

文章编号:1671-8860(2007)04-0350-03

文献标志码:A

基于 Quaternary 编码的球面三角格网邻近搜索算法

孙文彬¹ 赵学胜^{1,2}

(1) 中国矿业大学(北京)资源与安全工程学院,北京市海淀区学院路丁 11 号,100083)

(2) 国家基础地理信息中心,北京市海淀区紫竹院百胜村 1 号,100044)

摘要:分析了球面 Quaternary 编码的特点,给出了三角格网方向判断的规则,并对该算法与 Bartholdi 邻近搜索算法的时间复杂度进行了对比。结果表明,分解算法的平均时间耗时量约是 Bartholdi 邻近搜索算法的 23.66%。

关键词:Quaternary 编码;邻近搜索;球面三角格网

中图法分类号:P208

在全球离散格网的多尺度数据管理中,格网编码的邻近搜索是空间聚类、索引^[1]、范围查询^[2]、动态扩张^[3,4]等空间操作的基础,已成为全球离散格网研究中的关键问题之一。现有的球面三角格网编码有 Dutton 编码^[5,6]、Goodchild 编码^[7]、Fekete 的球面四叉树编码(SQT)^[8]、Otoo 的半四叉树编码(SQC)^[9]、Quaternary 编码^[1]等。这些编码都具有很好的层次性和嵌套性,但其中只有 Quaternary 编码是连续的^[1]。然而,Quaternary 编码的邻近搜索建立在空间坐标计算的基础上,涉及大量的浮点运算,计算复杂,从而降低了与邻近相关的空间操作效率。本文在分析 Quaternary 编码特点的基础上,提出了分解邻近搜索算法。

1 分解邻近搜索算法的原理

1.1 Quaternary 编码的特性

Quaternary 编码是规则编码,具有一些有助于编码邻近搜索的特性。这些特性主要包括:
① 根据编码中含有“1”的个数可以判断出三角形的方向性,如果编码中含有奇数个“1”,则此三角形是(向)下三角形,其余情况的三角形都是(向)上三角形。
② 如果两个三角形格网邻近,则这两个三角形的父三角形相同或邻近。
③ 整个层次剖分过程中,一个父三角形格网剖分成的四个子

三角形格网编码只有四种情况,如图 1 所示。

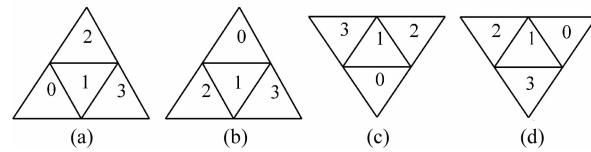


图 1 子三角形的四种不同编码顺序

Fig. 1 Four Different Ordering of Sub-triangle

1.2 邻近搜索的原理

在分解邻近搜索中,首先根据邻近格网附加码间的规律,建立对应的编码变换规则;再根据这些规则和原格网编码逐步分解出的附加码推求该格网的邻近编码。由性质③可知,子三角形的编码由父三角形编码和附加码组成,子三角形的附加码存在着四种编码方案,不同的编码方案导致了邻近格网编码间的规律不同,因而需根据三角格网的方向和编码中“3”个数的奇偶性将邻近格网的搜索分成四种情况,再对其进行一一处理。

分解邻近搜索的具体方法是按两位数字长度对编码由后到前依次进行分解,位数是奇数的编码在编码前增添一位数字 0,将编码位数补齐为偶数,然后对分解出的后两位编码进行编码变换,并判断该格网和它的邻近格网的二级父三角形格网是相同的还是邻近的,如相同,则停止编码变换;若不同,则重复上述过程,直到所有分解后的编码处理完毕为止。

2 编码变换的规则

对常用的符号和运算作一些约定, Code 表示原格网编码, Code-1 是按四进制对编码进行减 1 运算后得到的编码, Code-2 是减 2 运算后的编码, Code+1 是加 1 运算后的编码, Code+2 是加 2 运算后的编码, “ $<ab$ ”运算是将原编码的最后两位变成 ab , 其余部分保持不变, “ $>ab$ ”运算是将原编码的最后两位变成 ab , 其余部分变成二级父格网对应的邻近格网编码。假设原格网的编码为 Code, 则它的邻近格网编码可以表示为 Neighbor1、Neighbor2、Neighbor3, 具体情况如表 1 所示, 原格网与其邻近格网之间的关系如表 2 所示。

表 1 编码的变换规则

Tab. 1 Rules of Code Transformation

	编码 Code	Neighbor1	Neighbor2	Neighbor3
以 0 结尾	以 1 结尾	Code-1	Code+2	Code+1
	10 结尾	Code-1	Code<32	Code+1
	20 结尾	Code-1	Code>02	Code+1
	030 或 130	Code>03	Code<12	Code+1
	300	Code>33	Code>22	Code+1
	100、200、000	Code-1	Code>22	Code+1
以 2 结尾	02 结尾	Code-1	Code>20	Code<13
	12 结尾	Code-1	Code<30	Code<23
	22 结尾	Code-1	Code>00	Code>33
	132 或 232	Code-1	Code<10	Code>23
以 3 结尾	103 或 203	Code+1	Code-2	Code>30
	13 结尾	Code<20	Code-2	Code<02
	023 或 123	Code>32	Code-2	Code<12
	133 结尾	Code>22	Code-2	Code+1

表 2 对应的邻近关系

Tab. 2 Neighbor Relations

编码中 Code 含的个数	左邻近	右邻近	上/下邻近
奇数个“1”和偶数个“3”	Neighbor1	Neighbor2	Neighbor3
奇数个“1”和奇数个“3”	Neighbor3	Neighbor2	Neighbor1
偶数个“1”和偶数个“3”	Neighbor2	Neighbor3	Neighbor1
偶数个“1”和奇数个“3”	Neighbor2	Neighbor1	Neighbor3

3 算法的效率分析与实验

3.1 算法的效率分析

按分解邻近算法可以求出大部分格网的邻近编码, 然而并不是所有的邻近 Quaternary 编码都具有固定的规律。分解邻近搜索算法在下列情况时不适用: 以“30”结尾, 且倒数第三位编码为 2 或者 3 的情况; 以“32”结尾, 且倒数第三位编码为 0 或者 3 的情况; 以“03”结尾, 且倒数第三位编码为

0 或者 3 的情况; 以“23”结尾, 且倒数第三位编码为 2 或者 3 的情况; 以“33”结尾, 且倒数第三位编码为 0、2 或者 3 的情况, 总共有 11/64 的情况不能应用分解邻近算法进行邻近搜索。在这些情况下, 用 Bartholdi 算法替代该算法。

假设进行 k 层三角格网的邻近搜索。Bartholdi 的邻近搜索算法用的是层次细化剖分的方法, 将其剖分一次及相关的计算称为 Bartholdi 搜索的单次操作。同样, 分解邻近搜索算法中的“ $>$ ”运算记为分解法的单次操作。从单次操作来说, Bartholdi 算法涉及浮点数乘法、开方等运算, 计算复杂; 分解搜索算法是四进制的整数变换, 可将其转化为二进制位运算, 计算简单。 k 层编码邻近搜索中, Bartholdi 需要进行 k 次单操作; 而分解搜索算法每次处理 2 位编码, 最多需进行 $k/2$ 次单操作。

通过“ $>$ ”运算出现的概率来计算分解算法的平均时间复杂度。假设以“ a ”结尾的编码出现的概率是 b , 编码变换中, “ $>$ ”运算出现的概率是 c , 则以“ a ”结尾且出现“ $>$ ”运算的概率是 $b * c$ 。按同样的方法计算在各种情况下出现“ $>$ ”运算的概率总和。在分解算法中, “ $>$ ”运算出现的概率是 5/32, 有 53/64 的情况适用分解邻近搜索算法, 则改进后算法的平均时间复杂度为 $(53/64) * (5/32) * k/2 + k * 11/64$, 即平均所需的时间是 Bartholdi 算法的 23.66%。

3.2 实验结果

笔者通过实验对 Bartholdi 算法和改进后的分解算法效率作了更深入的分析。先找出 6~13 层所有的 Quaternary 编码, 然后分别计算出用 Bartholdi 算法和改进后的算法对所有格网编码进行邻近搜索所需的时间总量; 再用同样的方法分别计算出 14~25 层部分编码进行邻近搜索所需的时间(如表 3 所示)。1~5 层的格网数少, 且每个格网的编码长度小, 对所有格网进行邻近搜索所需的时间少, 故在实验中不予考虑。另一方面, 14 层剖分产生的格网数高达 268 435 456 个, 若对所有格网进行邻近搜索, 耗时量将会超过 6 h, 而且剖分层次越细, 进行全部格网的邻近搜索所需的时间越长, 因而 14 层以上只计算部分格网邻近搜索的耗时量。从表 3 可以看出, 随着剖分层次的增加, 邻近搜索的效率得到了显著的提高, 改进后算法的耗时量和 Bartholdi 算法的比值由 34.97% 收敛到 25.42%。实验与理论分析的结果基本一致, 分解算法邻近搜索的效率是 Bartholdi 算法的 4.2 倍。

表3 邻近搜索的效率对照表

Tab.3 Ratio of Neighbor-Finding with Bartholdi's and Decomposing Method

层次 N	邻近搜索的次数	改进后算法耗时量/ms	Bar 算法耗时量/ms	改进后算法 /Bar 算法/%
6	$4^N = 4\ 096$	179	512	34.97
7	$4^N = 16\ 384$	806	2 418	33.34
8	$4^N = 65\ 536$	3 564	11 003	32.39
9	$4^N = 262\ 144$	15 938	49 578	32.15
10	$4^N = 1\ 048\ 576$	67 874	221 389	30.66
11	$4^N = 4\ 194\ 304$	290 761	975 923	29.79
12	$4^N = 16\ 777\ 216$	1 221 196	4 196 468	29.10
13	$4^N = 67\ 108\ 864$	5 092 387	17 834 989	28.55
14	$4^{N-1} = 67\ 108\ 864$	5 665 024	20 085 120	28.21
15	$4^{N-2} = 67\ 108\ 864$	5 972 032	21 551 040	27.71
16	$4^{N-3} = 67\ 108\ 864$	6 364 160	23 308 992	27.30
17	$4^{N-4} = 67\ 108\ 864$	6 601 024	24 339 968	27.12
18	$4^{N-5} = 67\ 108\ 864$	6 913 024	26 049 088	26.54
19	$4^{N-6} = 67\ 108\ 864$	7 327 040	27 674 880	26.48
20	$4^{N-7} = 67\ 108\ 864$	7 595 008	29 131 072	26.07
21	$4^{N-8} = 67\ 108\ 864$	7 867 968	30 360 960	25.91
22	$4^{N-9} = 67\ 108\ 864$	8 188 992	32 061 056	25.54
23	$4^{N-10} = 67\ 108\ 864$	8 512 000	33 372 992	25.51
24	$4^{N-11} = 67\ 108\ 864$	8 924 992	35 052 032	25.46
25	$4^{N-12} = 67\ 108\ 864$	9 332 190	36 712 000	25.42

参 考 文 献

- [1] Bartholdi J, Goldsman P. Continuous Indexing of Hierarchical Subdivisions of the Globe[J]. International Journal of Geographical Information Science, 2001, 15(6): 489-522
- [2] Nulty W G. Geometric Searching with Spacefill-ing Curves [D]. Atlanta, GA: Georgia Institute of Technology, 1993

- [3] Lee M, Samet H. Navigating Through Triangle Meshes Implemented as Linear Quadtrees[J]. ACM Transactions on Graphics, 2000, 19(2): 79-121
- [4] Chen Jun, Zhao Xuesheng. An Algorithm for the Generation of Voronoi Diagrams on the Sphere Based on QTM[J]. Photogrammetric Engineering and Remote Sensing, 2003, 69(1): 79-89
- [5] Dutton G. Locational Properties of Quaternary Triangular Meshes[C]. The 4th International Symposium on Spatial Data Handling, Zurich, Switzerland, 1990
- [6] Dutton G. Encoding and Handing Geospatial Data with Hierarchical Triangular Meshes[C]. The 7th International Symposium on Spatial Data Handling, Netherlands, 1996
- [7] Goodchild M F, Yang Shiren. A Hierarchical Data Structure for Global Geographic Information Systems[J]. Computer Vision and Geographic Image Processing, 1992, 54(1): 31-44
- [8] Fekete G. Rendering and Managing Spherical Data with Sphere Quadtree[C]. Visualization'90, San-Francisco, CA, 1990
- [9] Otoo E J, Zhu Hongwen. Indexing on Spherical Surfaces Using Semi-quadtrees[C]. Advances in Spatial Database-Third Annual Symposium, SSD'93, Singapore, 1993

第一作者简介: 孙文彬, 博士生。主要研究方向为空间数据模型、GIS的应用等。

E-mail: swb1996@sina.com

Algorithm of Neighbor Finding on Sphere Triangular Meshes with Quaternary Code

SUN Wenbin¹ ZHAO Xuesheng^{1,2}

(1) School of Resource and Safety Engineering, China University of Mining and Technology (Beijing),

D11 Xueyuan Road, Beijing 100083, China)

(2) National Geomatics Center of China, 1 Baishengcun, Zizhuyuan, Beijing 100044, China)

Abstract: The characteristic of Quaternary codes is analyzed. The rule of distinguishing triangle direction is given out. The algorithm of neighbor finding from back to front with decomposing Quaternary code is presented. The analysis of time complexity between this decomposing algorithm and bartholdi's algorithm is approached. The result is that average consumed time of this decomposing algorithm is about 23.66% of Bartholdi's.

Key words: Quaternary code; neighbor finding; sphere triangular meshes