

快速实现组合体绘制的索引数据块结构*

孙文武^{1**} 王文成¹ 吴恩华^{1,2}

1. 中国科学院软件所计算机科学开放实验室, 北京 100080; 2. 澳门大学科技学院, 澳门

摘要 对数据场中各种物质内容进行多种方式的组合并可视化, 是提高对感兴趣内容多方面认识的有效方法. 提出一种数据场的空间分布索引结构, 可以便捷地组合各种物质内容, 并在体绘制时只需对所选择的内容进行处理. 利用此结构, 可方便采用二分查找技术加快成像计算. 实验表明, 这种索引结构很好地提高了组合体绘制的效率.

关键词 数据结构 组合 体绘制 计算可视化

科学计算可视化的目的, 是用图像直观地揭示数据场中所蕴涵的内容, 提高人们处理数据的能力. 但随着数据量日益庞大, 绘制开销也越来越大. 为此, 人们提出了选择可视化^[1~3]和特征可视化^[4]等方法, 以便只处理感兴趣内容, 提高可视化效率. 一般而言, 为了解各种内容的情况及其相互关系, 人们需要对各种内容进行多种方式的组合绘制, 即对数据场进行多次成像, 而每次绘制的内容是变化的. 在可视化的多种方法中, 体绘制方法^[5]能够体现多种内容的综合分布情况, 也是人们研究的重点. 在此, 本文讨论如何快速实现各种内容组合的体绘制. 对此, 如何组织感兴趣内容以尽可能避免对无兴趣内容的访问, 是决定组合体绘制效率的关键.

然而, 根据已有文献资料, 目前各种提高绘制效率的方法^[6~8], 基本上不能高效地进行组合体绘制. 这是由于组合体绘制需要经常改变绘制内容, 从而使得已有方法在每次成像时都要重新组织数据. 为此, 本文提出一种新的数据组织方式, 以方便组合绘制的实现. 其基本思想为: 首先为每类内容分别建立其空间分布的索引结构; 然后, 绘制时, 算法只访问所选择内容的索引结构以避免对无兴趣内容的计算; 当然, 对多种物质成像时, 算法根据这些索引结构, 也易判知它们各自数据间的前后关系, 保证了绘制过程的效率.

1 索引数据块结构

本文讨论规则数据场, 并假设其坐标系的 X 轴从左往右、 Y 轴由下向上、 Z 轴由前往后.

在进行组合体绘制之前, 本文分别建立每类数据的分布索引结构. 为此, 建立一类数据的索引结构时, 本文依次遍历平行于 XY 面的各个体切片, 并将该类数据在各切片中的每一行体

2000-07-19 收稿, 2000-11-24 收修改稿

* 国家自然科学基金资助项目(批准号: 69903009)

** Email: sww@ios.ac.cn

元信息组织成一个棒(stick)结构表^[9]. 其中,每一个结构元素表示该类数据的一段连续分布,由此段体元的最小和最大 x 坐标值标志. 每一体元行上的棒结构按其 x 坐标值有序排列. 之后,不同于文献[9]中的处理,同一切片中的这些体元行按其 y 坐标值组织成 y 桶结构. 同理,包含该类数据的各个切片按其 z 坐标值组织成 z 桶结构. 这样,就建立了关于该类数据分布的索引结构,如图1所示.

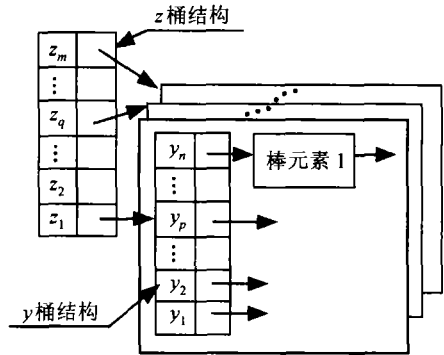


图1 一类数据的空间分布索引结构

由于这种索引结构是有序组织的,光线投射体绘制方法可以运用二分查找法,加快为各条光线查找相交体元的计算. 由于数据分布是不规整的,从而这使得绘制算法将增加二分查找时的迭代次数. 为提高数据组织的紧凑性,加快二分查找速度,本文将整个数据场沿 X 轴方向进行板块(slab)划分,然后再对每一块分别建立上述的索引结构. 块的划分,既要尽可能减少二分查找法的运用次数,同时也不宜分割太多,以免增加考察各个数据块的次数,降低利用索引结构的效率. 如图2所示,根据同类数据样点相对于 X 轴的分布,分割位置将选在 x_1 和 x_2 处. 相应地,分割后各块中数据的分布情况类似如图3所示,粗黑线段表示数据在此位置的分布.

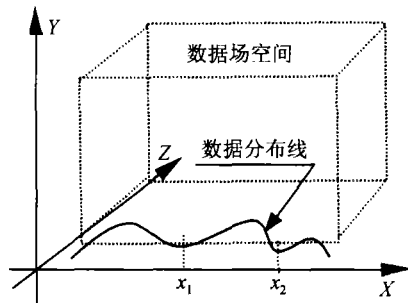
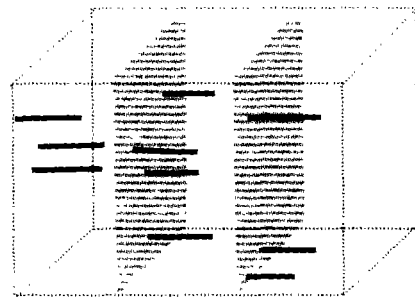
图2 根据一类数据相对 X 轴的分布而进行的块剖分位置选取

图3 数据在各板块中的分布情况

2 基于索引数据块的组合体绘制

体绘制方法大致可以分成两类,即光线投射方法和投影成像方法. 其中,光线投射方法就是从屏幕上各像素发出光线穿越数据场,将各条光线所获得的颜色值作为相应像素的色彩. 而投影成像方法则将数据体元依次投影到屏幕上,并合成这些体元对图像的贡献,生成结果图像. 基于上节所介绍的索引结构,这两种体绘制方法都可实现. 比如,投影成像方法就只需将选定内容的各个棒结构表中所记录的体元,按照前后顺序依次处理即可. 下面,本文讨论光线投射方法的实现.

为此,本文首先讨论对一种内容的体绘制实现,然后再讨论多种内容组合的体绘制.

根据光线当前位置,求取光线前进方向上下一个将要相交的体元,是决定绘制过程效率的

关键. 因此,当选择一种内容进行体绘制时,本文采用了一查找算法(见附录),以快速准确地找到下一个相交体元.

至此,我们讨论了一种内容的光线投射体绘制实现. 而对多种内容的体绘制实现,算法运用上面讨论的方法先分别找出各种内容中离光线当前位置最近的体元,然后再从中选出距离最近的体元. 之后,算法合成这个体元相关内容的色彩,并将光线前进到这个位置. 依此进行,就实现了对多种内容的组合体绘制.

3 实验结果

我们在一台 CPU 400 MHz 的奔腾 II 微机上实现了本文所讨论的组合体绘制方法. 此微机内存大小为 128 MB, 操作系统为 WINDOWS NT. 在此,本文采用了平行光线投射方式,生成图像大小为 400 像素 $\times$ 400 像素. 作为比较,我们也实现了绘制速度较快的方法^[7]. 该方法首先将数据体元分为有兴趣和无兴趣两类;然后,对任一无兴趣体元,也就是空体元,再利用三维距离变换算法,计算此空体元到感兴趣体元之间的最近距离,并将此值记录在空体元的数据项中. 这样,当光线到达此空体元位置时,光线就可跳跃一个存于此空体元中的距离步长. 实验中,我们检测了两组数据,一个是关于树獭的 CT 数据,数据量为 $128 \times 128 \times 197$,另一个是关于人体头部的 MRI 数据,数据量为 $150 \times 200 \times 192$.

对于树獭数据,本方法在预处理时将其划分为 20 个数据块,为它们建立的索引结构占用 396 K 内存空间,处理时间为 3.77 s. 而文献[7]中的方法计算此数据场中每一空体元的跳跃距离值,所花时间为 7.42 s. 之后,实验选择了 3 种组合方式,即:骨骼、软体组织以及骨骼与软体组织的混合体. 绘制图像如图 4 所示,其中图 4(a)为骨骼图像;图 4(b)为软体组织图像;图 4(c)为骨骼与软体组织混合体图像. 同样,对于人头数据,本方法在预处理时将其划分为 5 个数据块,为它们建立的索引结构占用 405 K 内存空间,处理时间为 9.32 s,而采用文献[7]中的方法的预处理时间为 16.31 s. 之后,实验选择了 2 种组合方式,即:脑硬质部分、整个脑部. 绘制图像如图 5 所示,其中图 5(a)为脑硬质部分图像;图 5(b)为整个脑部图像. 表 1 给出了对上述两组数据的组合绘制时间,以及文献[7]的方法对上述内容的测试时间. 其中,文献[7]中的方法对感兴趣内容进行成像时,需要分辨所交的非空体元内容.

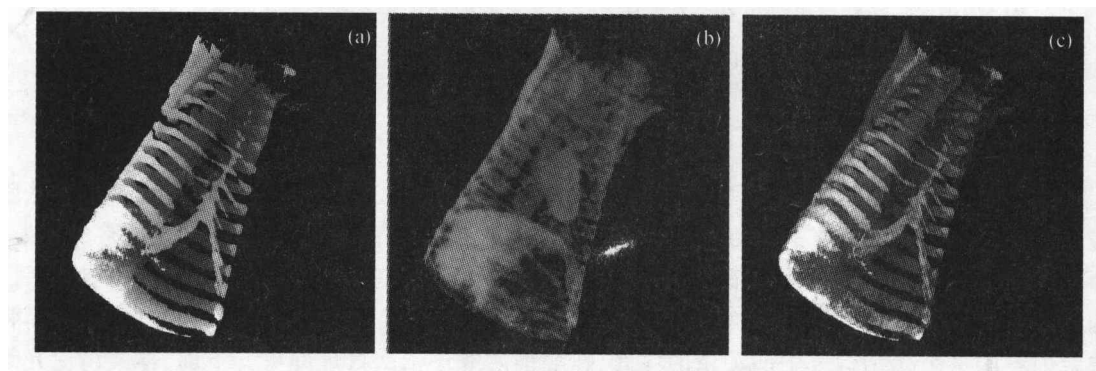


图 4 树獭体绘制图

从实验结果可知,新方法仅需占用很少的内存空间以存储各类内容的索引结构,并且明显

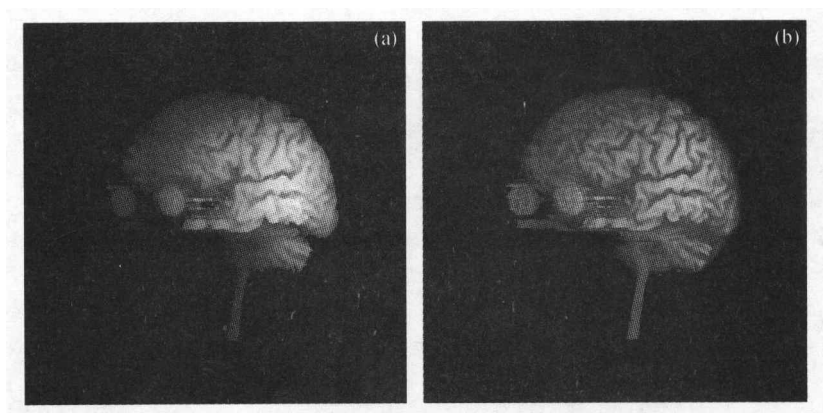


图5 人脑体绘制图

提高了成像速度. 特别是在穿越无兴趣区域时,与文献[7]中的方法相比,新方法的光线所走的步数要少得多. 其中,表中的加速率 = $(t_2 - t_1)/t_1$.

表1 新方法 with 文献[7]中的方法的测试对比结果

数据	图像	新方法		文献[7]的方法		加速率/%
		成像时间 t_1/s	平均每根光线穿越无 兴趣区域所走的步数	成像时间 t_2/s	平均每根光线穿越无 兴趣区域所走的步数	
树獭	图 4(a)	12.3	4.56	17.9	9.42	36.6
	图 4(b)	10.4	14.11	20.5	30.23	97.1
	图 4(c)	15.2	4.66	17.1	11.51	12.5
人脑	图 5(a)	16.0	3.31	33.0	8.11	106.3
	图 5(b)	27.6	0.91	33.9	5.65	22.8

4 结论

为快速实现数据场的组合体绘制,本文提出了一种有效的数据索引结构. 基于这种结构,绘制算法不仅可以便捷地实现各种内容的任意组合,而且在成象时只需处理所选择内容的数据. 这种索引结构占用的内存空间很少,并且其相关的预处理时间也不长. 这对提高大规模数据场的可视化效率很有益.

在可视化中,数据场分类可能需要经常改变. 为此,我们可以为每一种数据值建立其索引数据块结构. 然后,根据感兴趣内容的数值范围挑选相关数据,进行绘制.

参 考 文 献

- 1 Walsum T V, et al. Selective visualization of vector fields. Computer Graphics Forum, 1994, 13(3): 339
- 2 Wang W C, et al. A selective rendering method for data visualization. The Journal of Visualization and Computer Animation, 1999, 10(3): 123

- 3 Ihn I, et al. Indexing data structures for fast volume rendering. *Computer & Graphics*, 1997, 21(4): 497
- 4 Silver D, et al. Tracking scalar features in unstructured datasets. In: *Proc of IEEE Visualization '98*, New Jersey: IEEE Computer Society Press, 1998, 79
- 5 Lacroute P, et al. Fast volume rendering using a shear-warping factorization of the viewing transformation. In: *Proc of SIGGRAPH '94*, New York: Association for Computing Machinery, 1994, 451
- 6 Wilhelms J, et al. Octrees for faster isosurface generation. *ACM Transactions on Graphics*, 1992, 11(3): 201
- 7 Cohen D, et al. Proximity clouds-an acceleration technique for 3D grid traversal. *The Visual Computer*, 1994, 11(1): 27
- 8 Parker S, et al. Interactive ray tracing for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 1999, 5(3): 238
- 9 Montani C, et al. Rendering volumetric data using STICKS representation scheme. *Computer Graphics*, 1990, 24(5): 87

附录

```

found = 0; //表明未找到下一个相交的体元
while(! found)&&(光线在当前数据块空间中))
{if(当前数据块从未被查找过)
    找到与光线相交的最近切片置为当前切片;
while(! found)&&(光线在当前切片空间中))
{if(当前切片从未被查找过)
    查找与光线相交的最近体元行置为当前体元行;
while(! found)&&(光线在当前体元行空间中))
{if(当前体元行从未被查找过)
    将光线与该体元行空间相交的入点置为
    光线当前位置,其  $x$  坐标值  $x_i$ ;
    运用二分查找法,在当前体元行的棒结构
    表中,搜索  $x$  坐标区间包含  $x_i$  的棒结构元素;
if(搜索到一个棒结构元素)
    {该入点所在体元为所求结果;
    found = 1;}}
if (! found)
    {根据光线与该体元行空间相交体元行
     $x$  坐标值,判定当前体元行中离当前
    位置最近的体元是否为相交体元;
    if(最近体元为相交体元)
        found = 1;}
    将下一体元行置为当前体元行;|
    将下一切片置为当前切片;|
    将下一数据块置为当前数据块;|
}
}
}

```