

基于DEM数据库的可视域算法^①

晁玉忠, 张春明, 张磊

(山东省农村信用社联合社, 济南 250001)

摘要: 可视性分析在军事和工程领域有着广泛的应用, 可视域计算是可视性分析的重要因子. 首先讨论了可视域计算的常用算法, 分析了各个常用算法的优劣. 并针对大数据量的DEM数据库, 在Xdraw算法的基础上提出了一种改进的Xdraw算法, 大大减少了访问数据库的次数. 然后在实验的基础上从正确性和效率两个方面对该算法进行了分析, 并给出了继续研究的方向.

关键词: 可视性; 可视域; 数字高程模型

Viewshed Algorithm Based on DEM Database

CHAO Yu-Zhong, ZHANG Chun-Ming, ZHANG Lei

(Shandong Rural Credit Cooperative Union, Jinan 250001, China)

Abstract: Visibility analysis is widely applied to military and project fields, of which viewshed computing is the important factor. This article discusses the common algorithms for viewshed computing, analyses the advantages and disadvantages of them, and puts forward a improved Xdraw algorithm based on the old one to reduce obviously the frequency of access to database aiming at a large scale DEM database. Then the continuous research direction is shown through analysis of this algorithm from correctness and efficiency in practice.

Key words: visibility; viewshed; digital elevation model

1 引言

可视性分析也称通视分析, 它实质属于对地形进行最优化处理的范畴, 是地形本身固有的属性. 可视性分析历来是军事作战的重要情报来源^[1], 通过分析来选择有利地形达到有利于隐藏自己和攻击敌人的目的. 在电视台的发射站、道路选择、旅游等领域也有重要的应用^[2-3]. 可视性分析的基本因子有两个, 一个是两点之间的可视性(intervisibility), 另一个是可视域(viewshed), 即对于给定的观察点所覆盖的区域.

可视性分析是在一定的地形表达基础上进行的. 从数学的角度^[4], 可以用下述二维函数系列取值的有序集合来概括地表示数字地面模型(Digital Terrain Model)的丰富内容和多样形式:

$$K_p = f_k(u_p, v_p) \quad (k = 1, 2, 3, \dots, m; p = 1, 2, 3, \dots, n)$$

式中, K_p 为第 p 号地面点上的第 k 类地面特性信息的取值; (u_p, v_p) 为第 p 号地面点的二维坐标; m 为地面特性信息类型的数目; n 为地面点的个数; 当 k 为高度信息时为数字高程模型(Digital Elevation Model)即 DEM. DEM 有两种形式, TIN(Triangulated Irregular Networks)和 RSG(Regular Square Grids), 本文讨论基于 RSG 进行. 另外, 当数据量较大时, 通常按照一定的数据库结构来管理数据, 即 DEM 数据库.

2 计算可视性的算法

2.1 两点之间的可视性判断

两点间的可视性通常是基于视线判断的. 如果地形上两点 p_1 、 p_2 能够交互可视, 当且仅当 $q = (x, y, z) = p_1 + t(p_2 - p_1), 0 < t < 1$ 高于对应的地形点,

^① 收稿时间:2011-12-16;收到修改稿时间:2012-01-20

其中 p_1 称为视点 O , p_2 称为目标点 T , 则过视点 T 的射线 OT 称之为视线. 在视线的基础上, 主要有 Janus、Dynatacs、Modsaf、Bresenham 等算法. 两点之间可视性判断的所有算法基本相同, 不同之处在于高程内插 (DEM Interpolation) 计算方面和可视判断原则方面^[5].

2.2 可视域计算

可视域计算一般是基于视线来进行的^[6-8]. Shapira^[6]提出跟踪从视点 V 到任何其他点 P 的视线, 并沿着 V 与 P 之间的视线移动的方法, 在到达 P 之前遇到地形边与视线之间的交或到达 P 时, 就终止移动. 可视域计算是一项非常耗时的操作, 当数据量很大时尤其如此, 另一方面 RSG 较适合并行处理, 为此 Mills 等^[9]使用 Shapira 算法的并行版本, 对源区域的每个视点的视线进行了并行处理, Teng 等^[10,11]又在此基础上进行了改进. 目前, 根据地形剖面的选取和插值方法的不同, 常用的可视域计算方法有以下几种.

(1) R3 算法^[6-8]. 对于视域范围内的每个目标点, 建立从观察点到目标点的地形剖面, 地形剖面的取样点由剖面线与网格线的交点决定, 每个高程点的高程值通过相邻两格点上的高程值插值而得. 该算法是一种精确算法, 计算时无需考虑计算顺序. 在视域半径为 n 的情况下, 用该算法求可视域的时间复杂度为 $O(n^3)$.

(2) R2 算法^[8]. 是在 R3 基础上改进的一种近似算法. 因为在计算边界点的可视性时, 剖面上各取样点(与格网线的交叉点)的可视性已计算出, 将与剖面上取样点相邻的格点的可视性设置为该取样点的可视性, 并记下取样点与格点的距离, 当有新的与该格点更近的取样点时, 用此取样点的可视性更新该格点的可视性. 因此只需依次计算边界点的可视性, 内部格点的可视性采用最近点可视性逼近. 在视域半径为 n 的情况下, 该算法的时间复杂度为 $O(n^2)$, 另外需要 $O(n^2)$ 的额外空间开销来记录取样点与格点的最近距离.

(3) Xdraw 算法^[8]. 沿方形环自里而外判断, 每一点记录有该点上空最低可视视线. 当要计算第 i 环上的格点的可视性时, 可以利用 $i-1$ 环的相邻格点上记录的最低可视视线来判断该点的可视性, 高于内环视线则为可见, 否则, 该点不可见. 若视域半径为 n , 该算法时间复杂性为 $O(n^2)$, 另外需要 $O(n)$ 的辅助空间来存储最低可视视线等信息.

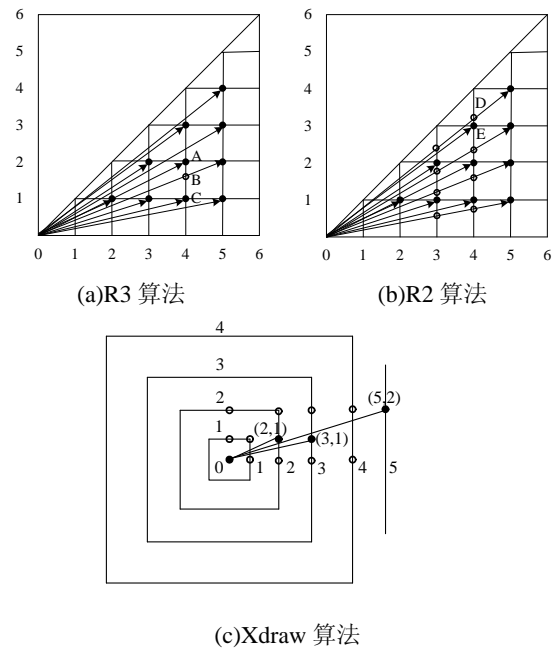


图 1 可视域分析的常用算法

2.3 基于 DEM 数据库的可视域算法

DEM 数据库中的数据一般数据量大, 不能一次性全部调入内存来进行分析, 一次只能调入其中的一小块. 因此对 DEM 数据库进行可视域分析来说, 如何减少访问数据库的次数是至关重要的, 另外, 由于数据量较大算法的时间复杂度也不能太高. 在此基础上, 对上述 3 种常用的可视域算法进行比较. R3 算法的取样点由剖面线与网格线的交点决定, 要判断某个点的可视性需要依次调入视线所经过的数据块, 另一方面 R3 算法复杂度过高, 不适合 DEM 数据库的可视域分析. R2 算法先计算边界点的可视性, 内部格点的可视性用最近点逼近, 较之 R3 算法大大减少了访问数据库的次数, 但是仍然存在一个数据块多次调入的情况, 并且 R2 算法是一种近似算法其精度不是很高. Xdraw 算法在较低复杂度的情况下具有较好的精度, 对 DEM 数据库的可视域分析是个很好的选择, 但如果直接依据 Xdraw 算法从内到外进行扩散分析, 仍然会有同一个数据块多次调用的情况, 下面在不改变 Xdraw 算法时间复杂度和计算精度的情况下, 采取一种调度策略对 Xdraw 算法进行改进, 使每个数据块被调用一次.

3 一种改进的 Xdraw 算法

3.1 算法的总体设计思路

基于 DEM 数据库, 该算法按照指定的视点和范

围给出该视点在这个范围内的可视域. 可视域的计算范围通常用多边形表示, 本文用多边形的外接矩形给出. 事实上, 判断矩形内每个格网点的可视性, 输出时只输出在多边形内的点即可. 该算法是为了改进 Xdraw 算法因多次冗余地访问数据库而造成效率低下而提出的, 它在点的可视性可以递推的基础上, 利用分块扩散的思想设计而成.

算法的总体思想如下:

(1) 在点之间的可视性可以递推的基础上利用扩散的思想, 如下图 2 所示, 在格网中有一视点 V, 包含在分块 ABCD 内, 则 ABCD 内的点可以容易的判断出对视点 V 是否可视. 利用该分块的边界分别往八个方向扩散, 得到第一次扩散后的矩形 EFGH. 根据点之间的可视性的递推关系, 利用 ABCD 的四个边界, 可以判断出新扩散出的八个分块内的点与视点 V 是否可视.

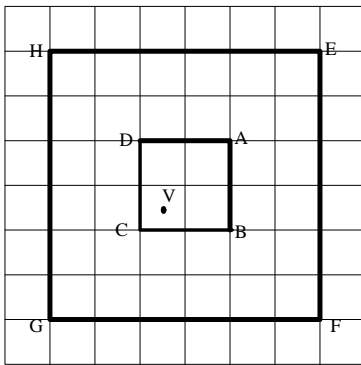


图 2 分块的扩散

(2) 重复上述操作, 利用上一次扩散得到的边界上点的可视信息, 可以判断出下一次扩散的各个分块内的点对 V 点是否可视.

(3) 当扩散后所得矩形的某个边到达给定范围的外接矩形的一条边时, 则这条边停止扩散. 当矩形的 4 条边都不能再扩散时, 停止.

据上所述, 本算法分四个主要的步骤讨论:

(1) 每一次扩散后各个分块之间的判断顺序.

(2) 每一次扩散后得到的矩形的边界信息是下一次扩散的基础, 边界上格网点的信息存储问题.

(3) 对每个分块内的点, 利用 Xdraw 算法判断各个点的可视性.

(4) 算法的终止条件.

3.2 扩散后分块的判断顺序

如下图 3 所示, V 为视点, 红色的线为每次扩散后

的边界, 最里层的为包含 V 的原始分块. 以 V 为中心, 每次扩散后的分块按照垂直、对角线方向分为 8 个部分, 编号依次为 0、1……8, 蓝色的线表示分块间的边界. 其中 0、1、2、3 每个部分可能含有多个分块, 4、5、6、7 每个部分含有一个分块. 现考虑第 4 部分内的两个点 A、C, VA、VC 连线分别于蓝色边界交于 B、D 点. 根据点的可视性的递推可以看出, 要判断 A、C 点的可视性, 必须已知 B、D 两点的可视性. 因此对第 4 部分来说, 要后于 0、1 部分判断.

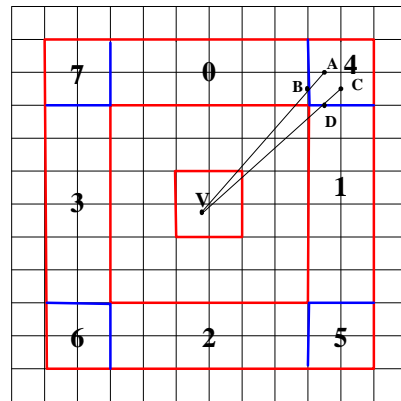


图 3 扩散后分块的判断顺序

据此可以得出, 要先判断 0、1、2、3 部分的分块, 再判断 4、5、6、7 部分的分块. 用类似的方法还可以得出, 对 0、1、2、3 每个部分含有多个分块(对第 i 次扩散, 含有 2*i-1 个分块)的情况, 应先判断在 V 垂直方向的分块, 再依次判断两边的分块.

3.3 边界信息的存储

由上图 3 可知, 算法中的边界分为两种, 一种为扩散的边界, 它是下次扩散的基础; 另一种是每次扩散后的内部边界. 对于这两种边界, 一个分块把它存储起来, 另一个以此边界为判断基础的分块应能够快速查找取出边界上格网点的信息. 为方便查找, 用红黑树来存储边界点的可视信息.

```
typedef struct //红黑树 key 的类型
{
    double row; //行坐标
    double col; //列坐标
} RBTreeKey;
```

3.4 分块内可视信息的判断

对于包含视点的原始分块, 用 R3 算法判断分块内格网点的可视性. 这样做的好处在于, 可以保证原始

扩散边界上格网点可视信息的精确性, 为扩散提供精确的原始可视信息.

对于由扩散形成的分块, 用 Xdraw 算法来判定分块内格网点的可视性. 以上图 3 中第 5 部分的分块为例来说明具体的判断方法. 如下图 4, 由扩散后分块的判断顺序可知, 此时该分块的上边界和左边界的可视信息已知. 对分块内的点, 按行从上到下、列从左至右的顺序来判断每个格网点的可视性. 现考虑分块内的格网点 C, C、V 的连线与 AB 交于 D, 由格网点的判断顺序此时 A、B 的可视信息, D 的可视信息可由 A、B 通过线性插值得到, 然后由 D 点的可视信息推出 C 的可视性. 利用相同方法能得到块内所有点的可视性.

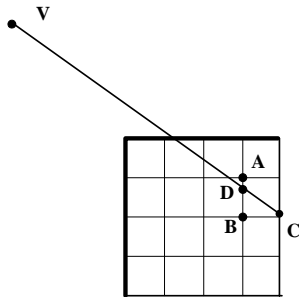


图 4 分块内可视信息的判断

3.5 算法的终止条件

对可视域计算范围, 其外接矩形的每个边都被扩散到时, 算法停止. 需要说明的是, 对外接矩形的四个边, 并不一定同时被扩散到, 在这样的情况下, 沿未被扩散到的边的方向继续扩散、其他的方向不动就可以了, 直到所有的边都被扩散到. 如下图 5, 图中的上、下两个方向首先扩散到外接矩形的边界, 此时上、下两个方向停止扩散, 左、右两个方向继续扩散.

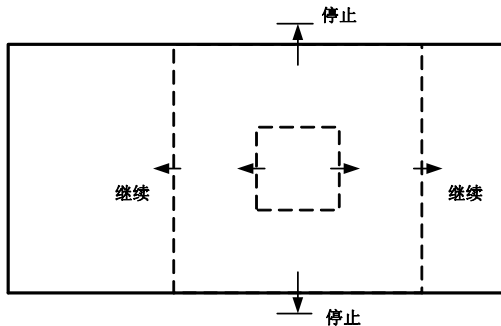


图 5 算法的终止

3.6 算法综述

根据上面的讨论, 算法的描述如下:

输入: 多边形 P(可视域的计算范围), 视点 V, 分块大小 N.

输出: P 内每个格网点的可视性.

step1:求 P 的外接矩形 R. 若 $R < N$, 用 R3 算法判断 P 内格网点的可视性, goto step5.

step2:用 R3 算法判断视点 V 所在原始分块内格网点的可视性并输出, 同时保存边界信息.

step3:扩散, 判断分块内每个格网点的可视性, 当在多边形内时输出这个点的可视性, 保存相应的边界信息.

step4:若 R 有未被扩散到的边, goto step3.

step5:终止.

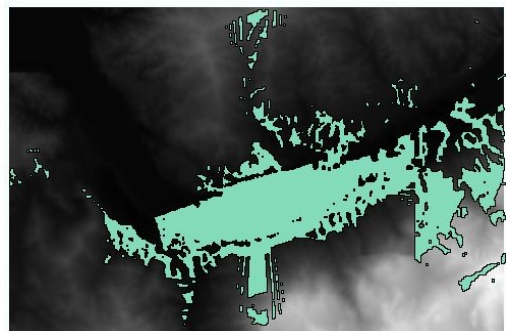
经分析可知, 若视域半径为 n, 算法的时间复杂度为 $o(n^2)$, 另外需要 $o(n)$ 的额外空间存放边界信息. 可以看出, 该算法与 Xdraw 算法具有相同的时间复杂度和空间复杂度. 此算法是在 Xdraw 算法的基础上改进的, 称之为一种改进的 Xdraw 算法.

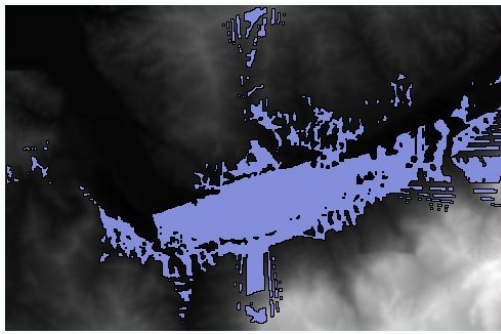
4 实验结果与比较

4.1 实验结果

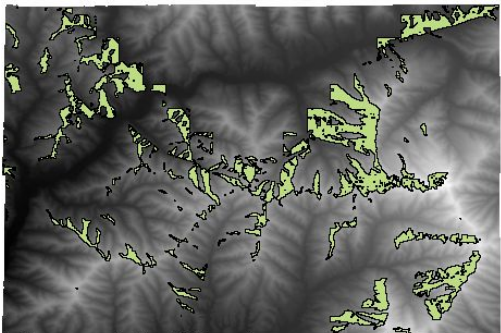
本实验的测试环境为: 赛扬 1.7GHz CPU, 256 内存, Windows XP 操作系统, Visual Studio 2005 平台, DEM 数据库类型为 Oracle, 存放在通过网线直接相连的另一台机器上.

采用三组不同规模的数据进行测试, 格网数分别为 750×488 , 2694×1848 , 2705×3589 . 视点位置为 DEM 矩形范围的中心. 用这种改进的 Xdraw 算法与 R3 算法进行比较, 三组实验结果如下图, 其中左边的是 R3 算法的结果, 右边的是改进的 Xdraw 算法的结果. 表一给出了实验的性能比较.

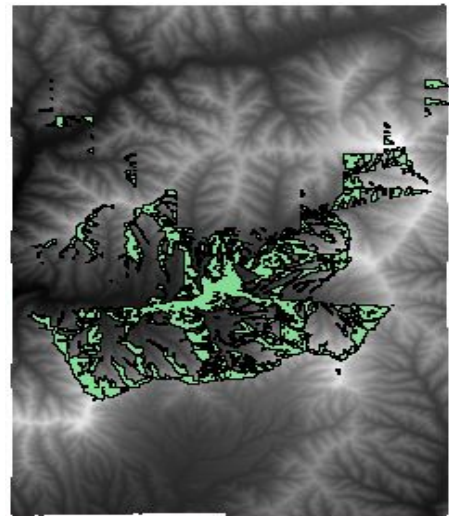
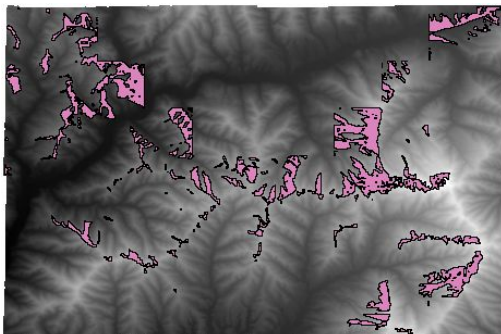




(a) 实验一



(b) 实验二



(c) 实验三

图 6 实验结果

表 1 改进的 Xdraw 算法与 R3 算法的比较

组别 \ 性能	实验一	实验二	实验三
格网数	750*488	2694*1848	2705*3589
R3 算法的时间消耗 (单位: 秒)	1247.028	5548.610	17952.919
改进 Xdraw 算法的时间消耗 (单位: 秒)	25.266	87.719	180.675
误差比例	7.476%	10.091%	11.378%

4.2 准确性比较

从表 1 可以看出, 改进的 Xdraw 算法与 R3 算法存在着一定的偏差, 比例在 10%左右, 这在工程领域是完全可以接受的. 造成的误差一方面是可视信息在从内向外传递的过程中造成的, 外层格网点的可视性通过内层格网点的可视信息而不是原始视点计算而来, 这会形成一定的误差. 另一方面, 分析的坐标系是基于笛卡儿平面直角坐标的, 而地球表面是一个弧形球面, 当分析范围较大时, 要对高程信息进行预校正.

4.3 效率比较

从表 1 可以看到, 当范围较大时, 用 R3 算法进行可视域计算是非常耗时的工作, 在实际应用中并不可行. 本文提出的改进的 Xdraw 算法较之 R3 算法有显著的提高, R3 算法与改进的 Xdraw 算法消耗的时间比如

下表 2 所示. 从表 2 可以看到, 在实验一格网数为 750*488 时, R3 算法与改进的 Xdraw 算法消耗的时间比约为 50 倍, 在实验三格网数为 2705*3589 时约为 100 倍. 可以预见, 当格网数继续增大时, 倍数会随之增大. 原因主要有两个方面, 首先, 改进的 Xdraw 算法与 R3 算法时间复杂度不同, 分别为 $o(n^2)$ 和 $o(n^3)$; 其次, 随着格网数的增大, 分析范围随之扩大, R3 算法访问数据库的次数会不断增大, 其增大的幅度远远大于改进的 Xdraw 算法增大的幅度.

表 2 R3 算法与改进的 Xdraw 算法消耗的时间比

组别 性能	实验一	实验二	实验三
R3 算法与改进的 Xdraw 算法消耗的时间比 (单位: 倍)	49.356	63.254	99.366

5 总结

可视域计算在工程中有着广泛的应用, 如何提高可视域计算的效率尤其重要. 本文在仔细研究已有可视域算法的基础上, 针对大数据量的 DEM 数据库, 提出了一种改进的 Xdraw 算法. 这种改进的 Xdraw 算法在保证计算准确性的前提下, 大大提高了计算的效率, 取得了良好的效果. 在该算法中, 分块的大小对计算准确性和效率的影响需要进一步讨论. 另外, 当范围较大时, 由于地球表面是一个弧形球面, 可视域的计算一般是基于笛卡尔平面直角坐标的, 要对高程进行预校正, 可以采用如下近似公式进行校正^[12]:

$$E = E' - (D/2R)^2$$

式中, E 为原始高程, E' 为校正后的高程, D 为试点到目标点的距离, R 是地球半径, 校正误差为 $(D/2R)^2$.

参考文献

- Zhang JF, Li L, Wang YX. Analysis of terrain visibility. Journal of System Simulation, 2005(17):1916-1921.
- Huang PZ. Application of DEM in radio site location. Journal of Southwest Jiaotong University, 1990(4):70-78.
- Deflorlanil M. Intervisibility on Terrains. Geographic Information Systems: Principles, Techniques, Management and Applications. New York: John Wiley & Sons, 1999, 543-556.
- Li ZL, Zhu Q. Digital Elevation Model. WuHan: WuHan University Press, 2001.
- Shen Y. Key Techniques and Applications of Spatial Visibility Analysis. PHD Dissertation, Wuhan University. 2005.
- Shapira A. Visibility and Terrain Labeling. Master Thesis. Troy, NY, Rensselaer Polytechnic Institute, 1990.
- Franklin WR, Ray CK, Shashank Mehta. Geometric Algorithms for Siting of Air Defense Missile Batteries. Research Project for Battle. Columbus Division, Contract Number DAAL03-86-D-0001, Delivery Order Number 2756, 1994.
- Franklin WR, Ray CK. Higher Isn't Necessarily Better: Visibility Algorithms and Experiments. 6th International Symposium on Spatial Data Handling, Sept., Advances in GIS Research, 1994(1):751-770.
- Mills K, Fox G, Heimbach. Implementing an Intervisibility Analysis Model on a Parallel Computing System. Computer & Geosciences, 1992(18):1047-1054.
- Teng YA, De Menthon D, Davis LS. Region-To-Region Visibility Analysis Using Data Parallel Machines. Concurrency: Practice & Experience 5, 1993, 379-406.
- De Floriani L, Magillo P. Visibility Algorithms on Triangulated Digital Terrain Models. International Journal of Geographical Information Systems, 1994(8):13-42.
- Franklin WR. Siting Observers on Terrain. <http://www.ecse.rpi.edu/Homepages/wrf>, 2002.