

八元树三维表示法的磁盘存储与恢复

赵海峰¹, 孔敏^{1,2}, 罗斌¹

(1. 安徽大学 计算机科学与技术学院, 安徽 合肥 230039;

2. 皖西学院 机械与电子工程系, 安徽 六安 237012)

摘要:八元树是一种优秀的三维空间表示方法。从序列断层图像或者三维数据场建立八元树的三维表示是一个比较费时的复杂过程。为了省略每次从原始数据重新构造八元树的过程, 将内存中已经建立好的八元树结构以文件的方式保存到磁盘上。当再次做八元树的三维处理应用时, 再直接从保存的磁盘文件恢复出八元树。文中提出了一种快速有效的八元树三维表示法的磁盘存储与恢复算法, 并实验验证了其可行性。

关键词:八元树; 磁盘存储; 恢复; 八元码; 节点; 体素

中图分类号:TP391.41

文献标识码:A

文章编号:1673-629X(2006)08-0176-03

Disk Storage and Restoration of Octree for 3D Representation

ZHAO Hai-feng¹, KONG Min^{1,2}, LUO Bin¹

(1. School of Computer Science and Technology, Anhui University, Hefei 230039, China;

2. Machine and Electronic Engineering Department, West Anhui University, Luan 237012, China)

Abstract: It is an excellent way to use octree for representing three-dimensional (3D) space. It is also a time-consuming complicated progress to build 3D representation of octree from slices or 3D data field. In order to omit the process of reconstructing octree from original data every time, need to store the structure of octree in RAM to harddisk by file. For another application of octree, can restore the octree from the saved file. So presents an efficient algorithm of disk store and restoration of octree, and its feasibility is proved by experiments.

Key words: octree; disk storage; restoration; octcode; node; voxel

0 引言

八元树 (Octree) 表示法^[1-7]作为一种简洁、紧凑、直观的基于体积的三维表示方法, 近年来被广泛地应用到计算机视觉的研究领域。成熟的树的遍历算法更为物体八元树表示法的许多操作提供了便利的条件。其本身所具有的层次化数据结构、空间易达性以及存储数据的预分类性等特点又极大地简化了该表示法的实际运用, 例如处理遮挡表面、消影、连接区域、近邻搜索等操作。此外, 由于八元树编码表示与空间位置能够实现一一对应, 因而便于图像理解时判别边、角、面的相邻位置关系。同时, 利用八元树的分层结构还可以达到任意要求的精度, 逼真地还原出三维物体。

利用八元树进行物体的三维重建的第一步是从序列断层图像中构造出八元树, 然后再进一步处理。已经构造好的八元树以一种非线性的结构被存储在计算机的内存

中, 它能够表达物体的三维结构。但是, 由于从许多序列断层图像文件或者三维数据场文件读取并计算三维数据之间的关系 (如密度梯度等), 建立完整的八元树三维表示是一个比较费时的复杂过程。因此, 考虑到可以将第一次所建立的八元树以磁盘文件的方式保存, 从而省略每次从原始数据重新构造八元树的过程, 并且可以直接从保存的磁盘文件快速恢复出八元树, 进一步完成基于八元树的三维处理应用。文中提出了一种快速有效的八元树三维表示法的磁盘存储与恢复算法, 并且用实验验证其可行性。

1 八元树和八元码

八元树是一种分层的树结构, 其每一个节点定义为一种递归分解的三维立方体空间。为了处理上的方便, 对每次分解的相邻 8 个节点进行排序 (0, 1, 2, ..., 7), 如图 1 所示。

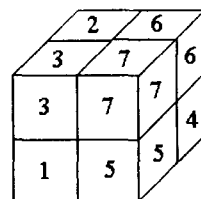


图1 八元树节点对应的三维立方体

收稿日期: 2005-11-15

基金项目: 国家自然科学基金项目 (60375010); 教育部“优秀青年教师资助计划”资助项目 (教人司[2003]355); 安徽省教育厅自然科学基金项目 (2006KJ053B)

作者简介: 赵海峰 (1972-), 男, 安徽舒城人, 讲师, 博士, 研究方向为图像处理、模式识别。

八元树的根节点表示整个空间,其余的节点都反映对空间递归分割的结果,叶节点是对整个空间最细分割的单元立方体。树的每个节点包括一个或更多的关于物体体积的特征值,例如最简单的节点只有一个特征——占空性,它有3个可能的值是 E 、 P 和 F ,其中 E 表示物体不占据这个空间,与此空间不相交; P 表示物体部分占据这个空间,与此空间部分相交; F 表示物体占据全部空间与此空间全部相交。 E 和 F 节点定义的物体与空间相互关系不能再进行进一步的分割,它们都是叶节点,而 P 节点称为非终结的树枝节点,对于 P 节点的空间对应关系是由其后代进一步定义的。例如,图2左图表示物体与整个空间的位置,而用八元树表示这一位置关系如图2的右图所示。

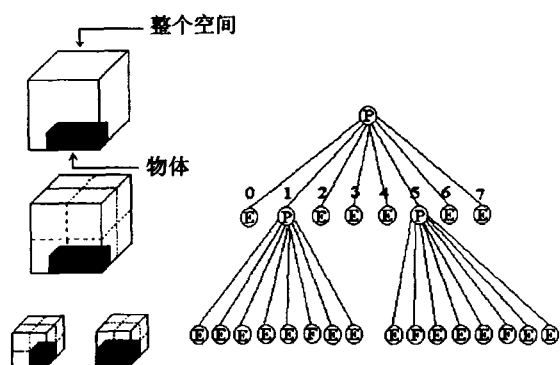


图 2 八元树与物体分布的空间对应

八元树表示三维物体的关键问题是找出断层图像中的像素与八元树中的节点一一对应的关系。由于八元树对空间分割的递归性,像素在八元树中每层的位置都是确定的(0,1,2,...,7)。例如,假设整个空间中只有一个像素位于空间的一个顶角,如图3所示。

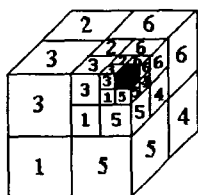


图3 仅有一个像素位于空间的顶角

整个空间对应的八元树,如图 4 所示。显然,经过空间的三次八元分割,每次它都处于第八个部分,对应的是八元树中每层的 7 节点。因此,这个像素对应应在八元树中

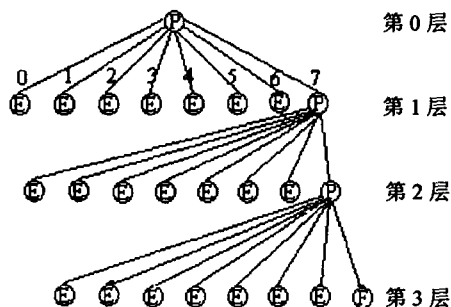


图 4 图 3 所对应的八元树

的位置可以用代码 777 表示, $O = O_1 O_2 O_3 = 777$, 把它称为八元码。通过八元码就可以确定空间物体的像素在八元树中的位置, 同样也可以从八元树中的节点找到对应空间物体的像素。

2 八元树的磁盘存储

因为八元树是一种非线性结构,所以在计算机内存中是用链式存储的方式。但是在计算机中文件存储信息的方式只能是线性的数据流。八元树是度为 8 的树形结构,可以采用先根遍历顺序对八元树进行遍历,从而将所有的数据变成一个数据流,完成磁盘文件的线性存储。同时,为了能够从磁盘文件正确地恢复出所存储的八元树结构,还需要考虑在磁盘上文件的存储方式。经过分析,两种形式可以用于八元树的存储。

形式一:在存入磁盘时,需要保存中间节点和叶节点的区分信息,以及叶节点的数据信息。在存盘时可按下述结构(C语言描述)存入:

```
typedef struct { unsigned char status;
                unsigned char data;
                } JD;
```

其中 status 为访问节点的状态(0 或 1),0 表示访问的是中间节点,1 表示访问的是叶节点;data 为数据信息,对叶节点来说为一具体值,例如可以存入颜色、亮度、密度信息等,对中间节点而言其值为空。

形式二:在存入磁盘时,需要保存八元码和叶节点数据信息。在存盘时可按下述结构(C语言描述)存入:

```
typedef struct { unsigned char o_code[8];
                unsigned char data;
                } JD;
```

其中 `o_code[8]` 中存放的是八元码。

形式二与形式一比较可以看出,因为所存储的节点不包括中间节点,所以所要存入的节点数大大减少,但是存入磁盘的每个节点信息要比形式一多。

具体的存储算法如下:

- (1)形式一:

- ①初始化:

- ②建立数据文件 filename:

- ③先根遍历八元树,判断条件,若为中间节点,则存入磁盘节点信息 status=0,继续先根遍历其子树;若为叶节点,则存入磁盘节点信息 status=1 和叶节点信息 data;

- ④八元树遍历结束,关闭数据文件,存储过程结束。

- (2)形式二与形式一存储算法类似,只是不需要存储中间节点,仅将叶节点的八元码和数据信息存入文件中。

3 八元树的磁盘恢复

恢复存储文件中的八元树,即它在内存中的链式结构,通过读取文件信息对八元树进行相应的操作,包括开辟节点空间、存入数据信息直到建立一个完整的八元树,

应该与原来的八元树一样。对于形式一,在恢复时根据 $status=0$ 或 1 判断是否为中间节点或叶节点,若为中间节点,则开辟 8 个孩子节点,直至叶节点,即按照它的存入的顺序来构造八元树。对于形式二,根据每个存入叶节点的八元码来恢复八元树。

具体的恢复算法如下:

(1)形式一:

①初始化;

②打开数据文件 filename;

③若至文件尾则转⑤,否则转④;

④读入磁盘中一个节点的信息,判断取出的状态信息 $status$ 是 0 还是 1,若 $status=0$,则开辟 8 个孩子节点空间,且指针指向其孩子节点;若 $status=1$,则取出的数据信息存入当前所指的节点中,重复前面的步骤;

⑤关闭数据文件,恢复过程结束。

(2)形式二是通过读入文件中的存储信息,根据读出的八元码开辟节点空间,按照树的先根遍历方法和八元码 $O_{L-1}O_{L-2}\cdots O_i\cdots O_0$ 访问八元树中各层的节点,当遇到所访问的节点不存在时创建一个新的节点,直到访问到树的最低层,这个节点必然是叶节点,把该节点的数据域用文件中的数据代替。

文中实验使用头部的 109 幅核磁共振断层图像(尺寸为 256×256),如图 5 所示(其中 6 幅)。取阈值为 25,即图像上灰度值大于 25 的点都被取为物体的体素。总共建立了 266708 个 Octree 型节点(叶节点包括原来的图像亮度信息和密度梯度信息),用形式一所存储的磁盘文件大小为 3 733 912 个字节,用形式二所存储的磁盘文件大小为 4 978 554 个字节。程序用磁盘文件恢复八元树的时间

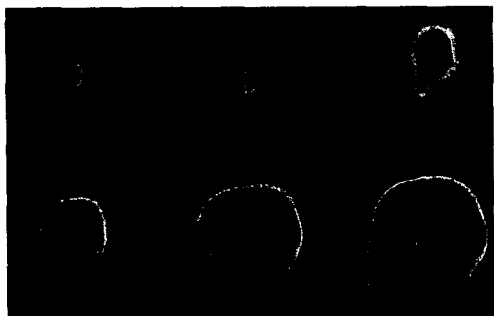


图 5 人头部的 109 幅 MRI 图像的其中 6 幅

约为第一次建立八元树时间的十分之一,因为第一次建立八元树需要计算八元码和每个叶节点的密度梯度信息,比较花费时间。利用恢复的八元树进行三维显示如图 6 所示,从得到的实验结果可以看出是正确的。

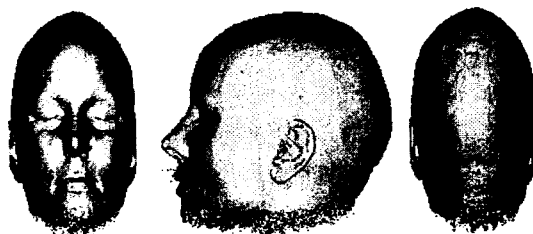


图 6 人头三维显示的结果

4 结 论

文中根据八元树空间表达的特点,提出一种十分有效的磁盘存储和恢复算法。因此,在基于八元树的三维重建过程中,不再需要从原始数据计算生成八元树,而是快速地从磁盘文件恢复出八元树来进一步处理。文中编程实现了八元树的两种形式的磁盘存储和恢复,获得了正确满意的结果。

参考文献:

- [1] Boada I. An octree - based multiresolution hybrid framework [J]. Future Generation Computer Systems, 2004, 11 (20): 1275 - 1284.
- [2] Yemez Y, Schmitt F. Multilevel representation and transmission of real objects with progressive octree particles [J]. IEEE Transactions on Visualization and Computer Graphics, 2003, 9 (12): 551 - 569.
- [3] Meagher D J. Geometric modeling using octree encoding [J]. computer graphics and image processing, 1982, 19: 129 - 147.
- [4] 赵海峰, 罗 斌. 一种改进的八元树三维目标表示方法 [J]. 计算机工程与应用, 2005, 41 (29): 8 - 10.
- [5] 赵海峰, 束学斌. 一种有效的序列断层图像的八元树构造算法 [J]. 中国图像学报, 1999 (5): 418 - 422.
- [6] 刘成君, 戴汝为. 广义线性八元树表示及物体的广义三维重建 [J]. 自动化学报, 1997, 23 (5): 694 - 697.
- [7] 罗 斌, 汪炳权. 基于 PC 机的断层图像序列 3D 表面重建 [J]. 电子科学学报, 1993, 15 (1): 75 - 78.

(上接第 175 页)

的研究中可以考虑应用数据挖掘的技术来确定权重。

参考文献:

- [1] 杨善林, 倪志伟. 机器学习与智能决策支持系统 [M]. 北京: 科学出版社, 2004.
- [2] Liao T W, Zhang Z, Mount C R. Similarity measures for retrieval in case - based reasoning systems [J]. Applied Artificial Intelligence, 1998, 12: 267 - 288.
- [3] 汤国安. 地理信息系统 [M]. 北京: 科学出版社, 2000.

- [4] 叶嘉安. 基于案例的推理和 GIS 相集成的技术在规划申请审批中的应用 [J]. 城市规划会刊, 2001 (3): 35 - 36.
- [5] 杜云艳. 地理案例推理及其应用 [D]. 北京: 中国科学院, 2001.
- [6] Jone. Model - Based case adaption [A]. In Proc of AAAI - 92 [C]. San Jose, CA: [s. n.], 1992. 673 - 678.
- [7] Holt A, Benwell G L. Applying Case - based reasoning technique in GIS [J]. IJGIS, 1999, 13 (1): 9 - 25.