  个投影数据(即沿第  $i$  条射线密度值的累加),  $0 < i < M \times M \times$ ;  $w_{ij}$  表示被测物体的第  $j$  个体素对第  $i$  投影数据  $p_i$  的贡献权值.在锥束扫描方式下,当参数  $N$ 、 $M$  和(较大时,由公式(1)可知正投影的计算量很大.

## 3 锥束 CT 正投影的 GPU 加速实现算法

### 3.1 基于切片的体表示模型

本文采用基于切片的体表示模型作为被测物体的离散模型.该模型通过定义空间网格点上的值来定义被测物体,每个空间网格点上的值表示物体在该点处的密度值.沿着某个坐标轴方向将这些离散点划分成一系列

切片.射线穿过被测物体时与这些切片相交,交点值由其周围邻近的四个离散点值通过双线性插值得到.根据射线与各切片的交点值以及在切片间的前进步长,便可用梯形法求出沿着射线对物体密度值的积分.

例如用  $N \times N \times N$  个空间网格点表示一个被测物体,这些点被划分为  $N$  片切片,每片切片包含  $N \times N$  个网格点.我们将公式(1)所示的传统正投影公式近似为公式(2)所示的正投影计算.

$$p_i = \sum_{j=0}^{N-1} (w_{ij} \cdot v_j) \quad l_i \cdot \sum_{k=0}^{N-1} v_{ik} \quad (2)$$

公式(2)中,  $k$  表示切片的序号;  $v_{ik}$  表示第  $i$  条射线穿过第  $k$  个切片时的交点处的密度值,由其周围四个邻近点密度值插值求出. ( $l_i$  表示第  $i$  条射线在切片间的前进步长,相当于公式(1)的正投影权值.在固定的扫描角度下,射线  $i$  的步长  $l_i$  在任意两个相邻切片之间都是等长的,因此只依赖于射线  $i$ ,与切片  $k$  无关.

### 3.2 基于 GPU 的正投影算法

根据被测物体的切片模型,可将组成物体的每个切片都设为一个二维纹理,二维纹理中的纹素与切片中网格点一一对应.利用 GPU 纹理映射技术,通过将一个切片纹理映射到平板探测器上,就实现了该切片中所有网格点的并行正投影计算.

锥束 CT 数据采集时的射线源、被测物体和平板探测器的几何关系如图 1 所示.设探测器四个顶点在被测物体某片切片纹理空间中的对应点分别是  $s_1$ 、 $s_2$ 、 $s_3$  和  $s_4$ ,下面推导求这四点坐标的并行计算公式.我们用公式(3)描述锥束 CT 的投影关系.公式(3)中,向量  $v$  表示所求四个点  $s_1$ 、 $s_2$ 、 $s_3$  和  $s_4$  中某一点在被测物体坐标系( $X_v$ - $Y_v$ - $Z_v$ )中的坐标,由齐次坐标  $v = (x_v, y_v, z_v, 1)^T$  给定.矩阵  $R$  为绕  $Y_v$  轴顺时针将被测物体坐标系旋转(度)的旋转矩阵.矩阵  $T$  为将被测物体坐标系分别沿着  $X_v$ 、 $Y_v$ 、 $Z_v$  轴方向平移  $-T_x$ 、 $-T_y$ 、 $-T_z$  距离的平移矩阵.经过矩阵  $R$  和  $T$  的变换,将  $v$  的坐标从被测物体坐标系变换到了射线源坐标系( $X_s$ - $Y_s$ - $Z_s$ ).被测物体坐标系的移动与射线源坐标系的移动互逆.如矩阵  $R$  也可看作是逆时针将射线源坐标系旋转了(度).矩阵  $R$  和  $T$  的

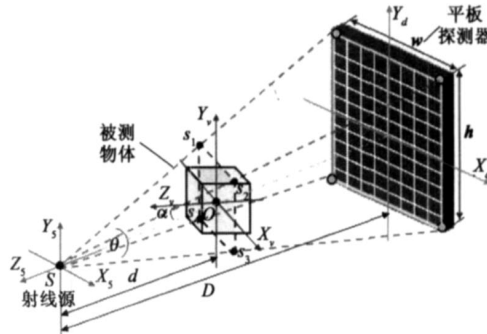


图1 锥束CT投影几何关系示意图

组合能够描述锥束 CT 大多数扫描模式中的射线源轨迹。

式(3)中矩阵  $P$  为透视投影矩阵,用于定义锥束投影方式下视景体的形状和方向,由射线源位置和探测器的宽  $w$  和高  $h$  确定。矩阵  $P$  中参数  $D$  为射线源  $S$  到探测器面的距离。参数  $n$  和  $f$  分别是视景体的近裁剪面和远裁剪面距射线源点的距离。透视投影将视景体映射为一个坐标在  $-1$  到  $1$  之间的正方体裁剪空间。

$$P \times T \times R \times v = v_p$$

$$\begin{bmatrix} 2D & 0 & 0 & 0 \\ w & & & \\ 0 & 2D & 0 & 0 \\ h & & & \\ 0 & 0 & -\frac{(f+n)}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & -T_y \\ 0 & 0 & 1 & -T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix} = \begin{bmatrix} x_p \cdot w_p \\ y_p \cdot w_p \\ z_p \cdot w_p \\ w_p \end{bmatrix} \quad (3)$$

经过矩阵  $R$ 、 $T$  和  $P$  的变换,被测物体坐标系中的点  $v$  被变换成为裁剪空间中的一点  $v_p$ 。 $v_p$  也用齐次坐标表示,由于其  $w_p$  尚未确定,因此将该点齐次坐标表示成  $v_p = (x_p \cdot w_p, y_p \cdot w_p, z_p \cdot w_p, w_p)^T$ 。

解公式(3)中的方程得公式(4)。根据公式(4),只需将  $(x_p, y_p)$  分别设为  $(-1, +1)$ ,  $(+1, +1)$ ,  $(+1, -1)$  和  $(-1, -1)$  传递到 GPU 的顶点处理器中,就可并行求出  $s_1$ 、 $s_2$ 、 $s_3$  和  $s_4$  的坐标  $(x_v, y_v)$ 。参数  $w_p$  为某一点到射线源所在  $X_s Y_s$  平面的距离,用于透视校正。

$$x_v = \frac{z_v \cdot (2D \sin - x_p \cdot w \cos) + x_p \cdot T_x \cdot w + 2D \cdot T_x}{2D \cos + x_p \cdot \sin \cdot w}$$

$$w_p = -\sin \cdot x_v - \cos \cdot z_v + T_z$$

$$y_v = \frac{y_p \cdot w_p \cdot h}{2D} + T_y \quad (4)$$

求出平板探测器顶点在切片平面上所对应的坐标  $(x_v, y_v)$  后,还需要进行纹理坐标转换以得到所对应的纹理坐标  $(x_t, y_t)$ 。当用矩形纹理表示切片时,设切片的宽和高分别为  $s_w$  和  $s_h$ ,切片中横向和竖向排列的网格点个数分别为  $N_w$  和  $N_h$ ,则求纹理坐标  $(x_t, y_t)$  的公式为:

$$\begin{cases} x_t = \left\{ \frac{x_v}{s_w} + 0.5 \right\} \cdot N_w \\ y_t = \left\{ \frac{y_v}{s_h} + 0.5 \right\} \cdot N_h \end{cases} \quad (5)$$

根据以上纹理坐标计算原理,基于 GPU 的正投影算法总结如下:

(1) 传递一个各顶点坐标分别为  $(-1, +1)$ ,  $(+1, +1)$ ,  $(+1, -1)$  和  $(-1, -1)$  的表示探测器的四边形到

GPU 的顶点处理器中,利用公式(4)和(5)以 SIMD 方式并行计算得到探测器各顶点在切片纹理空间上所对应纹理坐标  $(x_t, y_t)$  及参数  $w_p$ 。根据参数  $w_p$  进行透视校正。

(2) 在 GPU 光栅化器中,根据探测器的采样率,自动生成与探测器单元一一对应的栅格状排列的像素,并对探测器四个顶点所对应的纹理坐标和透视校正参数进行均匀线性插值,插值出每个像素对应的纹理坐标和透视校正参数。

(3) 在 GPU 像素处理器中,通过作透视除得到正确的纹理坐标。然后利用这些纹理坐标在切片纹理中采样取值,并赋值给相应的表示探测器单元的像素,就得到了该切片的正投影图像。

(4) 通过以上经由 GPU 管线各阶段的一次渲染计算,就得到了一片切片的正投影图像。对被测物体的每片切片重复以上计算,得到各切片的正投影图像。在每次得到一片切片的正投影图像后,随即在像素处理器中累加到一个输出纹理中,就得到了整个被测物体的正投影图像。

## 4 算法优化

### 4.1 基于多纹理的加速图像重建

图形学中的多纹理技术可以将多种纹理融合后同时映射到物体表面。该技术目前在 GPU 中被广泛支持。本文的正投影算法可利用多纹理技术加速计算。每次渲染中同时处理多个切片的投影,减少所需要的渲染次数。

### 4.2 基于 GPU 的 RGBA 管道的四对称投影并行计算

图形学中用 RGBA(即红、绿、蓝、透明度)四个参数的组合表示颜色。为了加速颜色计算,GPU 在硬件上被设计为一个包含 RGBA 四个颜色通道的四元向量处理器。GPU 的指令既可以只控制任意一个颜色通道作标量计算,也可以同时控制四个颜色通道作四元向量运算,且作向量运算和作标量运算的速度基本相同。

CT 在圆轨迹扫描时,位于正交方向上四个旋转角度(如  $0^\circ$ 、 $+90^\circ$ 、 $+180^\circ$ 、 $+270^\circ$ )处的投影几何关系相同。因此可以利用 GPU 的 RGBA 四颜色通道并行计算特性,一个颜色通道计算一个角度的投影,通过相同的投影计算过程,同时实现四个方向的投影计算。

### 4.3 数据快速传输

本文算法中采用 OpenGL 快速数据传输扩展 PBO (Pixel Buffer Object) 进行切片纹理数据和投影数据的传输。该数据传输扩展不仅能加快数据在内存和显存之间传输速度,而且支持数据传输和 GPU 计算同步进行。

## 5 数值实验结果与分析

Shepp-logan 模型是公认的标准 CT 算法测试模型之

一,我们通过对该模型作正投影计算以及应用于迭代法进行三维图像重建来测试本文算法的精度和速度. 本文采用的 Shepp-logan 模型如图 2 所示. 所有的实验结果均在配置为 1.86GHz Intel Xeon 双核 CPU、2GB 内存、NVIDIA 8800 GTX 图形卡、768MB 显存的 PC 机上运行得到. 设定的锥束 CT 数据采集模式为圆轨道采集模式, 以下数据类型都是 32bit 浮点型.

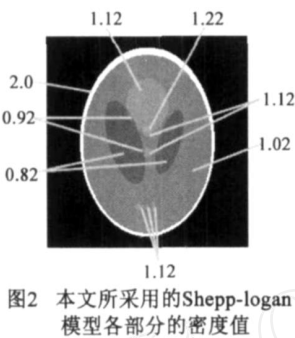


图2 本文所采用的Shepp-logan模型各部分的密度值

### 5.1 算法投影图像质量

本文分别用归一化均方根距离  $d$ 、归一化平均绝对距离  $r$  和最坏情况距离  $e$  度量图像质量. 我们比较了不同旋转角度下本文算法生成的正投影图像和解析计算出的投影图像. 例如在图 3 中给出了旋转角度为  $45^\circ$  时两种图像的对比. 在被测物体的采样率为  $256^3$ , 投影图像的采样率为  $256^2$  的情况下, 在  $45^\circ$  时三种图像距离分别是:  $d = 0.0081$ ,  $r = 0.0065$ ,  $e = 0.0319$ . 实验结果表明, 本文算法在各个旋转角度下生成的正投影图像的误差都很小. 即使在接近  $45^\circ$ 、 $135^\circ$ 、 $225^\circ$  以及  $315^\circ$  等角度时采样率逐渐降低, 图像误差略有增大, 但增幅微乎其微, 表

明本算法在各个旋转角度下都能生成高精度的投影图像. 在 GPU 中提供了邻近点插值和线性插值的硬件实现, 本文采用 GPU 的线性插值和传统 CPU 计算精度相当. 如采用邻近点插值, 则投影图像精度下降, 难以达到 CPU 下基于体素模型正投影的计算精度.

### 5.2 算法执行效率

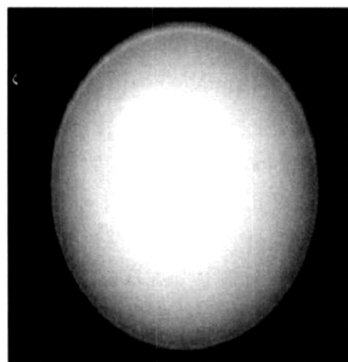
表 1 给出了本文算法和原 CPU 算法生成不同尺寸正投影图像时所耗时间的比较. 实验结果表明, 生成相同尺寸的正投影图像, 本文算法比原 CPU 算法快 50 倍以上. 对于采样率低的被测物体, 速度提高更明显. 基于投影几何关系的四对称性, 充分利用 GPU 处理 RGBA 四管道的指令级并行性, 算法速率大大提高. 表 1 中的运算时间包含数据从内存传输到显存的时间和在 GPU 中的计算时间. 实验表明利用 PBO 加速数据传输, 有助于算法整体运算速度的提高.

表1 算法速度测试

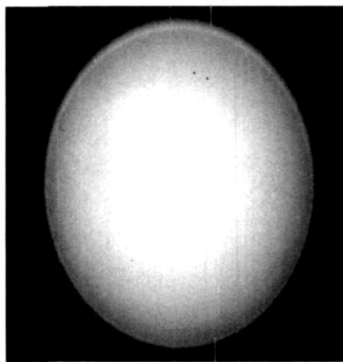
被测物体	投影角度 $\times$ 正投影图像	GPU 利用对称性	GPU 不利用对称性	CPU 算法
$256^3$	$360 \times 256^2$	2.1s	6.2s	6.7min
$512^3$	$360 \times 512^2$	26.0s	65.5s	57.3min

### 5.3 应用于迭代法加速图像重建

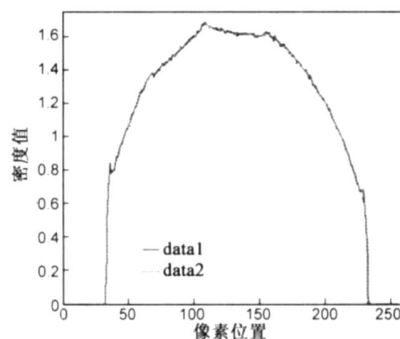
CT 图像的迭代重建算法可用于不同数据采集模式, 对不完全数据亦可重建图像. 还可以结合一些先验知识和约束条件进行求解, 重建图像质量较高. 典型的



(a)解析图像



(b)本文算法图像



(c)两种图像的第128行密度分布曲线比较

图3 在 $45^\circ$ 时正投影图像精度比较

迭代算法包括 SART<sup>[8]</sup> 和 ML-EM<sup>[9]</sup> 算法等. 迭代算法中正投影和反投影的运算量大, 使迭代法运算时间太长, 限制了迭代算法在实际系统中的应用. 我们利用本文正投影算法, 结合基于投射纹理实现的反投影算法<sup>[6]</sup>, 实现了完全驻留于 GPU 中的全浮点精度迭代算法 SART 和 OSEM. 图 4 给出了基于 GPU 的 SART 算法重建的 Shepp-Logan 模型图像与原始模型数据的比较. 所用的 3D Shepp-Logan 模型的密度对比度为 5%, 原始数据为 256 个角度的  $256 \times 256$  的投影数据, 重建图像的大小为  $256^3$ . 经过 15 次迭代, 重建图像与原始模型数据的归一化均方根距离为 0.0329, 归一化平均绝对距离为

0.0082, 最坏情况距离也仅为 0.1653, 表明将本算法应用于迭代法能实现较高精度的图像重建.

## 6 结束语

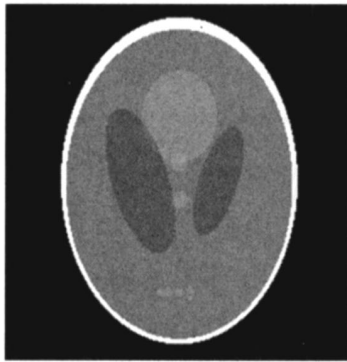
本文提出了一种在 GPU 可编程管线中实现的锥束 CT 正投影算法, 以及对该算法的多种优化方法. 与基于固定管线的正投影算法相比, 本文提出的正投影算法有如下优点:

(1) 在一次渲染中即可算出物体多个切片的正投影, 所需渲染次数少.

(2) 目前普通 PC 机显卡上的 GPU 固定管线普遍仅

支持 8bit 整型计算,而可编程管线大多支持 32bit 浮点计算,因此本文算法比基于固定管线的正投影算法计算精度高,适用范围广。

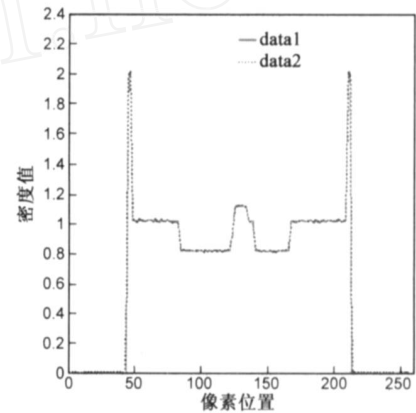
(3) 本算法在 GPU 顶点处理阶段和像素处理阶段,都可通过编写程序控制,因此更具有灵活性,可方便地在算法中添加一些先验知识和约束条件。如实现工程中依赖于旋转角度的投影几何关系 (angle-dependent projection geometry)<sup>[7]</sup>等。



(a)模型图像



(b)用SART算法重建的图像



(c)两个图像中第115行的密度分布曲线比较

图4 Shepp-Logan模型图像重建(其第96片切片图像)

#### 参考文献:

- [1] Cabral B, et al. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware[A]. Proceedings of the 1994 Symposium on Volume Visualization[C]. Virginia, United States: Tysons Corner, 1994. 91 - 98.
- [2] Mueller K, et al. Rapid 3D cone-beam reconstruction with SART by using texture mapping hardware [J]. IEEE Trans Med Imag, 2000, 19(12): 1227 - 1237.
- [3] Li X, et al. Parallel iterative cone beam CT image reconstruction on a PC cluster[J]. Journal of X-Ray Science and Technology, 2005, 13(2): 63 - 72.
- [4] Mueller K, et al. Why do commodity graphics hardware boards (GPUs) work so well for acceleration of computed tomography [A]. Proceedings of SPIE Electronic Imaging 2007 [C]. Pol-lak, Ilya: Computational Imaging V, 2007. 1 - 12.
- [5] Owens J, et al. GPGPU: General-Purpose Computation on Graphics Hardware[R]. San Diego: SIGGRAPH2007 Courses, 2007.
- [6] Zhao X, Bian J, Emil S. Y, et al. GPU-based 3D cone-beam CT image reconstruction: application to micro CT[A]. Nuclear Science Symposium Conference Record [C]. Hawaii: NSS apos, 2007. 3922 - 3925.
- [7] Xu F and Mueller K. Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware[J]. IEEE Trans Nuclear Sci, 2005, 52(3): 654 - 663.
- [8] Andersen A H, et al. Simultaneous Algebraic Reconstruction Technique (SART): a superior implementation of the ART al-

gorithm[J]. Ultrasonic Imaging, 1984, 6(2): 81 - 94.

通过对 Shepp-logan 模型的正投影计算以及应用于迭代法进行三维图像重建等实验,验证了本文算法以上优点。基于二维纹理的体表示模型在当旋转角度改变时,射线穿过切片之间的间隔会发生变化,造成物体采样率的不恒定。在将来的进一步研究中,还需研究怎样利用多片二维切片纹理,通过纹理取值的三线性插值,实现对物体的均匀采样,以提高正投影的精度。

- [9] Hudson H, et al. Accelerated image reconstruction using ordered subsets of projection data[J]. IEEE Trans of Medical Imaging, 1994, 13(4): 601 - 609.

#### 作者简介:



赵 星 男,1974 年生于山西省五台县。

2001 年于中国科技大学获计算机应用专业博士学位。现为首都师范大学检测成像实验室主任,副研究员,硕士生导师。主要研究领域为 CT 算法、虚拟植物、基于 GPU 的通用计算等。

E-mail: zhaoxing\_1999 @yahoo.com



胡晶晶 女,1978 年生于江苏省徐州市。

2005 年于北京理工大学获计算机应用专业博士学位。现为北京理工大学软件学院讲师。主要研究领域为人工智能、CT 成像算法等。

E-mail: hujingjing @bit.edu.cn

潘晓川 男,1960 年生于四川。1982 年北京大学物理系本科毕业,1991 年芝加哥大学物理系获取博士学位,现为芝加哥大学放射系教授。研究方向为成像物理和数学,信号处理和图像质量评估,以及它们在医学影像问题中的应用。

张 朋 男,1957 年生于陕西。1996 年于中科院数学所获博士学位。现为首都师范大学数学学院应用数学系主任,研究员。研究方向包括 CT 理论与算法、数学物理方程理论与数值计算、信号处理技术。