频域体绘制中的颜色合成*

邓俊辉 唐泽圣

(清华大学计算机科学与技术系 北京 100084)

摘要 和传统体绘制算法相比,频域体绘制算法具有较低的算法复杂度,能够更快地完成对三维数据场的显示.然而这种方法却不能简捷有效地表示和处理数据场的深度信息,因此其生成的图象缺少深度的遮挡效果.有鉴于此,Levoy 等曾通过诸如线性深度补偿的技术来弥补这一缺憾.尽管如此,运用这种技术得到的结果和传统空域方法仍有明显的差别.本文首先着眼于对空域体绘制方法中广泛采用的离散颜色合成技术的分析,然后引出了透明度密度分布函数的概念,进而推导出连续形式颜色合成过程的解析描述.根据这个结果,只要对三维数据场的透明度分布加以适当限制,就能够通过频域体绘制方法来实现颜色合成,增加绘制的深度效果.作者最后指出,基本频域体绘制方法和 Levoy 的线性深度补偿方法的实质可以看成是这种颜色合成过程的近似.文中对透明度均匀分布的情况做了详细讨论,并且给出了具体算法,同时用生成的图象实例相应地进行了比较.

关键词 三维数据场可视化,颜色合成,深度补偿,频域体绘制.

1 概 述

1.1 空域体绘制

在三维数据场可视化领域中,体绘制是一类基本而重要的方法.这类方法的实质,可以概括为对三维数据场的重采样和颜色合成 2 大步骤[1,2],这 2 个计算步骤在迄今为止的绝大多数体绘制算法中都是完全在空间域中进行的.以光线投射(Ray—Casting)算法[3]为例,因为每次绘制都至少要对所有体元(Voxel)访问一遍,所以如果数据场的大小是 N^3,这类算法的复杂度将不会低于 $O(N^3)$.[4~6]许多人都曾尝试过运用各种技巧来减少空域体绘制的时间[7,8],但却无法从根本上超越 $O(N^3)$这个理论上的复杂度界限.因而从绘制速度方面看来,空域方法离实时要求还有相当距离.

1.2 基本 FVR 算法

频域体绘制 FVR(frequency domain volume rendering)算法的提出虽然不过短短 5 年,却已很快得到完善而日臻成熟[4~6],为体绘制技术走向实时系统提供了一个新的途径.这类方法的理论依据来源于多维信号的投影重构,[9~11]早期的研究工作开始于医学领域内的应

---

* 本文研究得到国家自然科学基金资助.作者邓俊辉,1970 年生,博士生,主要研究领域为科学计算可视化.唐泽圣,1932 年生,教授,博士导师,主要研究领域为计算机图形学,科学计算可视化.

本文通讯联系人:邓俊辉,北京 100084,清华大学计算机科学与技术系

本文 1996—03—04 收到修改稿

用.[4]后来 Tom Malzbender 在对这些算法雏形进行总结归纳的基础上,整理了其理论框架,并且提出了一个通用的算法.[5]本文称之为基本频域体绘制算法(Fundamental FVR Algorithm).

Tom Malzbender 的理论基础是傅立叶切片投影定理.[9~11]这个定理指出了可以从另一个途径来得到三维函数在某个方向上的投影:在此三维函数对应的频谱函数中,截取一个二维切片,如果这个切片是经过原点并且平行于投影平面,那么,其反变换就是原函数对应的投影.

根据 Tom Malzbender 构思,可以把三维数据场的体绘制结果粗略地看作为沿视线方向的投影,这样就可以按傅立叶切片投影定理指出的方法来得到这个投影.于是他设计了由2个步骤组成的基本频域体绘制算法.算法第1步进行预处理,计算三维数据场的频域表示.尽管对大小是 N^3 的数据场而言其复杂度将高达 $O(N^3\log N)$,但其计算结果可以中间结果的形式保存下来,所以对任何数据场这步计算只需进行1次.

在得到了这个中间结果之后,算法第2步可以重复执行:对应于某给定视线方向,在频域数据场中沿此方向截取一个过原点的切片,然后将其反变换至空域.这样,在整个体绘制算法的时间消耗中占主要部分的重采样过程只需在二维切片上进行,而不再如空域方法那样在整个三维范围内进行.这部分计算的复杂度由 $O(N^3)$ 降至 $O(N^2)$.考虑到反变换,整个算法的复杂度是 $O(N^2\log N)+O(N^2)=O(N^2\log N)$.

1.3 方向光照和物质分类

上面提到的基本 FVR 算法仅仅是一个示意的框架,还有许多地方需要改进才能达到实用.与传统的体绘制算法相比较,基本 FVR 算法最突出的一个缺陷在于没有空域体绘制的光照效果.为了弥补这个不足,Takashi Totsuka 和 Marc Levoy[6]基于半球形光源模型的假设,提出了方向光照(Directional Shading)的方法.

然而这种方法与传统的光照模型还是有明显不同的:在方向光照模型中,体元的颜色是取决于光源的颜色,而不是象传统光照模型一样取决于物质分类的结果.换言之,这种方法舍弃了被空域体绘制算法证明为行之有效的物质分类技术.我们的一些前期工作[12]解决了这个问题,把物质分类技术引入了频域体绘制方法,使得图象效果更加接近于空域方法.

1.4 线性深度补偿

基本 FVR 算法的另一个不足在于缺乏物体的深度信息或遮挡效果,原因在于这种方法把图象视为简单的投影,也就是沿视线方向的线积分.Takashi Totsuka 和 Marc Levoy[6]使用线性深度补偿技术试图弥补这一缺憾.但是,下面详细讨论后指出,由于线性深度补偿只是在一定的假设条件下对颜色合成过程的近似,所以其图象的真实感仍然较差.下面将要针对上述问题提出新的深度补偿算法,以使得 FVR 方法生成的图象有更强的真实感.

1.5 符号约定

本文中使用小写字母表示空域中的函数和变量,大写字母表示频域中的函数和变量;通常字母代表三维函数和变量,加有"^"的字母代表二维函数和变量.

2 空域体绘制中的颜色合成

2.1 颜色合成的离散形式

无论是体绘制(Volume Rendering)还是面绘制(Surface Rendering),传统的空域绘制算法普遍地采用了颜色合成(Color Composition)技术.[1]以体绘制典型的光线投射(Ray-Casting)算法为例,要从每个象素沿视线方向发出一条光线穿过数据场空间,象素的颜色取决于相应的光线与数据场的相交部分.实际对这段相交部分的计算过程是:沿此光线按固定间隔对数据场进行采样,得到一个数值序列.如果事先已经对数据场进行了物质分类,这个数值序列就可以转化为颜色属性的序列.颜色值用向量 $c=(r,g,b)$ 表示,其中 r,g 和 b 分别对应于红、绿和蓝颜色分量;另一个数值 α 是[0,1]之间的实数,对应于采样点的不透明度.离散形式的颜色合成是以迭代的方式进行的,每次迭代把当前采样点的颜色和不透明度累加到当前相应的累加值中,得到新的累加值.这种迭代可以按由前到后或由后到前的方向进行,前一种方向的迭代[2]可以描述为:

$$\begin{cases} c_{out}\alpha_{out}=c_{in}\alpha_{in}+c_{now}\alpha_{now}(1-\alpha_{in}) \\ \alpha_{out}=\alpha_{in}+\alpha_{now}(1-\alpha_{in}) \end{cases} \tag{1}$$

其中带"*now*"下标的变量是当前采样值,带"*in*"下标的变量是当前的累加值;带"*out*"下标的变量是新的累加值,将作为新的当前累加值传给下轮迭代.迭代的最后数值用 c_{final} 表示,它就是颜色合成的结果,可以用来设置象素的颜色 $\hat{c}$,　$\hat{c}=c_{final}$

为了简洁起见,引入新变量 $\begin{cases} p=c\alpha \\ \beta=1-\alpha \end{cases}$

这样就可以将(1)式重写为

$$\begin{cases} p_{out}=p_{in}+p_{now}\beta_{in} \\ \beta_{out}=\beta_{now}\beta_{in} \end{cases} \tag{2}$$

这里的 β 对应于采样或累加的透明度数值.这样,如果对应于某象素的累加值为 p_{final},这个象素的颜色就是 $\hat{p}=p_{final}/(1-\beta_{final})$.如果记沿视线方向的诸采样点为 $k_0,k_1,\cdots,k_{n-1}$,相应的颜色采样值为 $c_0,c_1,\cdots,c_{n-1}$ 和 $\beta_0,\beta_1,\cdots,\beta_{n-1}$ 可以由(1)式或(2)式推导出每根光线颜色合成的最终结果为

$$\begin{cases} p_{final}=\sum_{k=0}^{n-1}\prod_{t=0}^{k-1}\beta_t p_k \\ \beta_{final}=\prod_{k=0}^{n-1}\beta_k \end{cases} \quad \text{其中} \quad p_k=c_k(1-\beta_k) \tag{3}$$

如果把(3)式中的 $\prod_{t=0}^{k}\beta_t$, $k=0,\cdots,n-1$ 看作加权值,那么颜色合成就是各采样点采样值的线性组合,换而言之,由(1)式或(2)式定义的迭代过程就是对数据场的离散积分.

2.2　颜色合成的连续形式

和空域方法一样,FVR 方法将数据场看成是连续分布于三维空间中,并且这种分布是由某三维连续函数刻画的.在空域方法中这仅仅是插值重构的前提,而 FVR 方法则真正地依此前提进行计算.所以为了探讨在频域中实现颜色合成的途径,必须首先对连续形式的颜色合成做一讨论.

如果认为数据场是连续分布的,我们仍可以用函数 $c(x)$ 来表示颜色的分布.但是透明度的表示则与离散情况有所不同,这里要用透明密度分布函数(Transparency Density Distribution Function)来替换透明度函数(Trans-

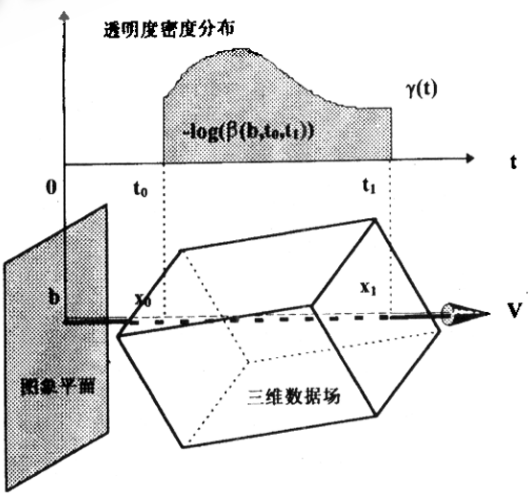

图 1　透明度密度分布函数

parency Function),下文中将用$\gamma(x)$表示前者.图1简单地介绍了透明密度分布函数的物理意义:如果数据场的透明密度分布由函数$\gamma(x)$描述,那么由x_0到x_1区间的透明度是该函数在此区间的线积分值的负指数函数,即

$$\beta(b,t_0,t_1)=exp(-\int_{t_0}^{t_1}\gamma(b+\tau V)d\tau)$$

其中V是视线单位法向,b是x_0x_1方向与图象平面的交点(如果x_0x_1是视线方向,b就是对应于此光线的象素),t_0和t_1分别是x_0和x_1到图象平面的距离.

在颜色合成中所考虑的区间都是起始于图象平面而且与视线方向一致的,即$t_0=0$,$(x_1-x_0)\times V=0$.这样,透明度只与x_1有关,透明度的定义和计算式为

$$\beta_v(x)=\beta_v(b(x)+t(x)V)=exp(-\int_0^{t(x)}\gamma(b(x)+\tau V)d\tau) \tag{4}$$

这个连续形式的透明度函数就相当于2.1节中讨论的离散透明度,它们都指示采样点的透明度或遮挡程度.由(4)式,对应于象素b的颜色合成可以表示为

$$\hat{p}_v(b)=\int_0^{\infty}p(b+tV)\beta(b+tV)dt=\int_0^{\infty}p(b+tV)exp(-\int_0^{t}\gamma(b+\tau V)d\tau)dt \tag{5}$$

其中 $p(x)=c(x)(1-\beta(x))$

在这个积分式中,透明度函数$\beta(b,t)$可以看作是个加权值,它随着透明密度分布函数积分值的增长而迅速下降,从而产生颜色合成的效果.所以,我们称这个加权函数为深度核(Depth Kernel).

3 频域体绘制中的颜色合成

3.1 基本FVR算法与颜色合成

在基本FVR算法中,各象素的颜色是数据场沿对应投射光线的连续线积分,这可以形式地描述为

$$\hat{p}(u)=\int_{x_0}^{x_1}p(x)dx \tag{6}$$

其中u是二维图象上的一个象素,$\hat{p}(u)$为其颜色值,$p(x)$为三维数据场的颜色分布函数,x_0和x_1是投射光线与数据场的交点.式(6)与式(5)的差别仅在于深度核.由于式(6)的深度核可视为常数1,也就是说基本FVR算法没有考虑深度,其生成的图象自然就没有遮挡效果.

3.2 线性深度补偿与颜色合成

如上面曾提到的,Takashi Totsuka和Marc Levoy[6]提出了线性深度补偿算法,这个算法的实质是按照下面公式进行频域线积分

$$\hat{p}(u)=\int_{x_0}^{x_1}p(x)(x\cdot V)dx \tag{7}$$

其中V为视线的单位法向.与式(6)比较,这里考虑到了采样点到图象平面的距离(采样点的位置向量和视线单位法向的点积),用此距离作为线积分的加权值.由于这个加权值沿深度方向线性递减,所以图象就产生了一定的遮挡效果.

3.3 指数深度补偿与颜色合成

如果能够将由(5)式定义的颜色合成在频域中实现,就可以大大改善FVR算法的深度效果.然而实际上由于其中由(4)式定义的深度核是一个任意的透明密度分布函数的线积分,设计这样一个通用算法是不可能的.所以为了使得这个线积分的计算能在频域中找到算法,我们可以对数据场的透明密度分布函数作一些假定.在此我们对一种简单情况进行讨论,即假设整个数据场的透明密度分布函数为常值,我们称这类数据场为均匀透明密度场.

尽管多数的三维数据场都是由多种物质组成的,在各种应用领域中仍然有相当多的数据场是由一种物质构成的.当然二者的物质分类方法也是有所不同的:对于前一类数据场,每一种物质的颜色和透明度都是互异的;而后一类数据场只是按某种属性来确定颜色,至于透明度则统一确定为固定值.例如计算流体力学 CFD(computational fluid dynamics)中考虑的流体场通常只有一种流质,因此其透明密度分布是各处同性和均匀的,同时可以按照某种物理量(密度、速度或梯度等)给不同位置赋予相应的颜色.又如在石油勘探领域,同一岩层是由一种岩质构成,在透明密度上是均匀的.为了找到油区的位置,要根据爆炸声波的幅值变化,所以可以这种按幅值来确定不同的示意颜色.

满足这个假设条件的数据场,其透明密度分布函数可以表示为　$\gamma(x)=R$　(8)

这里的 R 是一个实常数.原来由(4)式定义的深度核变成

$$\beta_v(x)=exp(-\int_0^{t(x)}\gamma(b(x)+\tau V)d\tau)=exp(-\int_0^{t(x)}Rd\tau)$$
$$=exp(-Rt(x))=exp(-R(V\cdot(x-b(x))))\quad(9)$$

原来由(5)式定义的颜色合成变成

$$\hat{P}_v(b)=\int_0^{+\infty}P(b+tV)\beta(b+tV)dt=\int_0^{+\infty}P(b+tV)exp(-Rt)dt\quad(10)$$

可以看到在上述假设条件下,这个计算过程很适于在计算机上实现.我们可以在对线性深度补偿进行修改后得到这样一个算法,由于这个算法使用一个负指数函数作为深度核,我们称之为指数深度(Exponential Depth Cueing)算法.

颜色分布为 $c(x)$ 的三维数据场,其对应的傅立叶变换记做 $C(s)=F(c(X))$

基本 FVR 算法从 $C(s)$ 抽取的二维切片为 $\hat{C}_0(s)=C(x)*H(s)$

其中 $H(s)$ 是频域重采样用的重构核.在线性深度补偿算法中,频域重构核被修改为

$$H_1(s)=F(h(x)d(x))=H(s)*D(s)$$

其中的 $d(x)$ 是空域中的重构核.指数深度算法中使用的深度核为

$$\beta_v(x)=\beta_v(b(x)+t(x)V)=exp(-\int_0^{t(x)}\gamma(b(x)+\tau V)d\tau)\quad(4)$$

对应的抽取平面为　$\hat{C}_1(s)=F(c(x)\beta(x))*H(s)$

根据卷积定理有

$$\hat{C}(s)=F(c(x)\beta(x))*H(s)=(C(s)*B(s))*H(s)=C(s)*(B(s)*H(s))=C(s)*H'(s)\quad(11)$$

其中　$B(s)=F(\beta(x))$, $H'(s)=B(s)*H(s)$ 是修改后的滤波器(如图 2).

根据(11)式,指数深度算法的流程可以设计为

```
对于某视线方向{
    用深度核 B(s)修改重构滤波器 H(s),得到新的重构核 H'(s);
    用这个被修改后的重构核 H'(s)对数据场重采样,得到二维切片 Ĉ1(s);
    将 Ĉ1(s)反变换到空间域;
}
```

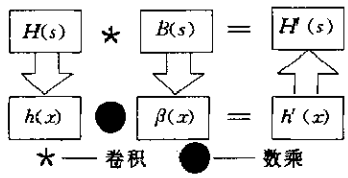

图2 负指数深度补偿的实现

这样,我们应用指数深度算法就能够实现频域颜色合成.尽管对于每个视线方向都要重新修改滤波器,但正如 Marc Levoy 指出的,由于滤波器的尺寸与数据场比较而言要小得多,所以这种方法在绘制速度上和基本 FVR 算法是近似的.

4　结论与展望

这个通过指数深度补偿完成频域颜色合成的算法已经在我们的 SGI indigo2 工作站上

实现.图4和图5给出了这个算法的计算实例,它们都是同一数据场(由红色球体和兰色平面组成)的绘制结果(因图片为黑白图,所以红色球体表现为白色圆圈,而兰色平面则表现为灰色方框).在图4中红色球体因在兰色平面后面而被部分遮挡;图5的视线方向与图4相反,可以把球在平面前面的效果与图4作一比较.我们也对不同数据场应用了此算法,图7给出了一个分子模型数据的例子.为了其显示深度效果,图6中给出了用基本FVR算法绘制的同一数据.

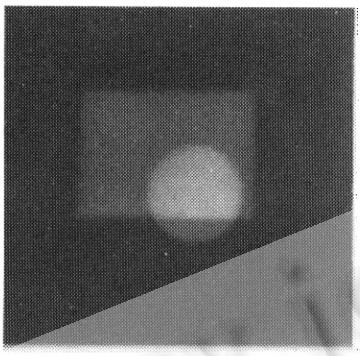

图4 红色球体在兰色平面后

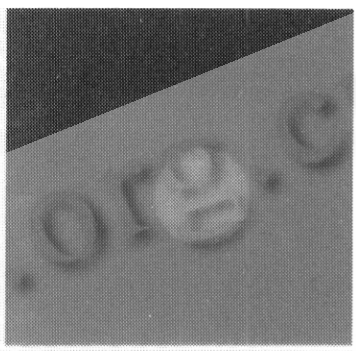

图5 红色球体在兰色平面前

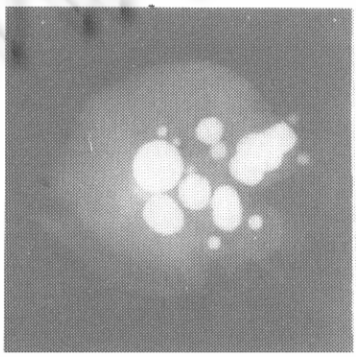

图6 分子模型(基本FVR算法)

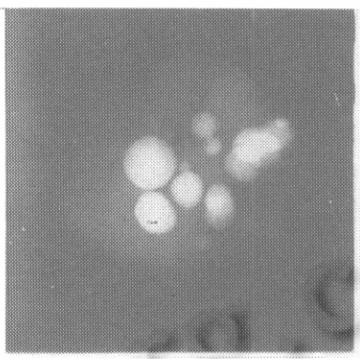

图7 分子模型(指数补偿的颜色合成)

我们可以将指数函数按Taylor级数展开为

$$exp(x)=1+\frac{x}{1!}+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots \tag{12}$$

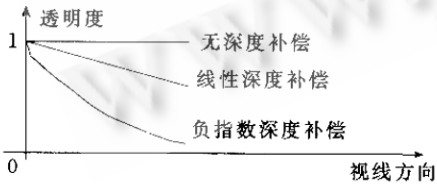

图3 空域中的深度核

基本FVR算法使用的深度核是一个常值函数,从这一点看这种算法只是指数补偿算法的零阶近似;而线性深度补偿的深度核是一个线性函数,也仅仅是指数补偿算法的一阶近似.总之,以往的频域算法都可以看成是在作了数据场透明密度均匀分布这一假设后,对颜色合成在不同程度上的近似.

在本文中介绍的算法虽然也作了相同的假设,但这并不表明只能局限于此.相反,只要所做的假设保证由(4)式定义的深度核能够参与频域计算,就是可以接受的.所以我们今后的工作将围绕构造更多的这类深度核并设计相应算法,进而使这类算法更加通用.

参考文献

1 Robert A Drebin, Loren Carpenter, Pat Hanrahan. Volume rendering. In: John Dill, ed., Computer Graphics, August 1988, **22**: 65～74.

2 Thomas Porter, Tom Duff. Compositing digital images. Computer Graphics (SIGGRAPH'84 Proceedings), 1984, **18**: 253～259.

3 Todd Elivins T. A survey of algorithms for volume visualization. Computer Graphics, August 1992, **26(3)**: 194～201.

4 Dunce, Napel S, Rutt B. Interactive display of volumetric data by fast Fourier projection. Computerized Medical Imaging and Graphics, 1992, **16(4)**: 237～251.

5 Tom Malzbender. Fourier volume rendering. ACM Transactions on Graphics, July 1993, **12(3)**: 233～250.

6 Takashi Totsuka, Marc Levoy. Frequency domain volume rendering. Computer Graphics Proceedings, Annual Conference Series, 1993. 271～278.

7 Koyamada K, Uno S, Doi A *et al*. Fast volume rendering by polygonal approximation. Journal of Information Processing, 1992, **15(4)**: 535～544.

8 Wilhelms J, Gelder A V. A coherent projection approach for direct volume rendering. Computer Graphics, July 1991, **25(4)**: 275～284.

9 Bracewell, Ronald. The Fourier transform and its applications. Revised second edition, McGraw—Hill, 1986.

10 Bracewell, Ronald. The Hartley transform. London: Oxford University Press, 1986.

11 Dudgeon D, Mersereall R. Multidimensional signal processing. Prentice—Hall, N.J., 1984. 81～82, 363～383.

12 Deng Z H, Tang Z S, Xu M H. Color assignment and boundary sharpening in frequency domain volume rendering. Proceeding of Pacific Graphics'95, 1995. 240～252.

COLOR COMPOSITION IN FREQUENCY DOMAIN VOLUME RENDERING

Deng Junhui Tang Zesheng

(*Department of Computer Science and Technology Tsinghua University Beijing* 100084)

Abstract The FVR(frequency domain volume rendering) approach explores a 3D data set faster than traditional spatial domain methods because of its innate lower computational complexity. Unfortunately, images produced by the FVR methods always lack occlusion or depth cue since it is hard to represent and utilize the depth information within frequency domain efficiently. Some researchers have sought to make a remedy of this by means such as of linear depth cuing. This approach, however, is still far different from the conventional spatial domain methods which, using the color composition technique, generate images with realistic depth effect. In this paper, the discrete color composition of spatial domain algorithms is reviewed, before an analytical result of the continuous color composition is deducted. These equations become feasible for calculating in frequency domain on certain assumptions such as the transparency inside the data set is distributed homogeneously. The authors continue to discuss the detail implementation of this new algorithm based on their previous work, the substance classification FVR approach. The experimental result shows that both the computational complexity and the run—time cost are kept at the same level of former FVR algorithms.

Key words Visualization, color composition, depth cue, frequency domain volume rendering.