

# 基于 OpenGL 的光线投射算法的研究

江兆尧, 路 游, 贾广忠, 亓永刚, 徐 振

(中国石油大学 计算机科学与技术系, 北京 102249)

**摘 要:**利用 VC++ 6.0 以及 OpenGL 实现光线投射算(Ray Casting)法。介绍了光线投射算法以及 OpenGL 的知识。主要实验方法为:第一、对原始 CT 图像进行数据预处理、数据分类,得到满足绘制要求的规则体数据,并且对各个数据点赋予颜色值和不透明度;第二、利用体绘制算法绘制出人头(主要是人脑)图像,并且能够实现空间上三个方向的浏览与绘制功能,但是绘制速度比较慢。实验结果表明:采用光线投射算法能够实现人脑的三维可视化,并且能够绘制出清晰图像,图像效果令人满意。

**关键词:**光线投射算法;重采样;体绘制

**中图分类号:**TP301.6

**文献标识码:**A

**文章编号:**1673-629X(2010)03-0218-03

## Research on Ray Casting Algorithm Based on OpenGL

JIANG Zhao-yao, LU You, JIA Guang-zhong, QI Yong-gang, XU Zhen

(Department of Computer Science & Technology, China University of Petroleum, Beijing 102249, China)

**Abstract:** Ray casting algorithm is achieved through using the tools of VC++ 6.0 and OpenGL. Introduced the ray casting algorithm and the knowledge of OpenGL. The crucial experimental method is following. Firstly, the regular volume data needed in the process of volume rendering are obtained by the preprocess of original CT images, and the values of color and opacity are distributed to every data. Secondly, the image of head which the main part is the brain of person is rendered with the volume rendering, and the function of browse and rendering can be completed in three directions of space. However, the speed of rendering is slow. Experimental results show that a proper example of volume rendering is exemplified in the visualization of 3D the person's brain by ray casting algorithm, and the image of rendering is quite clear, the result of image is very satisfied.

**Key words:** ray casting algorithm; resampling; volume rendering

## 0 引 言

三维数据场的可视化问题是科学计算可视化的核心。三维数据场可视化算法主要分为面绘制和体绘制,其中,面绘制是最早出现的绘制方法,面绘制方法需要构造几何图元(如等值线、等值面等),然后对几何图元进行拟合和逼近重构出物体表面轮廓,它的主要缺点是不能够显示出物体的内部信息。伴随着面绘制技术的发展出现了一些相应的算法,如轮廓重建表面法、移动立方体(Marching Cubes Algorithm)、立体沟纹法(Cuberille Algorithm)、剖分立方体法(Dividing Cubes Algorithm)等;体绘制不需要构造几何图元,而是由体素作为基本单元,采用体光照模型直接从三维数据场

中绘制出物体的轮廓。体绘制不仅能够显示物体的表面信息,更为重要的是能够显示物体的内部详细信息。由于体绘制不需要构造几何图元,因此又被称为直接体绘制。

常用的体绘制算法主要有:以图像空间为序的体绘制算法、以物体空间为序的体绘制算法、频域体绘制方法和基于纹理映射的体绘制方法。

以物体空间为序的体绘制的算法主要有:体单元算法又称为足迹法或脚印法、基于错切-变形技术(Shear-Warp)的体绘制算法、子区域投射法。

光线投射算法是以图像空间为序的体绘制的众多算法中的一种。光线投射算法已经在医学、气象、有限元分析、分子模型构造等领域得到广泛应用。文中将利用 OpenGL 具体实现光线投射算法。

## 1 光线投射算法

### 1.1 光线投射算法的基本原理

从屏幕上的像素出发,根据设定的观察方向发出

收稿日期:2009-07-07;修回日期:2009-10-21

基金项目:国家自然科学基金支持项目(60673188)

作者简介:江兆尧(1979-),男,硕士,CCF会员,研究方向为人工智能、计算机图形学;路 游,副教授,研究方向为CAGD(计算机几何辅助设计)、计算机图形学、计算几何。

一条射线,并使该射线穿过三维数据场,沿着这条射线选择  $k$  个等距离的采样点,并由距离某一采样点最近的 8 个数据点的数值做三线性插值,求出该采样点的不透明度和颜色值。最后对各个采样点进行图像合成。

该算法的原理如图 1 所示<sup>[1,2]</sup>。

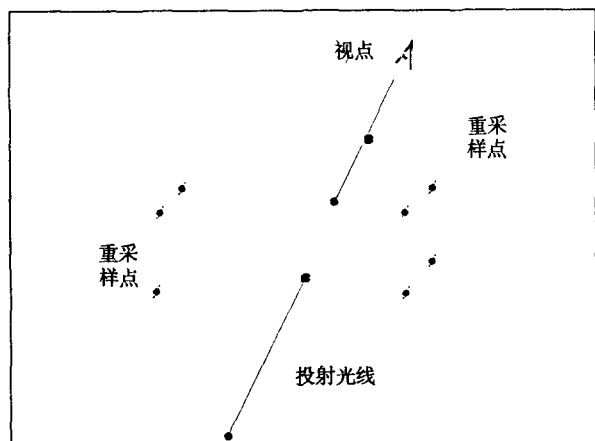


图1 光线投射算法的原理

## 1.2 光线投射算法的流程

通过光线投射算法的原理研究发现,要将离散的三维的数据场最终成像,主要经过数据值分类、空间投影变换、重采样以及插值计算、图像合成五个关键技术。

利用图 2 所示的流程图来具体表示光线投射算法<sup>[3]</sup>。

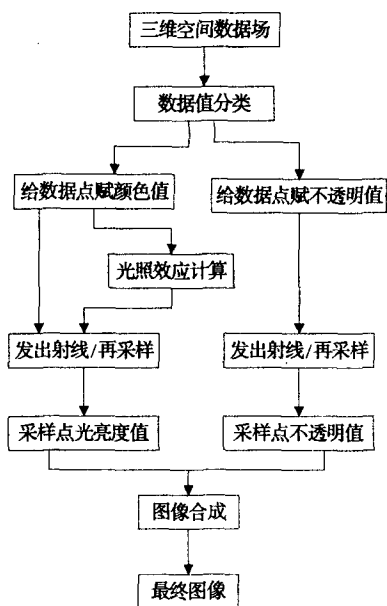


图2 光线投射算法的流程

## 1.3 数据值分类

三维体数据是三维空间中多种不同物质的测量结果或计算机数值计算的结果。最初的体数据不包含颜色和透明度,因此这就需要采用原始数据为依据,确

定物质的颜色以及不透明度。理论上,物质的一种或者几种属性,一定程度上可以跟其他物质鉴别或者区分开来。例如:人体断层扫描(CT)或者核磁共振(MRI)得到的数据,对应的骨骼、肌肉等有着不同密度的物质,就有不同的CT值,在图像中表现为不同的灰度值。为了在最终的图像中显示不同的物质,就需要对体数据进行分类,找出体数据与不同物质之间的对应关系。

常用的数据值分类法:

### (1) 阈值法。

对全部采样点的取值进行统计以后,设定若干阈值  $d_i (i = 1, 2, \dots, n)$ 。如果各采样点的数值用  $f(x_i, y_j, z_k)$  表示,则满足  $d_i \leq f(x_i, y_j, z_k) < d_j$  条件的采样点归入同一类中。

### (2) 概率法。

如果多种物质共存的物质中,每一个体元不是由单一的物质构成的,如果能够估算出每一个体元中不同物质所占的百分比,那么就可以得到精确的分类。虽然精确地求出在每一个体元中不同物质所占的百分比是不可能的,但是求出每一个体元中不同物质的概率是可能的,因此就产生了概率分类法<sup>[4]</sup>。

一组医学CT图像的灰度值分布的统计图形可以看成是由不同物质的CT图像灰度值分布组合而成。假设任何一个体元具有灰度值  $I$  的概率为:

$$P(I) = \sum_{i=1}^n p_i P(I|i) \quad (1)$$

其中,  $n$  表示该体元中存在的物质的种类的数目,  $p_i$  是该体元中第  $i$  类物质所占的百分比,  $P(I|i)$  是第  $i$  类物质CT图像具有的灰度值  $I$  的条件概率。在人体CT图像中,每一种物质所对应的灰度值概率密度分布函数是已知的,因此可以采用贝叶斯公式估算出灰度值为  $I$  的体元中所含有的第  $i$  类物质的概率:

$$P(i|I) = \frac{P(I|i)}{\sum_{i=1}^n P(I|i)} \quad (2)$$

各种物质的百分比在某区域内呈现线性变化,这符合医学CT图像的一般情况,一旦得到了个体元中不同物质的百分比,就相对比较简单的对其颜色进行赋值了。

## 1.4 空间、投影变换

空间变换是以三维几何变换为基础的。三维几何变换包括三维基本几何变换和三维符合变换。三维基本几何变换是相对于坐标原点和坐标轴进行的几何变换,而三维复合变换是指图形做一次以上的变换,变换结果是把每一次需要的三维基本变换矩阵相乘。空间变换主要涉及平移、旋转、缩放。OpenGL 实际上是一

种图形与硬件的接口<sup>[5]</sup>。在 OpenGL 中有相应的函数可以直接调用,如平移: `void glTranslate(fd)(TYPE x, TYPE y, TYPE z)`;它是把当前矩阵与一个表示移动物体的矩阵相乘,这个移动矩阵由  $x$ 、 $y$  和  $z$  值指定;旋转: `void glRotate(fd)(TYPE angle, TYPE x, TYPE y, TYPE z)`;把一个当前矩阵与一个表示移动物体的矩阵相乘,以逆时针方向绕着从原点到点  $(x, y, z)$  的直线进行旋转,  $angle$  参数表示旋转的角度;缩放: `void glScale(fd)(TYPE x, TYPE y, TYPE z)`;把当前矩阵与一个表示沿各个轴对物体进行拉伸、收缩或反射的矩阵相乘,这个物体中每个点的  $x$ 、 $y$  和  $z$  坐标与对应的  $x$ 、 $y$  和  $z$  参数相乘。

通常在三维坐标系中表示一个物体,是用该物体上的点的坐标分量  $(x, y, z)$  来表示,如果要把一个三维坐标系中的物体在一个二维的平面上显示出来,就需要对三维物体进行投影。OpenGL 中常用的投影方法是平行投影以及投射投影。投影变换的目的就是定义一个视景体,使得视景体外多余的部分裁剪掉,最终图像只是视景体内的有关部分<sup>[6]</sup>。

平行投影:使用一组平行投影将三维对象投影到投影平面去。经过这种投影后,物体始终保持它们原来的实际大小以及它们之间的角度。在 OpenGL 中使用的函数为, `void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far)`;

投射投影:使用一组由投影中心产生的放射投影线,把三维对象投影到投影平面上。这种投影方式能够产生近大远小的效果,它常被用在动画、视觉模拟等领域,它与日常生活中观察事物的方式相同。在 OpenGL 中使用的函数为, `void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble near, GLdouble far)`;

由于 OpenGL 提供众多关于计算机图形学方面的函数,为编程提供了很大的方便。

### 1.5 重采样以及插值计算

体绘制中关键步骤之一就是三维离散数据场的重采样。因为重采样的计算量相当大,所以体绘制中重采样是基于信号处理中的采样理论。理论上,实现重采样应有以下 4 步<sup>[7]</sup>:

①选择恰当的重构核函数,对离散的采样信号进行卷积运算,重构连续的原始信号;

②根据给定的观察方向对连续的三维数据场进行几何变换;

③一般来说,结果信号的采样频率是已知的,由此可计算出被采样信号的奈奎斯特频率极限,采用低通滤波函数去掉高于这一极限的频率成分;

④对滤波后的函数重新采样。

插值计算主要用在明暗计算中,计算各个数据点的梯度值。明暗计算的目的是为了增加图像的真实感而进行的。设三维数据场中某一数据点的函数值为  $f(x_i, y_j, z_k)$ ,采用中心差分方法求出该数据点处的梯度值<sup>[8]</sup>:

$$\begin{aligned} \text{Grad}_x &= f(x_i + 1, y_j, z_k) - f(x_i - 1, y_j, z_k) / 2\Delta x \\ \text{Grad}_y &= f(x_i, y_j + 1, z_k) - f(x_i, y_j - 1, z_k) / 2\Delta y \\ \text{Grad}_z &= f(x_i, y_j, z_k + 1) - f(x_i, y_j, z_k - 1) / 2\Delta z \end{aligned} \quad (3)$$

得到各个数据点的梯度值后,就可以用光照模型计算出各数据点处的漫反射分量,更加突出地显示体数据中的边界信息。

### 1.6 图像合成

从光线投射算法的流程图中看出,最后一步是图像合成,主要目的沿着某一像素点所发出的射线,计算该射线上各采样点的颜色值以及不透明度值,然后按照一定的规则合成,从而形成该像素点的最终颜色值,将屏幕上各个像素点的颜色值全部计算出来以后,就形成了最终的图像。目前,体绘制中主要有两种图像合成算法。假设:第  $i$  个体元的颜色值为  $C_{\text{now}}$ ,不透明度为  $\alpha_{\text{now}}$ ,进入第  $i$  个体元的颜色值为  $C_{\text{in}}$ ,不透明度为  $\alpha_{\text{in}}$ ,经过第  $i$  个体元后的颜色值为  $C_{\text{out}}$ ,不透明度值为  $\alpha_{\text{out}}$ 。

两种图像合成算法分别为:

①由后向前的图像合成<sup>[1,4]</sup>

$$C_{\text{out}} = C_{\text{in}}(1 - \alpha_{\text{now}}) + C_{\text{now}}\alpha_{\text{now}} \quad (4)$$

②由前向后的图像合成

$$\begin{aligned} C_{\text{out}}\alpha_{\text{out}} &= C_{\text{in}}\alpha_{\text{in}} + C_{\text{now}}\alpha_{\text{now}}(1 - \alpha_{\text{in}}) \\ \alpha_{\text{out}} &= \alpha_{\text{in}} + \alpha_{\text{now}}(1 - \alpha_{\text{in}}) \end{aligned} \quad (5)$$

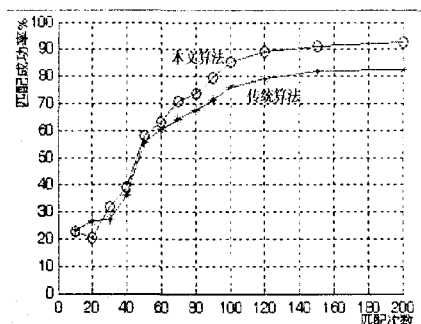
通过比较上面两式,不难看出,由前向后进行图像合成的过程中,不透明度  $\alpha$  必然逐步增大,当  $\alpha$  值趋近于 1 时,说明该像素点的图像已经接近于完全不透明,后面的体元不会再对该像素点的图像有所贡献,因而可以不再计算了。由于由前向后的图像合成方法可以省去无效的计算,速度较快,因此得到了较为广泛的应用。

## 2 利用 OpenGL 实现光线投射算法

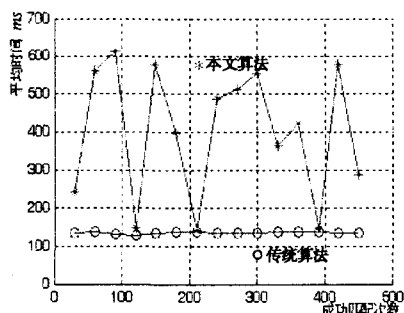
采用 VC++ 6.0 与 OpenGL 相结合来实现光线投射算法,实现了三个不同平面的切片,把这些切片数据按照位置 and 角度进行处理,目的是获得不同位置上的点、线、面的数据。最后完成对大脑的体绘制。在实

(下转第 224 页)

换及噪声情况下分析算法的适应性能力。其中,基准图大小为  $160 \times 160$ , 实时图大小为  $640 \times 640$ 。



(a) 匹配成功率



(b) 平均匹配时间

图3 文中算法和传统二维算法的匹配结果比较

### 3 结束语

综合上述分析,可以看出,文中的立体景象匹配算法总体的适应性良好。应用表明,该算法的实时性和

适应性较好,匹配时间能达到 0.14s,正确匹配率达到 92.5%。

有待研究的图像立体匹配方法很多,诸如多基准图对离散化的系列实时图进行匹配,以及定义、构造立体图像的数学模型并进行匹配等。

#### 参考文献:

- [1] 章毓晋. 图像工程(中册). 图像分析[M]. 第2版. 北京:清华大学出版社,2005:41-50.
- [2] LI Zhu-lin, ZHAO Zong-tao, ZHANG Hong-lei, et al. Binocular Stereo Matching Algorithm for Wide Baseline Based on Stable Extremal Regions[C]//FSKD'07(The 4th International Conference on Fuzzy Systems and Knowledge Discovery). Haikou, China: IEEE, 2007.
- [3] 曹菲. 景象匹配制导基准图选定准则研究[D]. 西安:第二炮兵工程学院,2006:71-73.
- [4] 詹总谦, 张祖勋. 基于多基线立体匹配技术的三维重建[J]. 地理空间信息, 2004, 6(2): 17-19.
- [5] 赵锋伟. 景象匹配算法、性能评估及其应用[D]. 长沙:国防科技大学, 2002: 34-36.
- [6] Stefano L D, Marchionni M, Mattocci S. A fast area-based stereo matching algorithm[J]. Image and Vision Computing, 2004, 22: 983-1005.
- [7] LI Zhu-lin, ZHAO Zong-tao, ZHANG Gen-yao, et al. A Stereo Matching Implementation Method for Uncalibrated Color Image Based on SCILab [C]//OSSS-EA'06. Hangzhou, China: [s. n.], 2006: 189-195.

(上接第 220 页)

现过程中,主要的操作步骤:

①原始医学数据的读取:读入医学图像的数据(人头,主要是大脑),关键是实现数据头与数据内容的对应;

②浏览、绘制功能的实现:利用 VC++ 6.0 与 OpenGL 实现了三维空间内三个方向的观察、绘制功能,并且利用光线投射(Ray Casting)算法实现图像的绘制。

### 3 结束语

利用 VC++ 6.0 以及 OpenGL 实现了光线投射算法,实现从 CT 扫描医学图像中读取数据并进行绘制,绘制图像质量比较高,能够取得比较满意的结果。利用光线投射算法实现的过程中,最大的困难之一是每次绘制的过程中,需要重新进行大量的计算。所以绘制速度与图像质量很难同步,因此后续工作中需要进行算法的优化。

#### 参考文献:

- [1] Levoy M. Volume Rendering by Adaptive Refinement[R]. [s. l.]: UNC, 1988.
- [2] Levoy M. Efficient Ray Tracing of Volume Data[J]. ACM Transactions on Graphics, 1990, 9(3): 245-261.
- [3] 唐泽圣. 三维数据场可视化[M]. 北京:清华大学出版社, 1999.
- [4] Drebin R A, Carpenter L, Hanrahan P. Volume Rendering[J]. Computer Graphics, 1988, 22(4): 65-74.
- [5] Angel E. OpenGL 程序设计指南[M]. 第2版. 李桂琼, 张文祥译. 北京:清华大学出版社, 2005.
- [6] 郭涛, 平西建. 基于 OpenGL 的三维地形可视化技术研究[J]. 计算机工程与设计, 2008, 29(1): 193-195.
- [7] Lims, Shinbs. Bi Directional Distance Map for Efficient Volume Ray Casting [C]//ISCIS, International Symposium on Computer and Information Sciences. Istanbul: [s. n.], 2006: 334-342.
- [8] 唐泽圣, 袁骏. 用图象空间为序的体绘制技术显示三维数据场[J]. 计算机学报, 1994, 17(11): 801-808.