

3 维图像数据场的快速重建

刘 峰 朱秀昌

(南京邮电大学图像处理与图像通信江苏省重点实验室, 南京 210003)

摘 要 3 维图像重建中最典型的一类算法是体绘制方法, 然而由于体绘制中的重采样和插值这两类基本运算, 其计算量却是很大的, 这往往影响重建的速度, 为此, 提出了在重建过程中采用减少插值、快速插值以及快速重采样等多项措施和方法, 以加速 3 维图像的重建。实验表明, 采用该方法重建时间约缩短了 1/3, 可见, 这些措施和方法是切实可行有效的。

关键词 3 维重建 体绘制 插值 重采样

中图法分类号: TP391.41 **文献标识码:** A **文章编号:** 1006-8961(2006)05-0700-06

The Realization of Fast Reconstruction of 3D Image Data Fields

LIU Feng, ZHU Xiu-chang

(Image Processing and Image Communications Key Lab., Nanjing Univ. of Posts & Telecomm., Nanjing 210003)

Abstract Volume rendering method is one of the most typical algorithms of 3D image reconstruction. The two basic operations, resampling and interpolation, are the most time-consuming operations, which usually affect the reconstruction speed. So, some measures and methods of decreasing interpolation, fast interpolation and fast resampling are proposed in paper. The result shows these measures is practical and effective.

Keywords 3D reconstruction, volume rendering, interpolation, resampling

1 引 言

3 维图像重建是数据场可视化的一个重要应用, 尤其是在医学诊断治疗方面。例如医生很容易通过 CT (computer tomography)、MRI (magnetic resonance image)、激光共焦扫描和超声波等医学成像技术来得到一系列 3 维断层图像数据场, 由于这些 3 维断层图像包含了人体组织、器官的 3 维信息, 因此可认为是 3 维图像数据场, 但由于仅仅依靠这些断层数据很难直观地体现和确定人体组织、器官的 3 维结构及其相互关系, 因此, 3 维图像数据场的可视化技术作为医生诊断病灶病情的重要手段和工具, 已成为当前医学影像学科研究的热点。特别是随着医学图像数据格式 DICOM3.0 的推出, 3 维重

建图像质量的改善以及重建速度的提高, 3 维医学图像重建将会得到更加广泛的应用。

3 维图像重建技术可以分为以下两类: 一类是面绘制 (surface-rendering)^[1-6]; 另一类是体绘制 (volume rendering)^[7-11]。其中面绘制的基本思想是首先从数据场中提取感兴趣物体的表面信息, 并把 3 维图像数据场转换为由一系列多边形表面片拟合成的等值表面 (iso-surface); 然后再用计算机图形学中的面绘制算法根据光照、阴暗模型进行消隐来得到 3 维的显示图像。由于这种方法要从数据场中提取表面信息, 因此必须要判断每一个体元 (cell) 是否在相应的表面上, 这对于复杂的、边界模糊的人体组织和器官来说, 则由于经常会产生虚假的显示表面或者显示面上有空洞出现, 致使重建图像质量较差。而体绘制则无需构建数据场中的等值表面,

基金项目: 江苏省自然科学基金项目 (KB2004151)

收稿日期: 2005-05-24; 改回日期: 2005-08-09

第一作者简介: 刘峰 (1964 ~), 男, 1994 年和 1997 年分别于南京理工大学物理电子学与光电子学专业获硕士和博士学位, 现为南京邮电大学信息工程系副教授。主要从事图像处理和图像通信等方面的研究。E-mail: liuf@njupt.edu.cn

它可按照体元对光线的吸收、衰减、反射等性质直接对数据场中的体元进行明暗处理,由于此时体元的许多细节得到保留,从而可合成具有 3 维显示效果的图像。体绘制重建的图像不仅真实感较强,而且图像质量要比面绘制好,其不足之处是由于体绘制需要对所有体元进行处理,因此计算量非常大,这就限制了图像的绘制速度。对于加快体绘制的方法,已有许多文章进行讨论,例如文献[6]采用正交小波变换,并充分利用了不同断层之间的相关性来加快体绘制;文献[9]采用了并行体绘制方法 PVR (parallel volume rendering),把处理器分成几个不同的节点来同时处理庞大的数据集;文献[11]讨论了利用自适应子块划分以及基于边界和填充策略的有效子块合并算法来加快体绘制;文献[12]提出了一个称为时空分割树 TSP (time-space partitioning) 的数据结构方法,用来加快时间变化场的体绘制;文献[13]将优化的传递函数用于直接体绘制系统;文献[14]在远程可视化中,通过服务器和客户端的同时处理来加快大数据场的体绘制。

2 体绘制处理显示过程

体绘制技术的中心思想是假定每一个体元本身可以发光,同时对入射到体元的光线有遮挡和吸收作用,体元具有不透明度 (opacity),而体元对光线的透射、发射和反射的共同作用则构成该体元对显示图像的贡献。其中光线的透射能力取决于体元的不透明度,而光线的发射能力则取决于体元的物质度 (objectness),即物质度越大,发射能力越强;光线的反射程度取决于体元的法向矢量与入射光线的夹角关系。如今最有代表的两种体绘制方法是单元投影法^[8]和光线投影算法^[9],其中前者是先根据 3 维图像数据场中的每一个体元的性质计算出它对屏幕上各像素点的贡献,然后通过自前向后或自后往前的合成来得到最后的图像;而后者则是先从 2 维显示屏幕上的每一点发出一条光线,并穿过 3 维图像数据场来得到重采样点,然后再用插值方法求出光线上各重采样点的光亮度值,最后对此光线上所有重采样点对屏幕上像素点灰度值的贡献加以合成,即可得到显示屏幕上该像素点的光亮度值。体绘制坐标示意图如图 1 所示。

定义原来 3 维图像数据场所在的坐标系为物坐标系 $\hat{o}\hat{x}\hat{y}\hat{z}$,所在空间为物空间。同时定义数据场外某观察点 P 所处的坐标系为显示坐标系 $oxyz$,对应

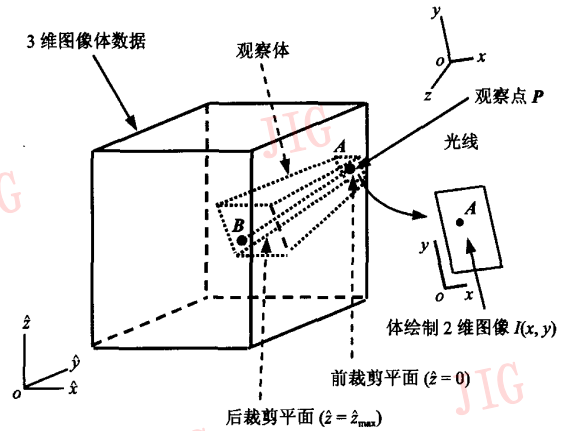


图 1 体绘制坐标示意图

Fig. 1 Relation of volume rendering coordinate

的 2 维显示平面称为视平面 xoy ,视平面用来显示体绘制后的 2 维图像 $I(x, y)$ 。其观察点位置可随着实际观察需要而变化,这样一条从观察点 P 发出的经过显示平面 A 点的光线,在穿过 3 维图像数据场后,光线在数据体中的长度为 AB ,用 l 表示。从物坐标系 (物空间) $\hat{o}\hat{x}\hat{y}\hat{z}$ 到显示坐标系 (显示空间) $oxyz$ 的变换矩阵为 $T_{4 \times 4}$,即 $T_{4 \times 4}:\hat{o}\hat{x}\hat{y}\hat{z} \Rightarrow oxyz$,其可以由平移 M 、缩放 S 、旋转 R 和投影 Q 等变换得到。从显示空间到物空间的变换矩阵为 $T_{4 \times 4}^{-1}$ 。

由于医学成像系统通常都采用均匀几何网格来获取数据,因此 3 维图像数据场是一个规则化和结构化的网格数据,也称为体数据 (volume data)。设某 3 维图像数据场由 N_z 层扫描得到的切片图像组成,每一层切片的大小为 $N_x \times N_y$, d_x, d_y, d_z 分别为在 $\hat{x}, \hat{y}, \hat{z}$ 方向上网格节点间的距离,则标号为 (i, j, k) 的网格节点的坐标为

$$\begin{aligned} \hat{x} &= id_x \quad i = 0, 1, 2, \dots, N_x \\ \hat{y} &= jd_y \quad j = 0, 1, 2, \dots, N_y \\ \hat{z} &= kd_z \quad k = 0, 1, 2, \dots, N_z \end{aligned} \quad (1)$$

网格节点上的数据为 $F(\hat{x}, \hat{y}, \hat{z})$,表示采集到的图像亮度值。由相邻 8 个网格节点构成的长方体即为体元,有些文献上称为体素 (voxel)。

假设在显示坐标系中从观察点发出的入射光线经过显示空间到物空间的变换 $T_{4 \times 4}^{-1}$ 之后,该光线穿过 3 维图像数据场后,在数据场中的长度为 l 。若其与物空间中位于网格节点 (i, j, k) 的体元相交,则重采样点坐标为 $(\hat{x}, \hat{y}, \hat{z})$ 。如该体元的不透明度为 $\alpha(\hat{x}, \hat{y}, \hat{z})$,则物空间 3 维图像数据场与该光线所有

相交的体元对体绘制 2 维显示平面图像的贡献为

$$I(x, y) = \sum_{\hat{z}=\hat{z}_{\min}}^{\hat{z}=\hat{z}_{\max}} \left[F(\hat{x}, \hat{y}, \hat{z}) \alpha(\hat{x}, \hat{y}, \hat{z}) \times \prod_{m=0}^{\hat{z}} (1 - \alpha(\hat{x}(m), \hat{y}(m), \hat{z}(m))) \right] \quad (2)$$

其中, $I(x, y)$ 为体绘制后的 2 维图像。 $F(\hat{x}, \hat{y}, \hat{z})$ 为光线穿过物空间时重采样点的亮度值。此时光线方向从前裁剪平面光线入射点 A 到后裁剪平面光线出射点 B 。

因此, 基于上述光线投影的体绘制过程如下:

(1) 对于视平面上的每一像素点 (x, y) , 若由观察点经此像素点发出一光线 v , 则在显示坐标系中的光线方程可用参数 t 表示, 即

$$v = (x + lt, y + mt, nt)$$

其中, (l, m, n) 代表显示坐标系的单位向量;

(2) 把此光线反变换到物空间, 由于显示平面上的点 $(x, y, 0)$ 对应于物空间上的点 $(\hat{x}, \hat{y}, \hat{z})$, 因此在物坐标系中的光线方程变为

$$\hat{v} = (\hat{x} + \hat{l}t, \hat{y} + \hat{m}t, \hat{z} + \hat{n}t)$$

其中, $(\hat{l}, \hat{m}, \hat{n})$ 是物坐标系的单位向量, 同时判断此光线穿过物空间时, 与 3 维图像数据场的交点个数;

(3) 如果交点数小于两个, 则显示平面上的此像素点的颜色为背景色, 否则转下一步;

(4) 找出物空间图像数据场中与此光线相交的所有体元, 同时得到与每个体元表面的交点, 再根据交点进行重采样, 并通过插值得到新采样点的亮度值和颜色值;

(5) 按自前向后或自后向前的顺序, 同时按照组织的不透明度、颜色等属性来计算出重采样点的亮度值对屏幕像素点的光强贡献, 然后即可进行图像合成;

(6) 再转到步骤(1)绘制屏幕其他像素点。

上述 3 维重建过程中非常重要的一个重要问题是重建速度。显然, 由于步骤(4)进行每一条光线与物空间 3 维图像数据场所有体元表面的求交运算、重采样过程, 以及新采样点的插值等运算的计算量很大, 因此直接影响体绘制的速度。下面进一步介绍本文在提高绘制速度上所采取的措施和方法。

3 提高快速重建的措施和方法

3.1 减少插值计算

众所周知, 由于人体的同种组织一般具有比较

相近的灰度值, 且不同组织的交界处灰度通常会有跳变, 也就是说, 当一个体元的 8 个网格节点上的灰度值很接近时, 就可以认为此体元内的物质为同类物质, 因此重采样点的插值计算可直接用网格点的灰度值来代替。笔者对本文的 3 维图像数据场做过统计, 在头颅切片的 $127 \times 127 \times 39 = 629\,031$ 个体元中, 8 个网格节点灰度差值(最大灰度值减最小灰度值)小于 2 的占 26%, 小于 5 的占 46.4%。因此, 选择适当的阈值去掉一些网格节点就可以大大减少求插值的计算量。在某些情况下, 如果采用一些算法对每个切片进行预处理, 比如求不同物质分界线的边缘算法, 则将使得相同物质区域的灰度值为 0, 而且由于相邻切片也具有很强相关性, 尤其是切片距离很小时, 更是如此, 因此经过对切片的预处理就可提高重绘制速度。

3.2 采用快速的插值方法

一条光线与体元的相交一般存在以下 3 种情况:

- (1) 光线刚好经过一个网格节点, 与体元只有唯一一个交点;
- (2) 光线经过一个体元的两个网格节点, 也就是说光线与体元的其中一条边重合;
- (3) 光线穿过体元, 与体元的两个表面相交, 具有两个交点。

第 1 种情况最简单, 无需插值计算; 第 2 种情况, 一个样点只要经过一次线性插值即可。如图 2 所示, 设网格节点 A, B 的灰度值为 F_1 和 F_2 , 则 AB 上任一 O 点($OA/AB = \gamma$)处的一元线性插值 F_o 为

$$F_o = \gamma F_1 + (1 - \gamma) F_2 \quad (3)$$

也可以写成

$$F_o = F_2 + \gamma(F_1 - F_2) \quad (4)$$

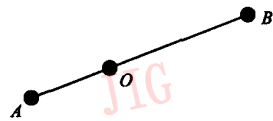


图 2 一元线性插值

Fig. 2 1-D linear interpolation

显然, 式(4)插值运算比式(3)插值运算要少 1 次乘法运算。如果针对上述第 3 种情况, 对体元内重采样点进行三元线性插值时, 每插值计算一个采样点就可以从原来利用式(3)的 14 次乘法减少到 7 次乘法。

对于第3种情况,如对体元内一点 $P = (\hat{x}, \hat{y}, \hat{z})$ 进行插值,可根据以下两种情况分别对待(为了描述简单起见,以网格节点 P_0 为原点,其灰度值用 F_0 表示,如图3所示)。

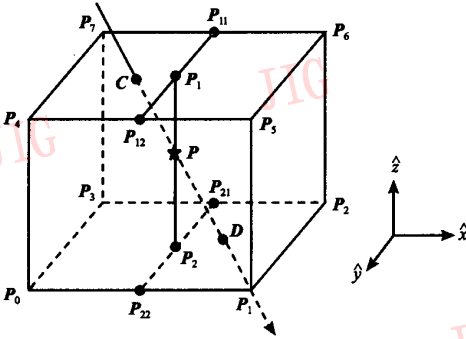


图3 体元内插值方法

Fig.3 Interpolation method for inside cell

(1) 如果知道了光线与体元表面的两个交点 $C = (\hat{x}_C, \hat{y}_C, \hat{z}_C)$ 和 $D = (\hat{x}_D, \hat{y}_D, \hat{z}_D)$, 则 $P = (\hat{x}, \hat{y}, \hat{z})$ 点的插值用如下公式表示:

$$F_P = \frac{F_D s_1 + F_C s_2}{s_1 + s_2} = \frac{1}{s_1 + s_2} [F_D \quad F_C] \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad (5)$$

其中, F_D 和 F_C 是光线与体元相交插值得到的 D 和 C 点的灰度值。

$$s_1 = \sqrt{(\hat{x}_D - \hat{x})^2 + (\hat{y}_D - \hat{y})^2 + (\hat{z}_D - \hat{z})^2}$$

$$s_2 = \sqrt{(\hat{x} - \hat{x}_C)^2 + (\hat{y} - \hat{y}_C)^2 + (\hat{z} - \hat{z}_C)^2}$$

$$F_D = \frac{1}{d_z d_y} [F_3 \quad F_2 \quad F_1 \quad F_0] \begin{bmatrix} (d_z - \hat{x}_D) \hat{y}_D \\ \hat{x}_D \hat{y}_D \\ \hat{x}_D (d_y - \hat{y}_D) \\ (d_z - \hat{x}_D) (d_y - \hat{y}_D) \end{bmatrix}$$

$$F_C = \frac{1}{d_z d_y} [F_7 \quad F_6 \quad F_5 \quad F_4] \begin{bmatrix} (d_z - \hat{x}_C) \hat{y}_C \\ \hat{x}_C \hat{y}_C \\ \hat{x}_C (d_y - \hat{y}_C) \\ (d_z - \hat{x}_C) (d_y - \hat{y}_C) \end{bmatrix}$$

(2) 如果仅知道 $P = (\hat{x}, \hat{y}, \hat{z})$ 点,而没有求得交点 C 和 D 时,则采用三线插值得到

$$F_P = \frac{1}{d_z d_y d_x} \begin{bmatrix} \hat{z} F_3 + \hat{z}_P F_7 \\ \hat{z} F_2 + \hat{z}_P F_6 \\ \hat{z} F_1 + \hat{z}_P F_5 \\ \hat{z} F_0 + \hat{z}_P F_4 \end{bmatrix}^T \times \begin{bmatrix} \hat{y} \hat{x}_P \\ \hat{y} \hat{x} \\ \hat{y}_P \hat{x} \\ \hat{y}_P \hat{x}_P \end{bmatrix} \quad (6)$$

其中, $\hat{x}_P = d_z - \hat{x}$, $\hat{y}_P = d_y - \hat{y}$, $\hat{z}_P = d_z - \hat{z}$ 。

3.3 采用快速的重采样点估计方法

根据3.2节分析可知,若知道了体元内要插值点的位置,便可以用式(5)或式(6)来得到插值。实际上,在 F_0, F_1, F_2, F_3 和 F_4, F_5, F_6, F_7 变化不大时,式(5)和式(6)的差值在2维情况下,小于 $|F_0 + F_3 - F_1 - F_2|/4$ 和 $|F_4 + F_7 - F_5 - F_6|/4$,也就是说,两种方法得到的数值比较接近。这给人们一个启示,能否直接快速估计出光线穿过物空间时每一个体元内的重采样点,而不必花时间求解光线与各体元表面的交点呢?

下面考虑平行投影方式显示3维图像数据场,并设光线投影方向在显示空间和物空间都是一个固定的方向,而不同的光线只是经过显示平面上的不同像素位置。显示空间的一条光线 $v = (x + lt, y + mt, nt)$,它与显示平面的交点为 $(x, y, 0)$,则该光线经坐标变换后在物空间上的光线方程为: $\hat{v} = (\hat{x} + \hat{l}t, \hat{y} + \hat{m}t, \hat{z} + \hat{n}t)$ 。若其与物空间中的平面族

$$\hat{z} = kd_z \quad k = 0, 1, 2, \dots, N_z - 1 \quad (7)$$

相交,则其交点所对应的参数为

$$t_k = -\hat{z}/\hat{n} + kd_z/\hat{n} \quad (8)$$

光线在物空间的交点坐标为 $(\hat{x} + t_k \hat{l}, \hat{y} + t_k \hat{m}, kd_z)$ 。

当考察物空间与 $\hat{z} = kd_z$ 相邻的另一族平面 $\hat{z} = (k+1)d_z$ 相交时,则与物空间的交点坐标对应的参数为

$$t_{k+1} = -\hat{z}/\hat{n} + (k+1)d_z/\hat{n} \quad (9)$$

则与物空间的交点坐标为

$$(\hat{x} + t_{k+1} \hat{l}, \hat{y} + t_{k+1} \hat{m}, (k+1)d_z)$$

因此,两个相邻平面与光线的交点在物空间3个坐标轴上的差值为

$$\Delta \hat{x} = \hat{l} d_z / \hat{n}$$

$$\Delta \hat{y} = \hat{m} d_z / \hat{n} \quad (10)$$

$$\Delta \hat{z} = d_z$$

也就是说,两个相邻平面与光线交点之间的距离为一常数(如图4所示)。

由此可见,一旦知道了一条过显示平面像素点 (x, y) 的光线到物空间的变换以及光线与物平面的外轮廓交点之后,就可以很方便地通过式(10)求出光线与任一平行平面的交点参数以及交点坐标来进行重采样点的插值。如果已知某些中间值如 $\hat{l} d_z / \hat{n}$ 、 $\hat{m} d_z / \hat{n}$ 等,则利用式(9)求交点参数只需一次加法及一次乘法,求交点坐标只需做3次乘法和两次加法即可。如果利用式(11)的递增迭代方式,则求交点的计算量将更少,这就大大减少了运算量。

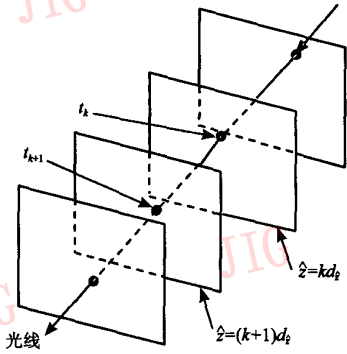


图 4 光线与相邻体元表面交点坐标

Fig.4 Coordinate points of ray go through adjacent ceel surface

$$\begin{aligned}
 t_{k+1} &= t_k + d_z / \hat{n} \\
 \Delta \hat{x}_{k+1} &= \hat{x}_k + \hat{l} d_z / \hat{n} \\
 \Delta \hat{y}_{k+1} &= \hat{y}_k + \hat{m} d_z / \hat{n} \\
 \Delta \hat{z}_{k+1} &= \hat{z}_k + d_z
 \end{aligned}
 \tag{11}$$

也就是说,通过式(11)可以很方便得到光线穿过3维图像数据场后的重采样点位置。由于根据式(6)便可以得到其插值,从而使得体绘制显示过程中的步骤4可以大大简化。3维图像快速重建框图如图5所示。

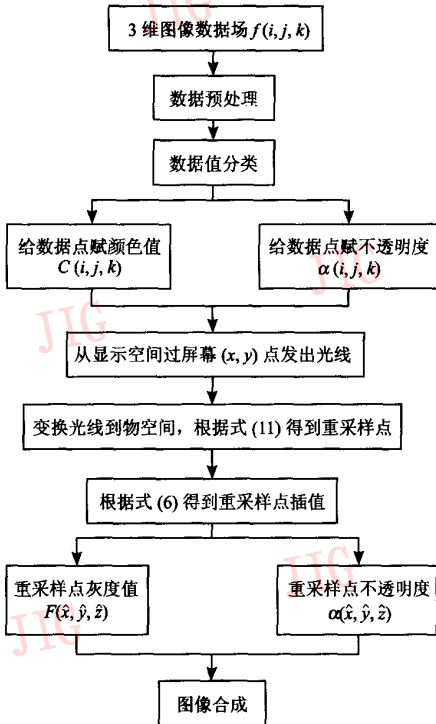


图 5 快速体绘制流程图

Fig.5 Flow diagram of fast volume rendering

4 实验分析

本文采用上述方法对头颅切片和螺旋藻断层图像进行3维重建实验,实验选用 Pentium 4 主频 CPU 2.4G PC 机,内存 256M, Window XP 操作系统,整个处理算法采用 Visual C++ 语言编写。头颅切片共 40 层,每层大小是 128 × 128 pixels, 8bit 采样。螺旋藻断层共 49 层,每层大小是 512 × 512 pixels, 8bit 采样。实验时,首先通过对两个 3 维图像数据场进行分析来得出一体元内组织近似属于同一类物质的体元所占比例情况(如表 1、表 2 所示)。

表 1 一个体元中 8 个网格节点灰度值最大差值的体元数占整个图像场体元的比例(头颅切片图像)

Tab.1 The ratio of the cells of max. difference gray value for 8 nodes to image data fields

(head cross-sectional image)				
最大差值	<1	<2	<5	<10
体元数	116 584	163 578	292 344	414 346
所占比例(%)	18.5	26.0	46.5	65.9

表 2 一个体元中 8 个网格节点灰度值最大差值的体元数占整个图像场体元的比例(螺旋藻切片图像)

Tab.2 The ratio of the cells of max. differena gray value for 8 nodes to image data fields

(spiral alga cross-sectional image)				
最大差值	<1	<2	<5	<10
体元数	7 430 834	9 829 059	10 811 674	11 105 766
所占比例(%)	60.4	80.1	88.1	90.5

当然,由于坐标变换(与观察距离、观察角度、平移、旋转、缩放等有关),实际上可能在同一体元内有多于一个重采样点,而且不同观察角度时,平均每一光线在图像数据场中的插值点数也不相同。表 3、表 4 分别列出了两种图像场在几种不同位置

表 3 不同观察角度改进前后的头颅切片图像重建时间及成像效果

Tab.3 Head cross-sectional image reconstruction time and 3D effects from difference view point

观察位置	位置 1	位置 2	位置 3	位置 4
插值总点数	476 124	429 508	445 802	423 252
每一光线平均插值点数	23.4	26	26	22
采用本文方法后的成像时间(s)	0.609	0.547	0.561	0.516
采用本文方法前的成像时间(s)	0.913	0.902	0.905	0.894
实际成像效果图	图 6(a)	图 6(b)	图 6(c)	图 6(d)

表 4 不同观察角度改进前后的螺旋藻图像重建时间及成像效果

Tab.4 Spiral alga cross-sectional image reconstruction time and 3D effects from difference view point

观察位置	位置 1	位置 2	位置 3	位置 4
插值总点数	11 614 995	11 594 947	11 487 150	11 520 605
每一光线平均插值点数	44.31	44.23	43.82	43.94
采用本文方法后的成像时间(s)	2.335	2.242	2.191	2.402
采用本文方法前的成像时间(s)	3.381	3.237	3.075	3.112
实际成像效果图	图 7(a)	图 7(b)	图 7(c)	图 7(d)

时,采用本文一系列措施和方法以及不采用这些方法的成像时间比较,在此所列时间不包括重建之前读入 3 维图像数据的时间(单位为 s)。给出的实际成像效果如图 6、图 7 所示。

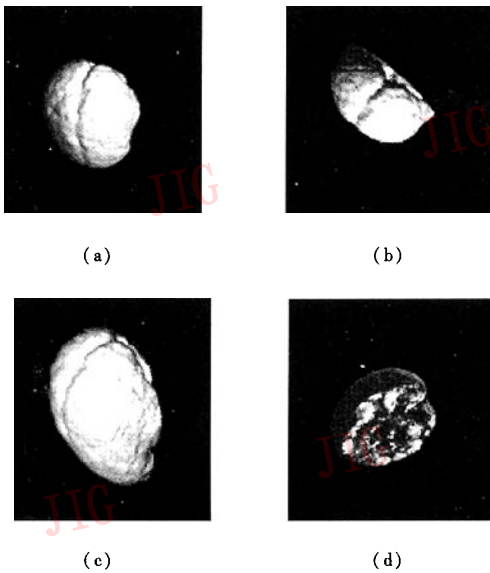


图 6 实际 3 维头颅图像成像结果

Fig.6 Example of head 3D reconstruction image

5 结 论

为了加快基于光线投影体绘制方法的 3 维图像重建的速度,本文在分析 3 维图像数据场同类组织具有相同灰度值的性质以及光线投影体绘制的重采样和插值特点的基础上,提出了减少插值运算、快速插值方法以及快速重采样等多项措施和方法,从而

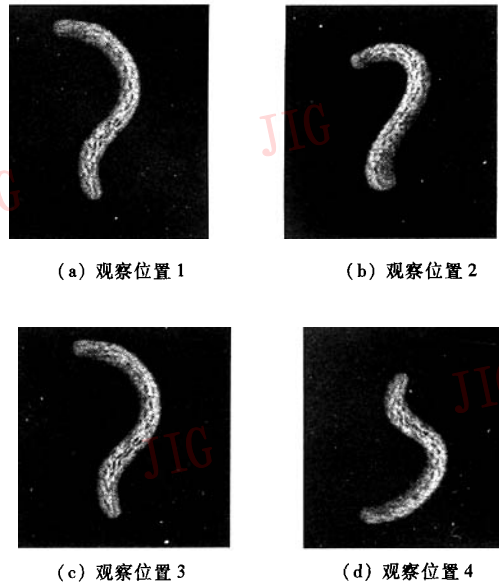


图 7 不同观察位置实际 3 维螺旋藻图像成像结果
Fig.7 Example of spiral alga 3D reconstruction image

大大加快了体绘制的绘制速度,已达到准实时性。这些方法在规则化图像数据和在平行投影情况下,重建时间大大缩短,而图像质量影响却很小(从主观质量评价来看),重建前后图像的峰值信噪比平均相差小于 1dB。可见这些方法在实际应用中是切实可行有效的。

参考文献 (References)

- 1 Nan W, Yang Y H, Pierson R. Three-dimensional surface reconstruction using optical flow for medical imaging [J]. IEEE Transactions on Medical Imaging, 1997, 16(5):630 ~ 641.
- 2 Barillot C. Surface and volume rendering techniques to display 3-D data[J]. IEEE Transactions on Engineering in Medicine and Biology Magazine, 1993, 12(1): 111 ~ 119.
- 3 Grevera G J, Udupa J K, Odhner D. An order of magnitude faster isosurface rendering in software on a PC than using dedicated, general purpose rendering hardware [J]. IEEE Transactions on Visualization and Computer Graphics, 2000, 6(4): 335 ~ 345.
- 4 Mroz L, Wegnekittl R, Groller E. Mastering interactive surface rendering for JAVA-based diagnostic applications [A]. In: IEEE Proceedings of Visualization and Computer Graphics [C], Salt Lake City, Utah, USA, 2000: 437 ~ 440.
- 5 Lin Chin-feng, Yang Don-lin, Chung Yeh-ching. A marching voxels method for surface rendering of volume data [A]. In: IEEE Proceedings of International Conference on Computer Graphics [C], Hong Kong, 2001: 306 ~ 313.
- 6 Amoroso A. Fast colored surface rendering for entertainment applications on small devices-an experimental study [A]. In: IEEE

- Global Telecommunications Conference, Globecom Workshops [C], Dallas, Texas, USA, 2004 : 536 ~ 539.
- 7 Dobashi Y, Cingoski V, Kanda K, *et al.* A fast volume rendering method for time-varying 3-D scale field visualization using orthonormal wavelets [J]. IEEE Transactions on Magnetism, 1998, 34 (5) : 3431 ~ 3434.
- 8 Lee W. Footprint evaluation for volume rendering [J]. Computer Graphics, 1990, 24(4) : 367 ~ 376.
- 9 Claudio T S, Kaufman A E, Pavlakos C. PVR: High performance volume rendering [J]. IEEE Computational Science and Engineering, 1996, 3(4) : 18 ~ 28.
- 10 Levoy M. Display of surfaces from volume data [J]. IEEE Computer Graphics and Application, 1988, 8(3) : 29 ~ 37.
- 11 Choong H L, Kyu H P. Fast volume rendering using adaptive block subdivision [A]. In: IEEE Proceedings of Computer Graphics and Applications [C], Seoul, Korea, 1997 : 148 ~ 157.
- 12 Shen H W, Chiang L J, Ma K L. A fast volume rendering algorithm for time-varying fields using a time-space partitioning (TSP) tree [A]. In: IEEE Proceedings of Visualization [C], San Francisco, CA, USA, 1999 : 371 ~ 545.
- 13 Nicoletti G M. Optimal generation of transfer functions for direct volume rendering system theory [A]. In: IEEE Proceedings of the 35th Southeastern Symposium on System Theory [C], West, Virginia, USA, 2003 : 367 ~ 371.
- 14 Ebara Y, Watashiba Y, Koyamada K, *et al.* Remote visualization using resource monitoring technique for volume rendering of large datasets [A]. In: IEEE Proceedings of International Symposium on Applications and the Internet [C], Tokyo, Japan, 2004 : 309 ~ 312.