

闫天峰, 孙岩, 孙怡. 基于 GPU 实现的数字合成 X 射线体层成像投影数据的模拟方法 [J]. CT 理论与应用研究, 2011, 20(1): 1-10.
Yan TF, Sun Y, Sun Y. Tomosynthesis projection simulation based on GPU[J]. CT Theory and Applications, 2011, 20(1): 1-10.

基于 GPU 实现的数字合成 X 射线 体层成像投影数据的模拟方法

闫天峰, 孙岩, 孙怡[✉]

(大连理工大学信息与通信工程学院, 辽宁 大连 116023)

摘要: 数字合成 X 射线体层成像技术能利用有限角度下的投影数据重建物体任意断层的图像。在数字合成 X 射线体层成像重建算法研究中, 模拟投影数据是重要的步骤, 本文提出了一种基于 GPU 光线投射算法的数字合成 X 射线体层成像投影数据模拟方法。比较传统 CPU 模拟手段, GPU 模拟方法计算速率快, 且基于硬件支持的三线性插值能够得到更加接近实际的投影结果。本文采用传统 CPU 模拟方法和 GPU 模拟方法分别计算投影数据, 对两种方法获得投影使用 ART 算法和小波-伽辽金方法进行重建, 得到较好的重建效果。

关键词: Tomosynthesis; 投影数据模拟; 光线投射算法; GPU

文章编号: 1004-4140 (2011) 01-0001-10 **中图分类号:** O 242 **文献标识码:** A

1972 年, Grant 提出数字合成 X 射线体层成像 (Tomosynthesis) 的概念^[1], 它具有如下特点: 无需先验知识即可根据有限角度投影进行重建, 能够得到与身体轴向平行的断层, 而且容易推广到三维体积重建。Tomosynthesis 技术广泛应用于医疗诊断中, 在乳腺、胸和关节等部位的临床检查中发挥了重要作用; 在工业探伤、无损检测, 核工业检测等领域也具有广泛的应用前景。最近十几年, 随着计算机技术的发展和数字平板探测器的成功研制, Tomosynthesis 技术成为研究热点。

近年来图形处理器 (Graphic Processing Unit, GPU) 快速发展, 开始大量应用于高性能计算领域。2006 年 Nvidia 公司提出了 CUDA (Compute Unified Device Architecture) 体系架构, 为 GPU 编程提供了一个全新的软硬件架构, 具有专门的硬件访问接口, 降低了对开发人员计算机图形学知识的要求, 提高了开发效率。GPU 在医学图像的体绘制研究领域中已大量应用, 2008 年 Marsalek 等^[2]实现了基于 CUDA 的光线投射算法, 获得良好的重建质量和速度。近年 GPU 在 CT 计算方面的应用研究也十分活跃, 2009 年赵星等^[3]实现了基于 GPU 的锥束 CT 正投影算法, 其中采用二维纹理作为体表示模型, 具有较高的计算速率和精度。2009 年, Lu 等^[4]应用 CUDA 架构实现了体积 CT 的重建算法, 基于 GPU 加速使得 SART 算法与常用的滤波反投影算法在速度上达到同等水平。

利用 CT 进行图像重建第一步是要获取投影, 为了研究方便通常使用计算机模拟投影数据。孙怡等^[5]提出了一种体积 CT 投影数据模拟方法, 得到的投影数据能够精确反映各断层

收稿日期: 2010-11-12。

基金项目: 国家自然科学基金 (61071210); 辽宁省优秀人才基金。

细节信息, 黄建林等^[6]对图像重建中 X 射线投影模拟的常用方法进行了对比和分析, 这些文献中的模拟方法均在 CPU 上实现。借助 CUDA 体系架构, 本文提出了一种基于 GPU 光线投射算法的 Tomosynthesis 投影数据模拟方法。该方法使用三维纹理作为体表示模型并借鉴光线投射算法的思想进行投影模拟, 能够快速、精确地得到投影图像。与传统 CPU 上实现的模拟方法相比, 本文方法凭借 GPU 硬件支持的三线性插值采样, 能够实现对物体的均匀采样, 并快速获得更加接近实际情况的投影数据。针对 Shepp-Logan 模型组成的体数据分别采用传统 CPU 模拟方法和 GPU 模拟方法计算出投影数据, 并利用 ART 算法和小波-伽辽金方法进行重建, 验证了本文模拟方法的优点。^[7]

1 基于 CPU 的 Tomosynthesis 投影数据模拟

线型 Tomosynthesis 成像结构如图 1 所示, X 射线源沿直线轨迹运动, 探测器做同步反方向运动。在运动轨迹的一系列点上 X 射线源照射待重建物体, 在探测器平面上形成物体的一系列投影值。平面 ABCD 为该系统的支平面, $A'B'C'D'$ 表示 $ABCD$ 的投影平面。 P_l 表示第 l 条射线在探测器平面 $A'B'C'D'$ 的投影点, 其投影可以表示为物质的衰减系数沿 X 射线投影方向的线积分。离散化为二维切片后的投影可以表示为 X 射线穿过离散体数据各层切片上点的衰减系数的累加:

$$P_l = \sum_{i=1}^n f_i(x, y, z) \quad (1)$$

式 (1) 中, $f_i(x, y, z)$ 表示第 i 条射线穿过第 i 个切片的衰减系数, n 表示物体被分割的切片个数。投影数据模拟的一个重要步骤是计算各条 X 射线穿过物体各切片的对应点, 离散数据场中, 射线源和支平面上各点相连获得穿过物体的各条 X 射线的数学表示, 根据空间直线几何对应关系可计算 X 射线穿过各切片的对应点。

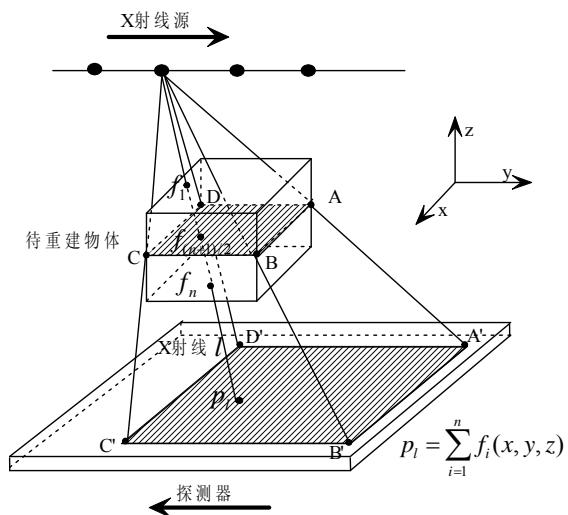


图 1 线型 Tomosynthesis 成像结构图
Fig. 1 The structure of liner tomosynthesis

图 2 中, $EFGH$ 所确定的平面为支平面, 其中心点为 O ; $S(x_s, y_s, z_s)$ 为任一射线源位置, 其所在的平行于 $EFGH$ 的平面为射线源平面, 射线源平面中心直线 SO' 为射线源 S 的移动轨迹; $E'F'G'H'$ 为平行于平面 $EFGH$ 的切片平面; $A(x_r, y_r, z_r)$ 为支平面上任意一点, SA 为连接 S 点和 A 点的任意一条 X 射线; $A'(x_i, y_i, z_i)$ 为射线 SA 穿过切平面 $E'F'G'H'$ 上的点。以支平面中心点 O 为坐标系原点, 建立三维空间坐标系, 由线型 Tomosynthesis 基本原理可得系统的已知条件为: ① 射线源坐标 $S(x_s, y_s, z_s)$; ② 支平面参考点坐标 $A(x_r, y_r, z_r)$; ③ 各切平面坐标 z_i 。由条件①和②建立直线方程:

$$\frac{x - x_s}{x_r - x_s} = \frac{y - y_s}{y_r - y_s} = \frac{z - z_s}{z_r - z_s} \quad (2)$$

由条件③得:

$$\Rightarrow \begin{cases} x_i = \frac{z_i - z_s}{z_r - z_s}(x_r - x_s) + x_s \\ y_i = \frac{z_i - z_s}{z_r - z_s}(y_r - y_s) + y_s \\ z_i = z_i \end{cases} \quad (3)$$

由此可得到射线 SA 上的点 $A'(x_i, y_i, z_i)$ 的坐标。同理, 可以计算出各条 X 射线穿过物体各切片的对应点, 进而在切片内对应点处进行临近离散采样得到衰减系数, 累加衰减系数获得该条 X 射线对应的投影数据。本文将上述投影模拟的实现过程称为传统 CPU 模拟方法。

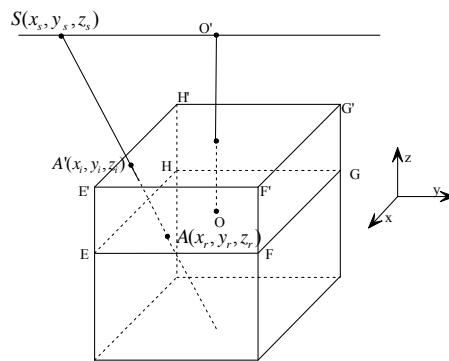


图 2 空间直线关系图
Fig. 2 Schematic of spatial line

2 基于 GPU 的 Tomosynthesis 投影数据模拟

2.1 GPU 光线投射算法

基于少量抽象概念的 CUDA 并行编程模型, 开发人员可以简单、高效地使用 NVIDIA GPUs

的强大并行计算能力。CUDA 模型中比较重要的概念是线程 (Thread) 层次结构, CUDA 将 GPU 设计成能够同时执行大量线程操作的计算设备。大量的线程以网格的形式组织成线程块 (block), 且每个线程有自己的 ID, 便于计算和管理。线程块终止时, 将在空闲的多处理器上启动新的线程块。

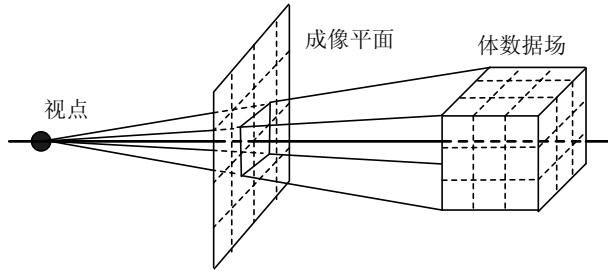


图 3 光线投射算法
Fig. 3 Raycasting algorithm

光线投射算法 (Raycasting)^[8] 是一种经典的体绘制算法, 图像重建质量高。光线投射过程如图 3 所示。算法主要原理简述如下: 以视点为起点发出射线, 穿过成像平面每个像素形成投射光线, 光线穿过三维体数据场, 在光线方向上等间距采样三维体数据场, 对距离采样点最近的 8 个体素的光学属性值进行三线性插值, 赋色和光照计算得到采样点的不透明度值及颜色值, 将每条光线上采样点的不透明度值和颜色值按照由后向前或者由前向后的方式进行组合, 从而计算出像平面的像素值。

光线投射算法计算量大, 传统 CPU 实现方法串行计算成像平面的每个像素点, 计算时间长, 无法实现实时显示。随着 GPU 的发展, 很多学者致力于基于 GPU 的光线投射算法加速研究。2003 年, 德国慕尼黑工业大学的 Kruger 等^[9]发表了基于 GPU 的光线投射算法实现方法, 实验采用的显卡是 ATI 9700。由于当时的显卡片段程序不支持循环、分支判断, 所以需要进行多步渲染来实现光线投射算法。直到 DirectX Pixel Shader 3.0 API 和 NVIDIA's NV-fragment-program 2 OpenGL 扩展的发布, 片段程序具备了支持循环、分支判断的能力。利用片段程序卓越的新性能, 斯图加特大学的 Stegmaier 等^[10]给出了单步光线投射算法的实现方法。

2006 年 Nvidia 公司推出了 CUDA 体系架构, 在 CUDA SDK 2.0 中附带了一个简单的 volume rendering 例程, 实现了基本的光线投射过程。基于 CUDA 的光线投射算法实现中, 穿过像平面上每个像素的光线相当于单个线程, CUDA 架构中大量线程并行执行, 所以像平面上的像素值可以同时进行计算。CUDA 实现的光线投射算法大致步骤如下: ①载入体数据, 绑定为三维纹理, 线程块的维度设置为 16×16 ; ②体绘制包围盒计算, 包围盒计算使用 Owen^[11]的方法计算穿过体数据场的有效光线的参数; ③计算每条光线方向, 在每条光线上进行体数据采样, 赋色和光照计算并累加计算得到最终结果; ④显示最终绘制结果。

2.2 基于 GPU 光线投射算法的 Tomosynthesis 投影模拟

光线投射算法几何关系与 Tomosynthesis 投影几何关系十分相近。如图 4 所示, 光源 O

相当图 2 的 X 射线源，光线投射算法成像平面相当物体支平面 $EFGH$ 。一般情况下，选取物体的中心切片位置为支平面，所以将光线投射算法中成像平面平移至体数据包围盒的中心位置就可满足 Tomosynthesis 的投影关系。基于 GPU 的光线投射算法实现中，根据载入三维体数据各维度数据规格来设置包围盒坐标。例如读入 $x \times y$ 大小的切片 z 张，则体数据维度为 $x \times y \times z$ ，相应包围盒设置为 $(-x, -y, -z)$ 、 (x, y, z) 两点包含的空间，为方便计算常常将包围盒坐标进行归一化，如果 x 是最大维度，那么包围盒坐标为 $(-1, -y/x, -z/x)$ 、 $(1, y/x, z/x)$ 。

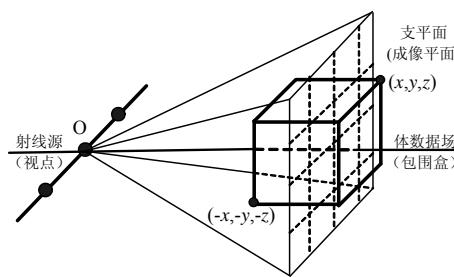


图 4 Tomosynthesis 和光线投射算法几何关系对应立体示意图
Fig. 4 3D schematic diagram of Tomosynthesis and Raycasting

射线源 O 同支平面（成像平面）上各点相连，得到穿过体数据场各条射线，可以求得各条射线的方向向量。图 5 中包围盒内的黑色粗线组成的六边形就是射线源 O 穿过体数据场的有效范围，在六边形内的各条射线沿各自方向进行等间距采样，将采样值进行累加，直到射线穿过包围盒为止，累加求得的采样值就是该条射线最终投影值。 O' 是位置改变后的射线源，灰色射线表示以 O' 为起点连接到支平面上各点的射线集合，得到射线源位置在 O' 处的投影。

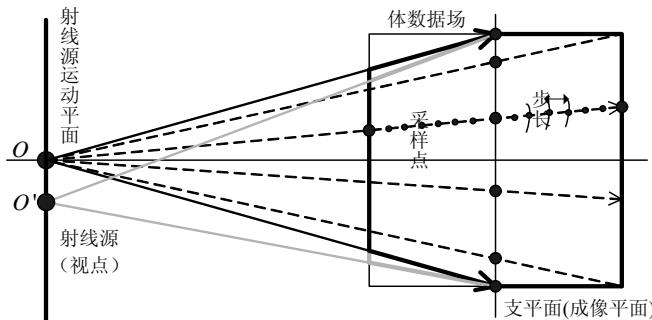


图 5 Tomosynthesis 与光线投射算法几何关系对应平面示意图
Fig. 5 2D Schematic diagram of Tomosynthesis and Raycasting

基于光线投射算法的 Tomosynthesis 投影模拟方法包括如下步骤：

- 1) 载入模拟体数据，绑定为三维纹理，根据体数据参数设置包围盒；调整光线投射体

绘制中的视点和视平面距离, 使之与传统投影模拟时射线源到支平面的距离相等。

- 2) 调整光线投射算法的视点位置, 对应为传统投影模拟时变化的射线源位置。
- 3) 连接视点和成像平面上各点, 计算穿过体数据场的各条光线方向向量。
- 4) 每条光线沿各自方向穿越体数据场包围盒; 每条光线上等间距采样, 每个采样点进行纹理三线性插值得到采样值, 累加该条光线上所有采样值得到投影值, 注意此步骤无需进行赋色和光照计算。

5) 将计算得到的投影数据规范化到 0~255 范围, 并输出为灰度的投影图像, 至此得到射线源在某一位置的投影数据。

- 6) 变化射线源位置, 转至 2); 若遍历完成射线源全部位置, 则结束模拟。

步骤 4) 使用三线性插值计算采样点数值, 每个采样点数值是根据距离采样点最近的 8 个体素值经过 x , y , z 各轴向三次线性插值计算得到, 所以更加接近光线穿过物体的实际情况。GPU 硬件支持纹理三线性插值计算, 所以 GPU 投影模拟方法可以快速地获得高精度投影。

基于 GPU 光线投射算法的 Tomosynthesis 投影模拟方法和传统 CPU 投影模拟方法具体实现时采样率方面存在一定差别, 图 6 为某一光源位置情况下两种投影模拟方法采样过程示意图。传统 CPU 投影模拟, 光线与切片的交点处进行切片内的临近采样, 采样率不恒定, 采样点插值计算只利用二维切片信息导致插值精度不高。基于 GPU 光线投射算法的投影模拟, 进行等间距的均匀采样且采样点使用高精度的三线性插值计算。除此之外, 基于 GPU 光线投射算法的 Tomosynthesis 投影模拟方法可以通过改变采样步长灵活控制采样率。若提高采样率, 采样点数增加, 可以更加充分地利用现有切片数据信息得到更加接近实际投影情况, 且对计算速率影响十分微小。为了合理对比, 本文实验中 GPU 模拟投影采样步长大小设置满足如下要求: 垂直于切片平面的投射光线采样点数与组成体数据场的切片层数相等, 即保证光线投射算法采样率与 CPU 模拟过程最大采样率相同。

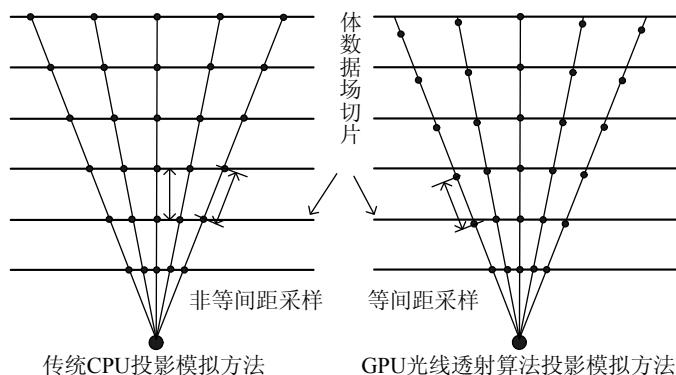


图 6 采样过程对比
Fig. 6 Comparison of sampling process

3 投影模拟结果与分析

Shepp-Logan 标准图像是 CT 算法测试通用模型之一, 本文通过该模型组成体数据来测

试模拟算法的速率和精度。本文实验使用的 PC 机配置如下：CPU 为 Intel Pentium 4，主频 3.0 GHz；内存 1 GB；GPU 为 Nvidia GeForce 9600 GT，显存大小 512 MB。本文实验程序在 Windows 系统 Visual Studio 2005 集成环境下开发，GPU 编程使用 CUDA 2.0，CPU 模拟方法编程使用 C++ 语言。

Tomosynthesis 投影模拟实验体数据由多幅 512×512 的 Shepp-Logan 标准图像组成，体数据模拟投影图像大小为 512×512 。模拟实验物体尺寸为 $512 \text{ mm} \times 512 \text{ mm} \times 48 \text{ mm}$ ，设定切片内像素大小为 1 mm，切片间距为 1.5 mm。传统 C++ 实现模拟方法读入的切片层数是 $48/1.5 + 1 = 33$ 层，GPU 体绘制模拟方法中读入切片仍是 33 层，GPU 内将体数据绑定为三维纹理。一般情况体绘制中标准包围盒空间坐标设置为 $(-1, -1, -1), (1, 1, 1)$ ，当三维体数据各方向长度规格不同时将包围盒相应方向坐标值乘以对应系数即可，针对上述数据包围盒空间最大和最小点坐标可以设为 $(-1, -1, -32 \times 1.5/512), (1, 1, 32 \times 1.5/512)$ ，这样 GPU 体绘制模拟方法中的切片间距也被设置为 1.5 mm。为了使基于光线投射算法模拟过程中采样率与 CPU 模拟过程最大采样率相等，将采样步长设为 $2 \times 1.5/512$ ，即保证垂直切片平面的投射光线上采样点数为 33，与组成体数据场切片层数相等。如图 7，射线源到物体支平面的距离为 1024 mm，射线源共有 9 处不同位置。

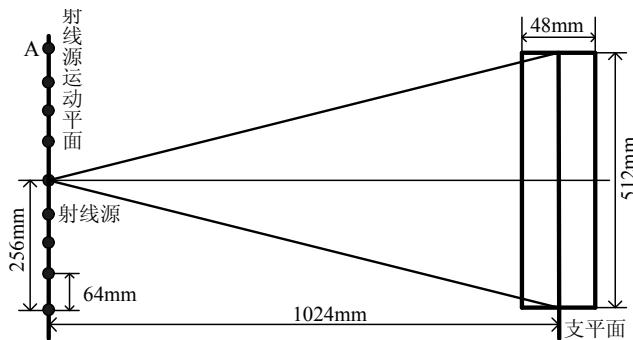


图 7 Tomosynthesis 投影模拟实验几何关系示意图
Fig. 7 Schematic diagram of Tomosynthesis projection simulation

按照上述几何关系，采用传统 CPU 模拟方法和基于 GPU 光线投射算法的模拟方法计算射线源位于 A 处的投影图像，并取图像的局部进行比较。图 8 (a) 是 CPU 模拟方法得到的投影图像及 100×100 像素大小的局部图像，图 8 (b) 是 GPU 模拟方法得到的投影图像及相同位置的局部图像，比较图 8 (a) 和图 8 (b) 局部图像细节效果可见 GPU 模拟投影更加平滑细腻，体现了三线性插值高精度的特点。图 9 是两幅局部图像第 10 行灰度分布对比，可见 CPU 模拟投影细节处存在明显的锯齿抖动，GPU 模拟投影则极为平滑更加接近光线穿越实际物体获得投影的情况。计算速率方面，传统方法 C++ 程序耗费时间为 21.79 s，基于 GPU 光线投射算法的模拟方法耗费时间为 127.19 ms，所用时间之比约为 171:1，加速效果十分明显。

对传统 CPU 模拟方法和本文 GPU 模拟方法得到的投影图像分别应用 ART 迭代重建算法和小波-伽辽金算法进行局部重建，这里选取 64×64 大小且密度变化较大的区域进行重建。应用 ART 迭代算法对上述两种模拟方法得到投影数据的支平面进行重建，图 10 (a) 是 ART

迭代算法重建 CPU 模拟数据得到结果, 图 10 (b) 是 ART 迭代算法重建 GPU 模拟数据得到结果, 图 10 (c) 是原始图像的对应部分; 应用小波-伽辽金算法分别重建上述两种模拟方法得到投影数据的支平面, 图 11 (a) 是小波-伽辽金重建 CPU 模拟数据得到结果, 图 11 (b) 是小波-伽辽金重建 GPU 模拟数据得到结果, 图 11 (c) 是原始图像的对应部分。

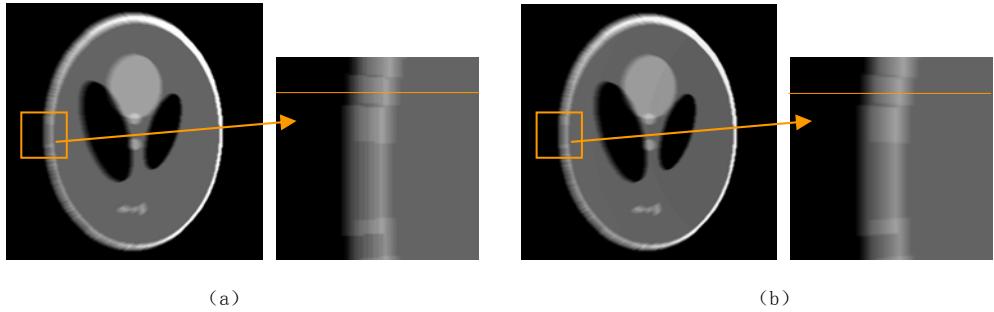


图 8 模拟投影和局部图像
Fig. 8 Projections and regions of interest

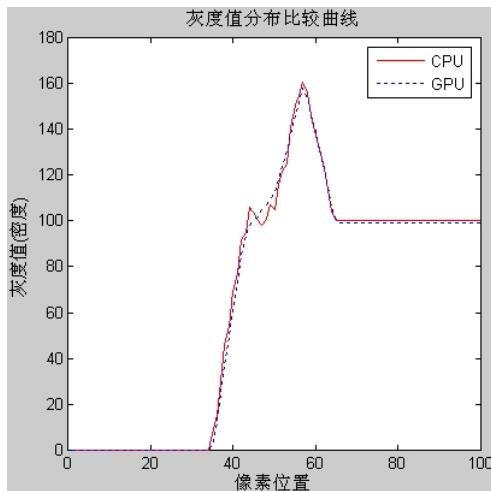


图 9 局部图像第 10 行灰度分布对比
Fig. 9 Comparison of the 10th row of region of interest

比较图 10 (b) 和图 10 (c) 以及图 11 (b) 和图 11 (c), 发现根据本文提出的 GPU 模拟方法获得的投影数据重建出支平面图像与原图像基本相同, 能够清楚地反映密度变化细节。但由于 ART 重建和小波-伽辽金重建均是近似重建算法且只重建投影局部图像, GPU 模拟投影数据重建出支平面与传统 CPU 模拟数据重建出支平面效果上差别并不十分明显, 与原始图像对应部分相比较两种方法模拟数据重建出支平面效果均较好。同时, GPU 投影模拟方法相比 CPU 投影模拟方法在速率上具有明显优势。根据上述结果进行对比分析可知, 基于 GPU 光线投射算法的 Tomosynthesis 投影数据模拟方法相对传统模拟手段具有如下优势: ①得益于 CUDA 架构强大的并行计算能力, 基于 GPU 光线投射算法的模拟方法计算速率更快。②采样点处数值通过硬件支持的三线性插值计算, 比较传统方法的采样计算可以方便、快速地获得高精度投影数据。

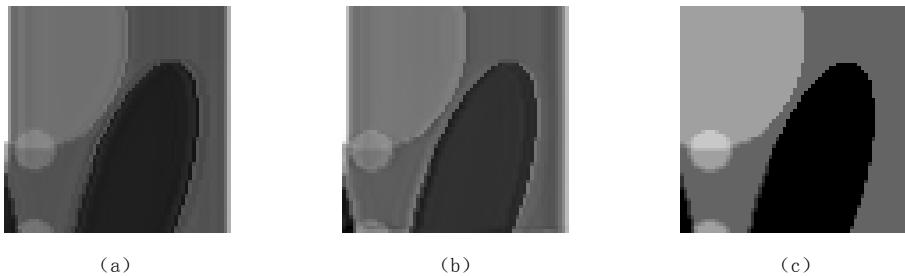


图 10 ART 迭代算法重建模拟数据支平面
Fig. 10 Reconstruction of projections by ART method

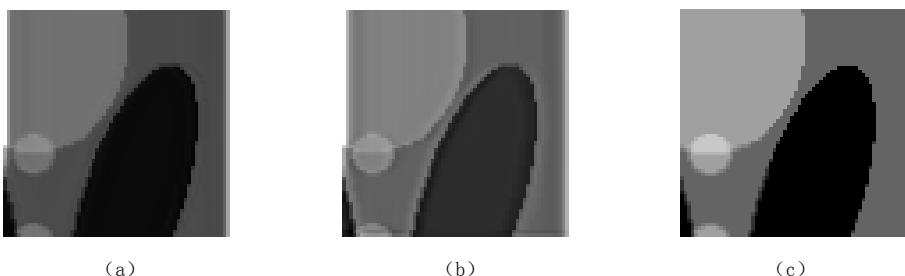


图 11 小波-伽辽金算法重建模拟数据支平面
Fig. 11 Reconstruction of projections by Wavelet-Galerkin method

4 总结

本文提出了一种基于 GPU 光线投射算法的 Tomosynthesis 投影数据模拟方法。该方法充分利用了光线投射过程和 Tomosynthesis 投影过程之间几何关系的相似性，将光线投射算法基于 GPU 的加速方法应用到 Tomosynthesis 投影模拟计算过程中，从而实现 Tomosynthesis 投影数据模拟的加速。凭借 GPU 强大的并行计算能力和硬件支持的纹理三线性插值，基于 GPU 光线投射算法的 Tomosynthesis 投影模拟方法计算速率快、投影图像精度高，为 Tomosynthesis 技术的进一步研究工作奠定了基础。另外，本文提出的基于 GPU 投影模拟方法同样适用于圆型 Tomosynthesis，在投影数据量较大情况下，本文模拟方法加速效果更加明显。

参考文献

- [1] Grant DG. Tomosynthesis: A three dimensional radiographic imaging technique[J]. IEEE Transactions Biomedical Engineering, 1972, 19(1): 20–28.
- [2] Marsalek L, Hauber A, Slusallek P. High-speed volume ray casting with CUDA[J]. IEEE Symposium on Interactive Ray Tracing, 2008: 185–85.
- [3] 赵星, 胡晶晶, 潘晓川, 等. 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 X, Hu JJ, Pan XC, et al. A novel GPU based cone beam CT forward projection method[J]. Acta Electronica Sinica, 2009, 37(6): 1165–1169.
- [4] Lu YQ, Wang WM, Chen SF, et al. Accelerating algebraic reconstruction using CUDA-enabled GPU[C]. International Conference on Computer Graphics, Imaging and Visualization. 2009: 480–485.

- [5] 孙怡, 侯颖, 胡家升, 等. 体积 CT 投影数据的模拟方法[J]. CT 理论与应用研究, 2005, 14(1): 1-6.
Sun Y, Hou Y, Hu JS, et al. A computer simulation method of projection data for cone-beam volume CT[J]. CT Theory and Applications, 2005, 14(1): 1-6.
- [6] 黄建林, 吕东辉. 图像重建中 X 射线投影模拟的常用方法[J]. 上海大学学报: 自然科学版, 2006, 12(3): 228-233.
Huang JL, Lv DH. General methods for X-ray projection simulation in image reconstruction[J]. Journal of Shanghai University (Natural Science), 2006, 12(3): 228-233.
- [7] 卢孝强, 白刚锋, 孙怡. 数字合成 X 射线体层成像的小波-伽辽金重建算法[J]. CT 理论与应用研究, 2010, 19(3): 13-23.
Lu XQ, Bai GF, Sun Y. Image reconstruction in tomosynthesis by wavelet Galerkin method[J]. CT Theory and Applications, 2010, 19(3): 13-23.
- [8] Levoy M. Display of surfaces from volume data[J]. IEEE Computer Graphics and Applications, 1988, 8(3): 29-37.
- [9] Kruger J, Westermann R. Acceleration techniques for GPU-based volume rendering[C]. IEEE Visualization, 2003: 287-292.
- [10] Stegmaier S, Strengert M, Klein T, et al. A simple and flexible volume rendering framework for graphics-hardware-based raycasting[C]. Fourth International Workshop on Volume Graphics, 2005: 187-241.
- [11] Owen S. Ray-box intersection[EB/OL]. (1998-04-01)[2010-08-01]. <http://www.siggraph.org/education/materials/HyperGraph/raytrace/rtinter3.htm>.

Tomosynthesis Projection Simulation Based on GPU

YAN Tian-feng, SUN Yan, SUN Yi[✉]

School of Information and Communication Engineering,
Dalian University of Technology, Dalian 116023, China

Abstract: Tomosynthesis can reconstruct any sectional image of object using limited angle projection. Before the research of reconstruction method, projection simulation is an important step, this paper proposed a projection simulation method based on GPU Raycasting algorithm. Comparing to CPU simulation method, GPU simulation method is faster in calculation and much closer to the practical projection based on hardware-based trilinear interpolation. This paper simulates projection using traditional CPU method and GPU method respectively, reconstructs the projections above by ART algorithm and Wavelet-Galerkin algorithm and achieves good results.

Key words: tomosynthesis; projection simulation; raycasting; GPU

作者简介: 闫天峰 (1984—), 男, 大连理工大学信息与通信学院硕士研究生, 主要从事CT断层重建和医学图像三维可视化研究, Tel: 0411-84707849, E-mail: ytf2004@163.com;
孙怡[✉] (1964—), 女, 大连理工大学教授、博士生导师, 主要从事光电信息检测、图像处理与模式识别等方面的研究, Tel: 0411-84707849, E-mail: lslwf@dlut.edu.cn。