利用 GPU 实现单层螺旋 CT 的三维图像重建

张慧滔[1,2],于 平[3],胡修炎[1],张 朋[1]
(1.首都师范大学数学科学学院,北京 100048; 2.中国科学院高能物理研究所,北京 100049;
3.华北电力大学科技学院,河北保定 071051)

摘 要: CT 数据的获取过程和 CT 图像的重建过程与图形学的渲染过程极其相似,因此利用图形处理器(GPU)来加速 CT 重建算法成为了近年来 CT 研究的热点之一.本文根据单层螺旋 CT 数据的特点,构造了"平行-扇束"投影模式,实现了基于 GPU 的单层螺旋 CT 的三维图像重建算法.数值实验表明,与 CPU 上的分层重建相比重建速度提高 10 倍以上.

关键词: 单层螺旋 CT; 图形处理器; 三维重建; 投影模式
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2011) 01-0076-06

Single-Slice Helical CT Three-Dimensional Image Reconstruction Using GPU

ZHANG Hui-tao[1,2], YU Ping[3], HU Xiu-yan[1], ZHANG Peng[1]
(1. *School of Mathematical Sciences, Capital Normal University, Beijing* 100048, *China*;
2. *Institute of High Energy Physics, Chinese Academy of Sciences, Beijing* 100049, *China*;
3. *College of Science and Technology, North China Electric Power University, Baoding, Hebei* 071051, *China*)

Abstract: CT scanning and CT reconstruction are similar to rendering process of computer graphics, so using GPU to accelerate CT reconstruction algorithm become one of the focuses of the CT study in recent years. Based on features of Single-slice helical CT, we construct the parallel perspective projection model, and using GPU to accelerate the single-slice helical CT three-dimensional image reconstruction. Numerical experiments show that reconstruction efficiency improved by ten times than conventional algorithm.

Key words: single-slice helical CT; graphic processing unit; three-dimensional reconstruction; projection model

1 引言

CT 图像重建与计算机图形学的渲染过程非常相似.CT 理论中的平行束投影和锥束投影的概念分别与计算机图形学中的正交投影和透视投影的概念一致.CT 理论中的 X 光源和探测器分别对应于计算机图形学中的摄像机和渲染目标.CT 理论中的投影地址的概念也相应于图形学中纹理坐标的概念.由于图形处理器(GPU)利用专用的纹理映射硬件计算纹理坐标(投影地址),并能够并行处理多个被投影的数据,因此利用 GPU 一般能够将重建速度提高一到两个数量级.而近年来发展起来的 GPU 可编程功能为图形处理以外的通用计算提供了一个良好的运行平台,这使得利用 GPU 加速 CT 图像重建成为了 CT 研究的热点之一.

利用图形硬件加速 CT 重建算法已有十多年历史.早在 1994 年 Cabral 等人就利用 SGI Reality Engine Onyx 的图形硬件的纹理影射功能首先实现了滤波反投影(FBP)重建算法[1].1998 年 Klaus Mueller 和 Roni Yagel 引入多纹理技术,在 SGI Octane 工作站上实现了代数重建法(Algebraic Reconstruction methods, ART)和联合代数重建法(Simultaneous ART, SART)[2,3].取得了较好的加速效果.1998 年后期出现了 PC 机的图形处理器,主要代表为 NVIDIA TNT2, ATI Rage 和 3dfx Voodoo3.1999 年 nVidia 公司首次提出 GPU 的概念,并在 GeForce 256 中实现了硬件变换和光照计算.随着芯片技术的发展和强大的市场推动,近年来图形处理器(GPU)性能得到大幅度提高,尤其引人注意的是出现可编程特性,为 GPU 加速 CT 算法开辟了更广阔的空间.Childlow and Moller 在只有 8 位计算精度的 GeForce4 上实现了三维电子发射 CT 的的 OSEM 算法[4],累加工作在 CPU 上进行.Mueller and Yagel[5]利用高 8 位低 8 位分别存储在不同通道的准精度方法来提高计算的精度,计算的结果需要在 CPU 上

收稿日期:2009-09-15;修回日期:2010-07-07
基金项目:国家自然科学基金(No.60971131,No.60972140);北京市自然科学基金(No.3102009)

进行重新装配.GPU 对浮点运算的支持后,Xu F and Mueller K[6]利用 GPU 实现了 SART,OS-EM 和 FDK 三种常用算法.基于 GPU 加速的重建算法大都集中在锥束扫描和断层扫描的重建算法上.2009 年赵星等利用 GPU 的可编成管线实现了一种基于 GPU 的正投影加速算法[7],为实现基于 GPU 的迭代类三维重建算法奠定了基础.基于 GPU 加速的重建算法大多集中在锥束扫描和断层扫描的重建算法上.

本文研究如何利用 GPU 加速单层螺旋 CT 的图像重建.尽管基于面阵列探测器的锥束 CT 具有射线利用率高、扫描速度快等优点.但目前面阵列探测器与线阵列探测器相比射线转换效率低,尤其是探测高能 X 射线的面探测器技术还不成熟.因此,基于线阵列探测器的工业 CT 仍然是工业检测的主流设备.

2　单层螺旋 CT 扫描

2.1　扫描模式

利用线阵列探测器工业 CT 获得三维 CT 图像的常用扫描模式有两种:步进-采集模式(即多层圆轨迹扫描)和单层螺旋扫描模式.

步进-采集模式包括数据的获取阶段和非数据获取阶段.在数据的获取阶段,射线源和探测器静止,待测体旋转获取三代扫描数据;然后非数据获取阶段开始,关闭射线源,射线源和探测器同步移动到下一个扫描位置;依次获取多断层的三代扫描数据.步进-采集模式在扫描过程中,射线源,探测器,待测物需要加、减速运动.而我们知道加减速在机械运动中会带来很多麻烦,在重建中也会带来一系列图像伪影.

为了克服步进-采集模式的问题.螺旋扫描模式被提出.螺旋 CT 采集数据的扫描方式是待测体连续匀速旋转,射线源和探测器沿垂直方向同步匀速移动.也可以看作待测体静止不动,射线源和探测器绕待测体作螺旋运动.如图 1 所示.

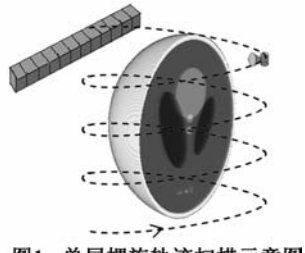


图1　单层螺旋轨迹扫描示意图

2.2　重建方法

步进-采集模式是在离散的位置获取切片的投影数据.重建切片与获取切片数据平面完全一致.采用传统断层的重建算法进行重建,对于非获取切片数据平面只能通过插值获得切片.螺旋扫描中没有优先平面,可以对任意位置平面进行重建.但是对于任何平面只有一个正常投影,其它投影数据为重建平面相邻的其它平面的投影数据.如果采用传统 CT 重建算法,把一个周期内的螺旋 CT 数据看作某个断层的投影数据进行重建,将产生螺旋状伪影.为减轻重建图象的伪影,一系列插值方法被提出[8~10].

本文采用临近采样周期的线性插值方法[8],其它插值方法可以类比实现.投影数据用 $p(u,\beta)$表示,其中 u 为探测器单元位置.β 为旋转角度.$p(u,\beta+2\pi)$表示与数据 $p(u,\beta)$差一个采用周期的投影数据.重建平面位于数据 $p(u,\beta+2\pi)$所在平面和数据 $p(u,\beta)$所在平面之间.x 表示重建平面和 $p(u,\beta)$所在平面之间的距离.h 螺旋扫描的螺距.$p'(u,\beta)$表示插值后的投影数据,

$$\bar{p} = p'(u,\beta) = \frac{h-x}{h}p(u,\beta) + \frac{x}{h}p(u,\beta+2\pi).$$

由于计算机图形学的纹理插值中的线性插值模式与上式相同.因此可利用 GPU 实现插值.

3　“平行-扇束”投影模式

单层螺旋 CT 数据获取扫描模式中每个断层的数据获取周期是完全相似的,因此可以看作有多个射线源和多个线阵探测器同时对被重建物体的不同断层进行圆轨迹扫描或螺旋扫描,当然每层的探测器只接受相应层的射线源所发出的射线.如图 2 所示.此扫描模式称之为单层螺旋 CT 扫描模式的等价模式.根据单层螺旋 CT 扫描模式的等价模式,本节构造了“平行-扇束”投影变换.

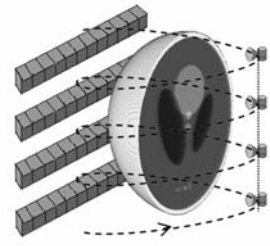


图2　单层螺旋轨迹等价CT扫描示意图

把三维物体变为二维图像的过程称为投影变换.根据视点到投影平面的距离不同,图形学中经常把投影分为平行投影和透视投影,平行投影中视点与投影平面之间的距离为无穷大,而对透视投影,这个距离是有限的.平行投影中常用的一种投影为正交投影,它与 CT 理论中的平行束扫描模式相对应;透视投影中常用的投影为单视点的透视投影,它与 CT 理论中的锥束扫描模式相对应.本节给出的“平行-扇束”投影模式与单层螺旋 CT 扫描模式的等价模式相对应.

“平行-扇束”投影模式在一个方向是透视投影,另一个方向是正交投影.与透视投影不同,“平行-扇束”投影中视点不是点,而是一条线段,用 l 表示.如图 3 所示,设空间坐标系$\{x,y,z\}$,O 为坐标原点,z 轴方向为视点方向;O'为 z 轴与投影平面的交点,坐标系$\{x',y'\}$为投影平面坐标系,O'为坐标原点,x'轴与 x 轴平行,y'轴与 y 轴平行.$\boldsymbol{x}_0$ 为取景空间中任意一点,用齐次坐标$[x_0,y_0,z_0,1]^{\mathrm{T}}$表示.$\boldsymbol{x}_0$ 与 l 确定一个平面 A,在平面 A 上过点 $\boldsymbol{x}_0$ 做 l 的垂线与投影平面交于点 $\boldsymbol{p}$ 坐标用$[p_x,$

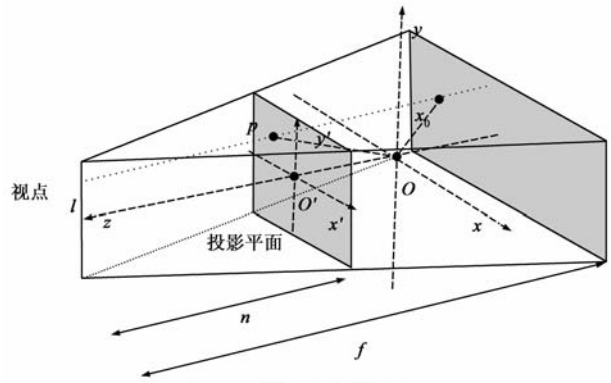


图3 “平行-扇束”投影

$p_y, p_z, p_w]^{\mathrm{T}}$ 表示. $\boldsymbol{p}$ 为 $\boldsymbol{x}_0$ 在投影平面上的投影点.

$$\boldsymbol{p}' = \boldsymbol{P} \otimes \boldsymbol{x}_0' \tag{1}$$

$\boldsymbol{P}$ 表示“平行-扇束”投影模式的投影矩阵.

$$\boldsymbol{P} = \begin{bmatrix} \frac{2n}{w} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

其中 n 表示视点到投影平面的距离,f 表示视点到后裁减平面的距离. w 表示投影平面的宽度,h 表示投影平面的高度.

因此式(1)即为:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ p_w \end{bmatrix} = \begin{bmatrix} \frac{2n}{w} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \times \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix}$$

由于“平行-扇束”投影模式在一个方向是透视投影,另一个方向是正交投影. 因此在“平行-扇束”投影模式中$[p_x, p_y, p_z, p_w]$不对应与二维坐标$(\frac{p_x}{p_w}, \frac{p_y}{p_w})$,而是对应二维坐标$(\frac{p_x}{p_w}, p_y)$. 因此在实现反投影算法时,不能简单的只在锥束重建算法的基础上改变投影矩阵利用投影纹理技术来实现.

4 GPU 实现方法

本节考虑由单层螺旋扫描数据重建 CT 图像的 GPU 实现方法. 实现了 GPU 上的反投影滤波型(BPF)算法[11,12],首先利用投影纹理技术实现投影数据导数反投影,获得导数反投影(differentiated backprojection,简称 DBP)的图像;然后对 DBP 图像滤波获得重建体数据. 传统算法中反投影时间占到重建总时间的 95% 左右,本文重点阐述反投影的实现步骤.

为了实现单层螺旋 CT 扫描的反投影,结合“平行-扇束”投影模式,本文对投影纹理[13]的方法做了两点改进:

(1)改进是纹理四通道对应得坐标分别计算,如图 4 所示.

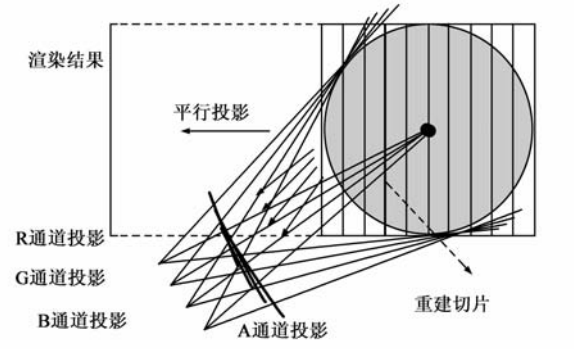


图4 改进的投影纹理的反投影过程

由于投影纹理的四个通道中存放的是不同角度的投影数据. 每个角度的模式变换矩阵是不同的因此顶点在四个角度下对应同一张纹理的四个通道下的坐标是不同的. 为了解决这个问题,Mueller 等利用圆轨迹扫描具有的对称性[6],将具有对称关系得四个角度下的投影存放在一张纹理中,反投影后,将渲染结果纹理返回 CPU 中,所对应得四个通道的数据根据对称性进行叠加,获得重建切片. 这种算法虽然利用了纹理四通道的性质,减少了渲染次数,但是由于需要将四个通道的计算结果传回 CPU 进行叠加. 因此加重了 GPU 到 CPU 的数据传输,而 GPU 到 CPU 的数据传输本身就是计算效率的瓶颈所在. 我们改进得投影纹理的方法是分别计算四个通道的纹理坐标. 这样渲染次数与 Muller[6]的方法相同,但是减少了 GPU 到 CPU 的数据传输,和 CPU 的叠加计算. 因此计算效率得到了一定提高. 当然由于对于螺旋扫描来说不具有四对称的性质. 因此也无法利用 Muller 的四对称方法.

(2)改进是根据“平行-扇束”投影的特点计算纹理坐标.

下面以一个通道为例阐述纹理坐标的计算方法. 在反投影过程中有两个关键操作:顶点程序中计算投影纹理坐标和片段程序中投影纹理查找. 投影纹理中不需要从应用程序明确指定每个顶点的纹理坐标. 而需要编写顶点程序自动地从物体空间的每个顶点位置计算纹理坐标. 变换公式为:

$$\textbf{TranCoord} = \textbf{OrthProjViewMatrix} * \textbf{ModelViewMatrix} * \textbf{Posion},$$

其中 Posion 为顶点坐标,用齐次坐标表示$[x_0, y_0, z_0, w_0]^{\mathrm{T}}$,**ModelViewMatrix** 为程序设定的某个角度下的模型变换矩阵,

ModelViewMatrix = **TranMatrix** * **RotaMatrix**, **TranMatrix** 为移动矩阵,**RotaMatrix** 旋转矩阵.

$$\mathbf{TranMatrix}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&\frac{h\times\alpha}{2\pi}\\0&0&0&1\end{bmatrix}$$

$$\mathbf{RotaMatrix}=\begin{bmatrix}\cos\alpha&-\sin\alpha&0&0\\\sin\alpha&\cos\alpha&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

h 为螺旋 CT 扫描时的螺距. 当 $h=0$ 时即为步进-采集模式. α 为旋转角度.

OrthProjViewMatrix 为"平行-扇束"投影变换矩阵, **OrthProjViewMatrix** = $\boldsymbol{P}$. **TranCoord** 是变换后顶点的投影坐标,用$[s,t,r,q]^{\mathrm{T}}$表示,$[s,t,r,q]^{\mathrm{T}}$不对应与二维坐标$(\frac{s}{q},\frac{t}{q})$,而是对应二维坐标$(\frac{p_x}{p_w},p_y)$. 因此顶点投影的齐次坐标表示为$[s,tq,r,q]^{\mathrm{T}}$.

计算的投影坐标在$[-1,1]$范围内,需要利用下式将其变换到纹理坐标内

$$\mathbf{TexCoord}=\mathbf{TexMatrix}*[s,tq,r,q]^{\mathrm{T}},$$

TexMatrix 是一个额外的矩阵,目的是经过缩放和偏移将位于$[-1,1]$范围的坐标变换到$[0,1]$.

$$\mathbf{TexMatrix}=\begin{bmatrix}\frac{1}{2}&0&0&\frac{1}{2}\\0&\frac{1}{2}&0&\frac{1}{2}\\0&0&\frac{1}{2}&\frac{1}{2}\\0&0&0&1\end{bmatrix}$$

如果投影纹理为矩形纹理,长宽为 width, height. 它的纹理坐标的范围为[0, width],[0, height],额外矩阵 **TexMatrix** 将位于$[-1,1]$范围的坐标变换到矩形纹理的坐标范围内. 因此

$$\mathbf{TexMatrix}=\begin{bmatrix}\frac{\mathrm{width}}{2}&0&0&\frac{\mathrm{width}}{2}\\0&\frac{\mathrm{height}}{2}&0&\frac{\mathrm{height}}{2}\\0&0&\frac{1}{2}&\frac{1}{2}\\0&0&0&1\end{bmatrix}.$$

经过一系列变换后 TexCoord 为重建切片四个顶点对应的纹理齐次坐标. 光栅化后,获得重建切层上每个重建点对应的纹理齐次坐标.

5　实验结果

本文采用 Visual Studio. Net 2003 作为开发平台,使用 GLSL 语言实现了以上算法. 算法测试的计算机配置为 Intel Xeon 5120(1.83GHz),4GB DDR2 内存, NVIDIA Qurdro FX4600 显卡(768MB 显存,显卡的核心频率 500MHz).

扫描过程中主视方向为 x 方向,俯视方向为 z 方向,侧视方向为 y 方向. 扫描模型为 Shepp-logan 模型. 参数如表 1 所示. 其中衰减系数是相对概念,有些椭球衰减系数为负值. 探测器为 512 个单元,扫描层数为 512 层,0 ~ 360°角度内采样 360 个投影. 射线源到转台中心的距离为 1910mm,转台中心到探测器的距离为 240mm,探测器单元尺寸为 0.127mm,螺距为 0.127mm. 待重建体的大小为 512 × 512 × 512. 重建 DBP 图像如图 5 所示.

(a) 主视方向x=256切层的DBP图像　(b) 俯视方向z=256切层的DBP图像　(c) 侧视方向y=256切层的DBP图像

图5　重建Shepp-logan模型的DBP图像

表 1　模型参数

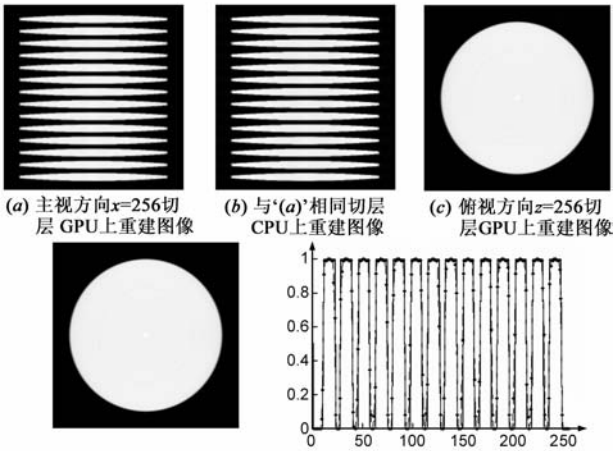
序数	x_c (mm)	y_c (mm)	z_c (mm)	a (mm)	b (mm)	c (mm)	$x-z$ 平面内转角(度)	衰减系数
1	0	0	0	10.53	10.53	13.16	0	1.00
2	0	0	0	9.55	9.55	12.19	0	-0.70
3	0	0	7.31	2.93	2.93	3.66	0	0.60
4	3.66	0	2.74	4.83	1.95	1.95	168	-0.9
5	-2.74	0	2.74	5.48	2.59	2.59	25.72	-0.45
6	0	0	3.66	0.73	0.73	0.73	0	0.50
7	0	0	0	0.73	0.73	0.73	0	0.50
8	1.46	0	-10.24	0.73	0.37	0.37	0	0.50
9	-1.46	0	-10.24	0.37	0.37	0.73	0	0.50
10	0	0	-10.24	0.37	0.37	0.37	0	0.50

重建体划分为横切片划分. 滤波后的重建体如图 6. 表 2 给出了投影数据分别为 256 × 256 × 360, 512 × 256 × 360, 1024 × 256 × 360 时,重建体分别为 256 × 256 × 256, 512 × 512 × 256, 1024 × 1024 × 256 时 GPU 上重建算法各阶段的耗时情况. 由表 2 可以看出 GPU 加速后的

算法与 CPU 上的重建算法相比,重建速度提高 10 倍以上,这是由于 CPU 上重建算法的主要计算量是计算投影地址;而 GPU 上计算投影地址的操作是由硬件完成的,计算速度很快,只占反投影部分计算时间的三分之一左右.由表 2 可以看出 GPU 加速后的算法反投影部分的计算时间占总时间的 60%以上,数据在显存和内存之间的传输时间占总时间的 30%左右.本文提出的改进投影纹理的方法可以有效的减少反投影部分的计算时间和数据在显存和内存之间的传输时间.改进投影纹理的方法是将四个投影角度下的投影数据放入 RGBA 四通道中,然后利用 GPU 的可编程特点,分别计算 RGBA 四通道的投影地址,四个通道同时渲染利用减少重建数据的赋值次数来提高重建速度.另外,由于避免了 4 倍重建数据的传输,因此缩短了数据在显存和内存之间的传输时间.图 7 是 GPU 加速后的三维重建算法和传统的 CPU 上的三维重建算法的重建结果比较,重建模型为 Dris 模型,可以看出加速后的重建效果与 CPU 上插值后的 FBP 重建效果相当.

表 2 GPU 重建与 CPU 重建性能的比较

重建体大小	投影数据尺寸	内存→显存(s)	投影数据求导	反投影耗时(s)	显存→内存(s)	GPU 重建共耗时(s)	CPU 重建时间(s)
256×256×256	(256×256)×360	0.406	0.109	1.641	0.328	2.484	63.095
512×512×256	(512×256)×360	0.600	0.531	5.094	1.984	8.209	131.443
1024×1024×256	(1024×256)×360	1.078	0.953	18.875	9.359	30.265	411.950



(*a*) 主视方向*x*=256切层 GPU上重建图像 (*b*) 与'(*a*)'相同切层 CPU上重建图像 (*c*) 俯视方向*z*=256切层GPU上重建图像

(*d*) 与'(*c*)'相同切层 CPU上重建图像 (*e*) 重建结果的比较,'--'为GPU加速的BPF重建数据;'*'为插值后FBP重建数据

图7 GPU和CPU重建结果比较

在重建过程中我们发现 GPU 的 float 精度只有小数点后两位的精度.因此为了保证计算精度,计算过程中首先将投影数据量化到(0.0,65536.0)区间.

6 结论

本文通过构造"平行-扇束"投影模式,改进投影纹理技术,在 GPU 上实现了单层螺旋 CT 数据的图像重建.数值实验表明本文提出的算法较传统的 CPU 下的算法,在计算效率上得到大幅度提高.

在反投影滤波算法的 CPU 实现中,滤波时间相对于反投影时间比较少,仅占总时间的 10%左右.但是利用 GPU 加速反投影后,反投影时间大幅缩短,滤波时间与加速后的反投影时间相比就相当可观了.由于本文主要研究利用 GPU 加速反投影重建.为了突出文章主题,有关 GPU 上滤波的算法将另文讨论.

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]. Tysons Corner, Virginia, USA, 1994. 91 – 98.

[2] Mueller K, et al. On the use of graphics hardware to accelerate algebraic reconstruction methods[A]. SPIE Medical Imaging Conference[C]. San Diego, California, USA, 1999, 3659(2): 615 – 625.

[3] Mueller K, et al. Rapid 3-D cone-beam reconstruction with the simultaneous algebraic reconstruction technique (SART) using 2-D texture mapping hardware[J]. IEEE Transactions on Medical Imaging, 2000, 19(12): 1227 – 1237.

[4] Chidlow K, Möller T. Rapid emission tomography reconstruction[A]. Proceedings Volume Graphics Workshop[C]. New York, UK, USA, 2003. 15 – 26.

[5] Mueller K, Xu F. Practical considerations for GPU-accelerated CT[A]. IEEE International Symposium on Biomedical Imaging [C]. Washington D C, USA. 2006. 1184 – 1187.

[6] Xu F, Mueller K. Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware[J]. IEEE Transaction nuclear science, 2005, 52(3): 654 – 663.

[7] 赵星,胡晶晶,潘晓川,张朋.一种新的基于 GPU 实现的锥束 CT 正投影算法[J].电子学报,2009,37(6):1165 – 1169.
Zhao Xing, Hu Jingjing, Pan Xiaochuan, Zhang Peng. A novel GPU based cone beam CT forward projection method[J]. Acta Electronica Sinica, 2009, 37(6): 1165 – 1169. (in Chinese)

[8] Nishimura H, Miyazaki O. CT system for spirally scanning subject on a movable bed synchronized to x-ray tube revolution [P]. United States Patent 4789929, 1988-12-06.

[9] Crawford C R, King K F. Computed tomography scanning with simultaneous patient translation[J]. Medical Physics. 1990, 17 (6): 967 – 982.

[10] La Rivière P J, Pan X. Fourier-based approach for interpola-

tion in single-slice helical CT[J]. Medical Physics. 2001, 28(3):381 – 392.

[11] Zou Y, Pan X. Exact image reconstruction on PI-lines from minimum data in helical cone-beam CT[J]. Physics in Medicine and Biology, 2004, 49(6):941 – 959.

[12] Noo F, Clackdoyle R, Pack J. A two-step Hilbert transform method for 2D image reconstruction[J]. Physics in Medicine and Biology, 2004, 49(17):3903 – 3923.

[13] Segal M, et al, Fast shadows and lighting effects using texture mapping[A]. Proceedings of the 19th annual conference on Computer graphics and interactive techniques[C]. Chicago, USA, 1992. 249 – 252.

作者简介:

张慧滔　男,1980 年生于河北,2008 年于首都师范大学检测成像工程中心获博士学位.现为首都师范大学检测成像工程中心助理研究员,中科院高能物理研究所博士后.主要研究领域:X 射线 CT 成像理论与应用,图像处理,应用数学.
E-mail:zhanght@ihep.ac.cn

于　平　女,1978 年生于河北,2005 年于湖南师范大学数学系获硕士学位.现为华北电力大学科技学院讲师.主要研究领域:应用数学,CT 成像理论.
E-mail:gfguyu@eyou.com