

一种基于可编程图形硬件的快速三维图像重建算法*

梁 亮, 张定华, 毛海鹏, 顾 娟

(西北工业大学 现代设计与集成制造技术教育部重点实验室, 陕西 西安 710072)

摘 要: 针对三维 CT 图像重建时间过长这一瓶颈问题, 给出了 FDK 算法的几何描述, 并据此导出了利用可编程图形硬件加速三维图像重建的方法, 最后利用图形处理器的可编程、高精度以及并行计算等特性实现了该方法。实验结果表明, 该三维图像重建方法非常有效, 与原始算法相比取得了 8 倍左右的重建加速比。

关键词: 三维图像重建; 快速重建; 图形处理器; FDK 算法

中图法分类号: TP301. 6 文献标识码: A 文章编号: 1001-3695(2006)01-0241-03

A Fast 3D Image Reconstruction Method Based on Programmable Graphics Hardware

LIANG Liang, ZHANG Ding-hua, MAO Hai-peng, GU Juan

(Key Laboratory of Contemporary Design & Integrated Manufacturing Technology of Ministry of Education, Northwestern Polytechnical University, Xi'an Shanxi 710072, China)

Abstract: Aiming at promoting the 3D image reconstruction of 3D Computed Tomography (CT), first gave the geometric description of the FDK method, and then educed the 3D fast image reconstruction algorithm using the programmable graphics hardware. At last made full use of the characters of GPU to implement the method. The result shows that the reconstruction speed is promoted effectively, and can get a reconstruction speedup of 8 times compared to the original algorithm.

Key words: 3D Image Reconstruction; Fast Reconstruction; GPU(Graphics Processing Unit); FDK Algorithm

近年来 CT 技术得到了突飞猛进的发展, 其扫描方式从平行束扫描过渡到扇形束扫描一直到现在国际上研究的热点之一——锥束扫描, 即三维 CT。随着三维 CT 硬件系统扫描速度的提高, 实际生产应用中的效率瓶颈也从扫描速度转变为图像重建速度。尤其是在某些要求实时成像的场合, 现有图像重建算法的效率根本无法满足实际需求, 这也极大地限制了三维 CT 的实际应用。

目前应用最广泛的三维图像重建方法是 FDK 算法^[1]。为了提高重建速度, 许多学者基于该算法做了大量相关研究工作。在重建算法的改进方面, 王贤刚等人^[2]分别改进原始算法, 通过使用预先计算好的几何查找表来加速重建过程, 但该方法所取得的重建加速比令人不是很满意。在重建运算的硬件化实施方面, Mueller 和 Yagel^[6]使用纹理映射技术实现了基于 SART (Simultaneous Algebraic Reconstruction Technique) 方法的快速重建。但是, 这些方法都是基于工作站上实现, 费用较高; 所采用的图形加速卡使用定点计算, 给重建图像带来精度损失; 图形加速卡不可编程, 限制了该算法的移植性。

随着可编程图形硬件的出现和快速发展, 计算机图形硬件的性能得到了很大的提升, 尤其是其核心图形处理单元 GPU 大约每六个月更新一代, 逐渐开始展现其作为通用处理单元的

巨大潜力和强大功能。目前 GPU 不仅能提供简单的算术运算功能, 还能提供复杂的图形并行计算能力以及 32 位的高精度浮点数据处理能力, 扩展了它的应用领域^[3 ~5]。

本文从 FDK 算法出发, 给出其几何描述, 并据此导出了利用投影纹理映射加速三维图像重建的原理, 最后采用 Nvidia 公司推出的 Cg^[8] (C for Graphics) 语言实现了该算法。在西北工业大学研制的 CBVCT 系统上实施了上述思想, 实验结果表明重建效率提升比较显著, 取得了 8 倍左右的重建时间加速比。

1 FDK 算法

1.1 FDK 重建算法

FDK 算法是 Feldkamp 等人在 1984 年针对锥束几何圆形扫描轨迹提出的一种近似三维图像重建算法, 当前大多数三维精确重建算法都是在该算法的基础上进一步发展而来的。图 1 给出了锥束圆形扫描轨迹的几何坐标关系, 其中, $O-xyz$ 表示世界坐标系, $O-uv$ 表示探测器投影数据坐标系, $O-ts$ 表示射线源坐标系。此外, 规定 z 轴是旋转中心轴, s 轴总是通过射线源中心, 且与探测器平面相垂直。为了简化 FDK 算法的表示和计算, 这里已将探测器投影数据按几何比例折算为过原点 O 的平面投影数据。FDK 计算公式分为两步, 如下所示:

(1) 滤波

$$P_{\theta}^{*}(i, j) = \left(\frac{d_{so}}{\sqrt{d_{so}^2 + i^2 + j^2}} P_{\theta}(i, j) \cdot h(i) \right)$$

(1)

(2) 反投影

$$f(x,y,z)=\int_0^{2\pi}u^2P_{\theta}^*(p,q)d\theta$$
 (2)

其中,
$$\begin{cases} t=x\cos\theta+y\sin\theta \\ s=y\cos\theta-x\sin\theta \\ u=\frac{d_{so}}{d_{so}-s}, p=ut, q=uz \end{cases}$$
 (3)

上式中, d_{so} 表示射线源中心到原点 O 的距离; $P_{\theta}(i,j)$, $P_{\theta}^*(i,j)$ 分别表示在旋转角 θ 下的投影和滤波投影数据; $h(i)$ 表示卷积函数; (p,q) 表示重建体素点在滤波投影平面上的反投影点地址; $f(x,y,z)$ 表示在体素点 (x,y,z) 处的重建图像数据。

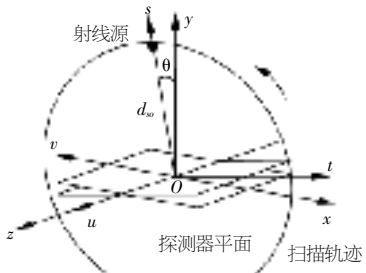


图1 锥束圆形扫描轨迹的几何坐标关系

1.2 FDK 算法的几何描述

图2给出了更为直观的三维锥束圆形扫描轨迹的几何坐标关系,其中, S 为射线源中心位置, $V(x,y,z)$ 为三维图像中待重建的一点, $V(p',q')$ 为 V 沿着射线方向与平板探测器相交的一点。 D 为探测器中心, uv 直角坐标系为探测器坐标系, VD 与 V 轴成 α 角度。过点 V 向直线 SD 作垂线, 并与 SD 相交于点 $M(m,n)$ 。 ts 直角坐标系为射线源坐标系, 其中 t 轴与 x 轴成 θ 角度。

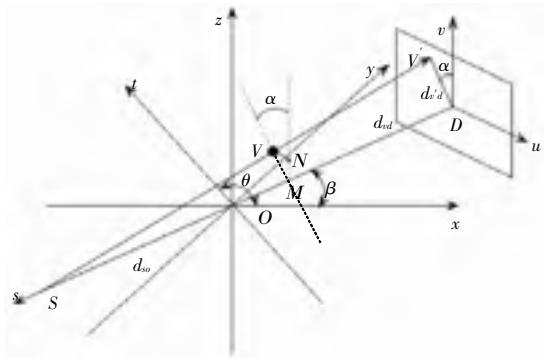


图2 三维锥束圆形扫描轨迹的几何坐标关系

根据几何坐标关系可得

$$\beta+\frac{\pi}{2}=\theta$$
 (4)

故点 M 的坐标为

$$m_x=d_{om}\cos\beta=-d_{om}\sin\theta$$
 (5)

$$m_y=d_{om}\sin\beta=-d_{om}\cos\theta$$
 (6)

点 V 与点 M 之间的距离为

$$f(V,M)=\sqrt{(x-m)^2+(y-n)^2}=\sqrt{(x+d_{om}\sin\theta)^2+(y-d_{om}\cos\theta)^2}$$
 (7)

又 $VM \perp SD$

$$\frac{df^2}{dd_{om}}=0$$
$$\Rightarrow d_{om}=y\cos\theta+x\sin\theta=s$$
 (8)

其中
$$\begin{cases} t=x\cos\theta+y\sin\theta \\ s=y\cos\theta-x\sin\theta \end{cases}$$
 (9)

从而可以得到

$$d_{vm}=\sqrt{d_{vo}^2-d_{om}^2}=\sqrt{x^2+y^2+z^2-s^2}$$
 (10)

综合式(8)和式(10)可得

$$\begin{cases} \cos\alpha=\frac{d_{mn}}{d_{vm}}=\frac{z}{\sqrt{x^2+y^2+z^2-s^2}} \\ \sin\alpha=\frac{d_{vn}}{d_{vm}}=\frac{\sqrt{x^2+y^2-s^2}}{\sqrt{x^2+y^2+z^2-s^2}} \end{cases}$$
 (11)

而根据三角形的比例关系可得

$$\frac{d_v'd}{d_{vm}}=\frac{d_{so}+d_{od}}{d_{sm}}$$
 (12)

因为 (p,q) 为 uv 坐标系下的坐标, 所以

$$\begin{cases} p=d_{vm}\sin\alpha \\ q=-d_{vm}\cos\alpha \end{cases}$$
 (13)

综合式(8)、式(10)~式(12)最终可以计算出点 V 的几何坐标:

$$p'=\frac{d_{so}+d_{od}}{d_{so}-s}t$$
 (14)

$$q'=\frac{d_{so}+d_{od}}{d_{so}-s}z$$
 (15)

将式(14)、式(15)按几何比例折算为过原点 O 的平面投影数据, 即分别乘以比例因子 $\frac{d_{so}}{d_{so}+d_{od}}$ 可得

$$\begin{aligned} p &= ut \\ q &= uz \end{aligned}$$
 (16)

与式(3)对比可知, FDK 算法中反投影的几何意义也即计算滤波后的投影图像沿着锥束射线方向对三维图像上每一点的贡献值, 从而可以得到 FDK 算法几何描述的伪代码如下所示:

```
// 滤波过程
For each projection image
    // 加权
    Weight pixels
    // 滤波
    Filter each column
End For
// 反投影过程
For each projection image
    For each voxel V_j
        Project V_j along cone-beam rays
        Calculate voxel update dV_j = P_theta^*(p, q) using bilinear interpolation
        Weight dV_j by depth factor u^2
        Update V_j = V_j + dV_j
    End For
End For
```

2 基于 GPU 的快速重建算法

2.1 快速重建算法

根据 FDK 算法的几何意义, 需要将滤波后的投影数据按照其采集时的几何关系反投影回待重建的各个切片图像, 而该过程类似于幻灯机的投影过程, 可以采用 Segal 等人^[7]所介绍的投影纹理映射(Projective Texture Mapping) 予以实现。

图3给出了利用投影纹理映射实现反投影的原理图。对于每一个投影图像 $proj_{\theta}$, 将其绕待重建切片中心 O 旋转相应的角度 θ 设置投影角度等于锥角 γ , 利用投影纹理映射将该投

影图像映射到各个待重建的平面上。在完成纹理映射之后,采用平行投影方式观看待重建的平面,从而得到了反投影后的图像。待所有投影图像处理完毕之后,将所有数据进行加权累加,最终可以得到重建后的切片图像。以下给出了反投影部分的伪代码:

```
For each projection image
    Setup the texture matrix
    For each slice to be reconstructed
        Perspectively project the projection image
        Read the image from the framebuffer
        Save the temporary result
    End For
End For
For each slice to be reconstructed
    Weight and sum the corresponding temporary result
    Save the final result
End For
```

2.2 算法实现

Vertex Shader 程序如下所示:

```
void v_projTexturing(
    float4 position: POSITION,
    float3 normal: NORMAL,
    out float4 oPosition: POSITION,
    out float4 texCoordProj: TEXCOORD0,
    uniform float4 x4 modelViewProj,
    uniform float3 lightPosition,
    uniform float4 x4 textureMatrix)
{
    oPosition = mul( modelViewProj, position );
    texCoordProj = mul( textureMatrix, position );
}
```

采用 Nvidia 公司推出的 Cg^[5] 语言,利用 GPU 实现了投影纹理影射。上面给出的该过程的 Vertex Shader 源代码主要完成投影纹理坐标的计算。下面给出了 Fragment Shader 源代码,根据投影纹理坐标插值计算该点处的颜色值:

```
void f_projTexturing( float4 texCoordProj: TEXCOORD0, out float4 color: COLOR, uniform sampler2D projectiveMap)
{
    //Fetch color from the projective texture
    float4 projColor = tex2Dproj( projectiveMap, texCoordProj );
}
```

3 实验结果

采用 Visual C++ 6.0和 Cg 1.2.1 编译器作为开发工具,使用C++ 6.0和 Cg 语言,基于 OpenGL 实现了以上算法的全过程。测试计算机配置为 Pentium 4 2.4GHz CPU, 512MB DDR 内存, GeForce FX5700 256MB 显卡。重建用例是使用 256 幅 512² 的空心涡轮叶片投影图像重建 512³ 图像,将感兴趣区域设置为重建空间的 Z 向内切圆柱域。原始 FDK 算法耗时 8 591.207s,而本文提出的快速重建方法只需要 1 045.922s,重建加速比大约为 8 倍。实验证明效率的提升是非常显著的。图 4 给出了快速重建方法计算出的部分三维切片图像。

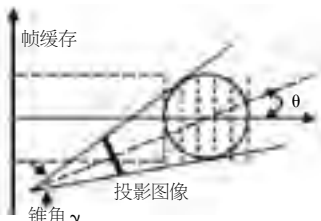


图 3 反投影原理图



图 4 快速重建得到的切片像

4 结论

本文利用 GPU 的可编程、高精度和并行计算等特性,实现了 FDK 快速重建算法,取得了令人满意的结果,并且该方法成本低廉,便于升级、再开发和维护,能够满足三维 CT 的实际需求。由于重建过程需要在计算机主存和图形加速卡之间进行大量的数据交换,因此目前系统重建的性能很大程度上还受到当前 PC 架构系统总线的限制。但是随着 GPU 性能的提升,尤其是 PCI Express 系统总线技术的逐渐成熟,数据交换不再成为系统的效率瓶颈,三维重建速度会得到质的提升。

参考文献:

[1] eldkamp LA, Davis LC, Kress JW. Practical Cone-beam Algorithm [J]. J Opt. Soc. Am. A. , 1984, 1(6) : 612-619.

[2] 王贤刚,郭志平,彭亚辉,等.用几何参数表实现快速三维重建 CT 图像 [J]. 清华大学学报,2002, 42(5) : 655-658.

[3] <http://www.gpgpu.org> [EB/OL].

[4] 吴恩华,柳有权.基于图形处理器(GPU)的通用计算[J]. 计算机辅助设计与图形学学报, 2004, 16(5).

[5] Jo o Luiz Dihl Comba, Carlos A Dietrich, Christian A Pagot, et al. Computations on GPUs: From a Programmable Pipeline to an Efficient Stream Processor [J]. Revista de Informatica Teoricae Aplicada, 2003, X(2).

[6] Klaus Mueller, Roni Yagel. Rapid 3-D Cone-beam Reconstruction with Simultaneous Algebraic Reconstruction Technique(SART) Using 2-D Texture Mapping Hardware[J]. IEEE Transactions on Medical Imaging, 2000, 19(12) : 1227-1237.

[7] M Segal, C Korobkin, R van Widenfelt , et al. Fast Shadows and Lighting Effects Using Texture Mapping[C]. SIGGRAPH '92, vol. 26, 1992. 249-552.

[8] Mark W R, Glanville, et al. Cg: A System for Programmable Graphics Hardware in a C-like Language[J]. ACM Transactions on Graphics, 2003, 22(3) : 806-907.

作者简介:

梁亮(1981-),男,博士研究生,主要方向为工业 CT、计算机图形/图像处理;张定华(1958-),男,教授,博士生导师,主要研究方向为 CAD/CAM、工业 CT、计算机图形/图像处理等;毛海鹏(1976-),男,博士研究生,主要研究方向为工业 CT、计算机图形/图像处理;顾娟(1979-),女,博士研究生,主要研究方向为 CAID/CAD。

[8] wok-Wai Wong, Kin-Man Lam, Wan-Chi Siu. An Efficient Algorithm for Human Face Detection and Facial Feature Extraction under Different Conditions [J]. Pattern Recognition, 2001, 34: 1993-2004.

作者简介:

王学武(1974-),男,湖南益阳人,硕士,研究方向为图像处理;石跃祥(1964-),男,湖南益阳人,教授,博士,主要研究方向为图像处理与模式识别。