

用于地面可视化的 DEM 四叉树改进

郝鹏威 黄波 朱重光

(中国科学院遥感应用研究所, 北京 100101)

摘要 四叉树表示在数字高程模型(DEM)的压缩存储和地面可视化中占有非常重要的地位。此文吸收了 DEM 普通四叉树表示和塔形分层表示的优点,提出了一种改进的四叉树结构。它主要是在结点的数据域中增加了高程最大值和最小值项,并把四叉树线索化了。经过改进的四叉树结构使 DEM 在地面可视化中进行的光线追踪时,光线追踪效率得到了提高。

关键词 数字高程模型(DEM),四叉树表示,线索树,地面可视化

1 引言

数字高程模型(DEM)的数据结构大致分为:格网数据结构、矢量数据结构和曲面数据结构^[1]。本文主要涉及格网数据结构,或称栅格结构。

栅格 DEM 的最大缺点是占用大量存储空间,为此,人们提出许多压缩存储的方案,如行程编码、四叉树结构编码等。其中四叉树结构编码因为其空间结构与 DEM 一致,查找、存取方便而被广为应用。其次,在地面可视化中,为使可视化过程中光线追踪效率更高,栅格 DEM 的数据结构又有八叉树表示^[2,3]、四叉树表示^[4]、塔形分层表示^[5]等多种形式。

八叉树表示是把 DEM 看成是一个三维的实体,由于其中大多数网格是空的,光线追踪就可以跨过大量“透明”空间,迅速到达地形表面。八叉树结构较适用于实体的三维场景表示。对于仅仅是一张曲面的 DEM,其厚度可以认为是零,用八叉树表示就会很不精确、结点数目激增或还需增加新的开销(如建立用以求交的曲面片表)。

四叉树表示与 DEM 压缩存储时所用四叉树一致,在地面可视化中,通过跨过不与光线相交的较大的四叉树网格而达到快速求交的目的。一个 $2^n \times 2^n$ 的 DEM 用四叉树表示时,其树高最大为 n 。没有线

索化的四叉树对邻接结点的查找较为复杂,这将直接影响到地面可视化中光线追踪的效率。

塔形分层表示把大小为 $2^n \times 2^n$ 的 DEM 分成 n 层,上一层是下一层的 $1/4$,上一层的一个网格的区域代表下一层 4 个网格的区域,值则只保存这 4 个网格高程值中之最大值。塔形分层的结构与满四叉树等同。对塔形分层表示的 DEM,在可视化过程中最多只需进行 n 次求交运算就可以得到被追踪光线与地面之交点。

本文提出一种改进的四叉树结构,这种结构既有着一般四叉树的结构化和可线索化的特点,又吸收了塔形分层表示的优点,使得地面可视化过程更加快捷。

2 DEM 四叉树的改进

2.1 DEM 一般四叉树的结构及其缺点

DEM 存储的四叉树构造形式一般为:

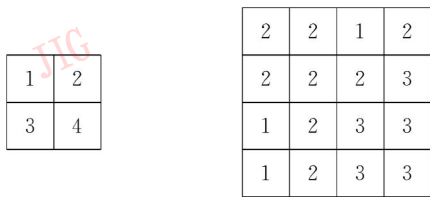
(1)根结点表示整个 DEM,结点内容包括 DEM 的大小和 4 个子结点指针。

(2)整个 DEM 之高程不完全一样则平分为 4 个子块,每一块由一个子结点表示。每一子结点的内容包括 DEM 子块的方位及 4 个子块再细分的子结点指针。

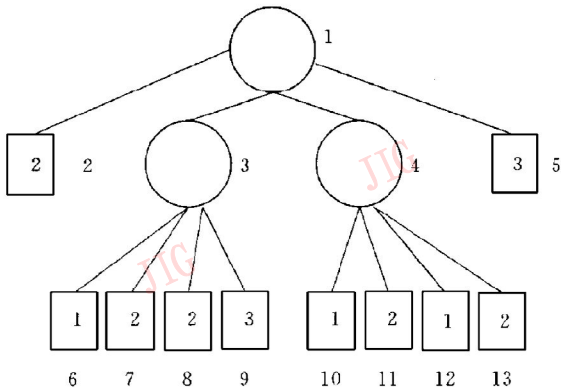
(3)如子结点 DEM 中所有栅格点高程完全相等,则子结点便是叶结点,其内容包括 DEM 子块方位及高程值,其 4 指针赋空。如子结点 DEM 之高程不完全相等则再平分为 4 个子块,每一块再由一个子结点表示。重复这一过程,直到子结点再细分为止。

如果无需快速获得任一结点的定位信息,则在上述子结点内容中可不包括方位信息。

二叉树结点的指针顺序编号为图 1(a)。对于图 1(b)表示的 DEM,其所对应的 DEM 一般二叉树表示为图 1(c),其中结点外的数字为树结点编号。其相应的表格表示为表 1。



(a) 子结点指针顺序编号 (b)原始栅格 DEM
(a)Numbered quadtree node pointers
(b)Original grid DEM



(c)一般二叉树表示
(c)Ordinary quadtree

图 1 DEM 的一般二叉树表示
Fig. 1 Ordinary quadtree of DEM

从图 1 和表 1 中可以看出,DEM 的一般二叉树表示存在两方面的缺点:一是非叶结点的高程值项没有得到利用,二是所有叶结点的指针域全赋空而没有得到利用。因此,对 DEM 二叉树进行改进可从两个方面进行:一是对二叉树结点内容的改进。对每一结点增加数据域,使之同时具有其 4 个子结点的高程最大值和最小值。二是把二叉树线索化,使之很容易就能找到邻接结点。

表 1 图 1(b)DEM 一般二叉树的表格表示

Table 1 Table for ordinary quadtree of DEM in Fig. 1(b)

序号 (Index)	高程值 (Z)	子结点指针 Child1	子结点指针 Child2	子结点指针 Child3	子结点指针 Child4
1	—	2	3	4	5
2	2	0	0	0	0
3	—	6	7	8	9
4	—	10	11	12	13
5	3	0	0	0	0
6	1	0	0	0	0
7	2	0	0	0	0
8	2	0	0	0	0
9	3	0	0	0	0
10	1	0	0	0	0
11	2	0	0	0	0
12	1	0	0	0	0
13	2	0	0	0	0

2.2 二叉树结点内容的改进

DEM 塔形分层表示的优点之一是保存了 4 个子结点高程值中之最大值,这使得高于此值的高程全部处于光线追踪时用于测试的包围盒外。这一优点可以被吸收进来,并增加 4 个子结点高程值中之最小值项,这样可以进一步缩小包围盒。

因此,二叉树结点内容可改进为:

- (1) 使所有结点(包括根结点和叶结点)内容一致。
- (2) 所有结点设子结点高程值中之高程最大值和最小值两数据域。叶结点的高程最大值和最小值两数据相等。

(3) 为使每一结点的定位容易,子结点内容中应加入结点方位信息和结点大小信息,或结点代表的区域范围。对本文改进二叉树结点的定位主要是邻接结点的定位,因此,每一结点只要有所在层数信息即可根据本结点定位信息简单计算出其邻接结点的定位信息。

如果叶结点的高程最大值和最小值可以不相等,其差允许小于等于某个宽限的值,这样可以形成带有一定容限的 DEM 改进二叉树。很显然,这会使改进的二叉树深度减小、结点数大大减少。此时,必须用一个标志位(tag)来标识叶结点。

也正是由于增加了结点内容,因此,改进的二叉树不适合于起伏频繁的 DEM 压缩。

2.3 二叉树的线索化

在地面可视化过程中,光线要顺邻接结点进行

追踪,因此,邻接结点的查找效率对于光线追踪效率自然十分重要。由图 1 和表 1 可见,其中四叉树的所有叶结点的子结点指针均赋空了,如果把这些空指针利用起来则还可以实现邻接结点的线索化,这对于地面可视化是十分有益的。线索化的途径如下:

(1) 每一叶结点的 4 个子结点指针指向 4 个邻接结点。其中 2 个是属于同一父结点的子结点(即兄弟),另 2 个结点则是不低于本叶结点层的结点(不一定是叶结点)。

(2) 叶结点处于 DEM 边缘,其某个方向的邻接结点不存在时,对应指针赋空。

(3) 判断结点是否为叶结点、指针是否为线索的方法是判断该结点之高程最大最小值是否相等,或判断标志位的值是否为 1。

把叶结点的 4 个子结点指针作为线索化指针时可按图 2 编号。这样,图 1 和表 1 表示的四叉树改进后的结果如表 2 所示。

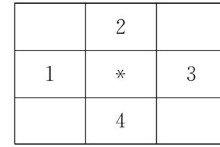


图 2 叶结点(*)的 4 个邻接结点编号

Fig. 2 Numbered adjacent nodes of quadtree leaf (*)

表 2 图 1(b)DEM 改进四叉树的表格表示

Table 2 Table for improved quadtree of DEM in Fig. 1(b)

序号 (Index)	最大高程 (Zmax)	最小高程 (Zmin)	结点层号 (Layer)	子结点指针 Child1	子结点指针 Child2	子结点指针 Child3	子结点指针 Child4
1	3	1	0	2	3	4	5
2	2	2	1	0	0	3	4
3	3	1	1	6	7	8	9
4	2	1	1	10	11	12	13
5	3	3	1	4	3	0	0
6	1	1	2	2	0	7	8
7	2	2	2	6	0	0	9
8	2	2	2	2	6	9	5
9	3	3	2	8	7	0	5
10	1	1	2	0	2	11	12
11	2	2	2	10	2	5	13
12	1	1	2	0	10	13	0
13	2	2	2	12	11	5	0

3 改进的 DEM 四叉树的操作

经改进的四叉树的建立、结点查找、邻接结点查找和定位等都与一般四叉树有所区别。线索的建立和利用是改进的四叉树操作的关键。

3.1 四叉树建立

DEM 四叉树的建立方法主要有两种:一种是由上而下大块分裂法,即如果 DEM 块内高程值不完全相等,则 4 等分为 4 个子结点,直到子块内高程值都相等。另一种是由下而上小块合并法,即把

DEM 小块与邻接的另三小块 DEM 内之高程值比较,如果完全一致,则合并为一大块。如不能合并,则建一新父结点,并把这 4 个结点划归其作子结点,直到父结点表示的区域为整个 DEM。

以下是用分裂法建立 DEM 四叉树的算法:

```

PROCEDURE BuildQuadtree( node, i, j, k );
{建立第 k 层坐标在(i, j)与(i+2n-k-1, j+2n-k-1)
之间的 DEM 四叉树,并返回给 node}
{调用方法:BuildQuadtree( root, 0, 0, 0 )}
BEGIN
node := newnode(); {得到新结点}
GetExtrema( Zmax, Zmin, i, j, k );
{得坐标在(i, j)与(i+2n-k-1, j+2n-k-1)

```

之间的 DEM 高程最大最小值,并赋给 Z_{max} 和 Z_{min}

```

node^.Zmax := Zmax;
node^.Zmin := Zmin;
IF ( Zmin=Zmax ) THEN
  FOR iChild:= 1 TO 4 DO node^.Child
[iChild]:= NIL {叶结点}
ELSE
  FOR iChild := 0 TO 3 DO {非叶结点继续建立各
子树}
    BuildQuadtree( node^.Child[iChild +1],
      i+(iChild MOD 2) * 2n-k, j+
(iChild DIV 2) * 2n-k, k+1);
END;

```

3.2 叶结点的判别

叶结点的高程值唯一,其最大值与最小值必然相等。因此,用来判别是否为叶结点的 Boolean 函数可定义为:

$IsLeaf(node) := (node^.Zmax = node^.Zmin)$ 。

对于带有一定容限的 DEM 改进二叉树,叶结点的高程值不唯一,是否为叶结点的判别必须通过设置的标志位(tag)来判别。

3.3 结点的查找

知道 DEM 中某个点的坐标(i, j),需要确定包含该点且层数不超过 k 的结点时,就要用到结点的查找。

若根结点(第 0 层)代表 DEM 的坐标范围为 $(0, 0)$ 与 $(2^n - 1, 2^n - 1)$ 之间,则其 4 个子结点(第 1 层)的对角坐标分别为 $(0, 0)$, $(2^{n-1} - 1, 2^{n-1} - 1)$; $(2^{n-1}, 0)$, $(2^{n-1}, 2^{n-1} - 1)$; $(0, 2^{n-1})$, $(2^{n-1} - 1, 2^n - 1)$; $(2^{n-1}, 2^{n-1})$, $(2^{n-1}, 2^n - 1)$ 。

对第 m 层中区域 $(p * 2^{n-m}, q * 2^{n-m})$, $(p+1) * 2^{n-m} - 1, (q+1) * 2^{n-m} - 1$, 块大小为 $2^{n-m} \times 2^{n-m}$, 最多有 $2^m \times 2^m$ 块。 p, q 取值范围为 0 到 $2^m - 1$ 。

具体查找结点的算法如下:

```

PROCEDURE SearchNode( node, i, j, k );
{查找包含坐标点(i, j)且层数不超过 k 的结点,并返回
给 node}
BEGIN
  IF (i < 0) OR (j < 0) OR (i ≥ 2n) OR (j ≥ 2n)
THEN node := NIL; {坐标(i, j)超出了 DEM 范围}
  i1 := 0; j1 := 0; k1 := 0; node := root; {初始化参
数}

```

```

WHILE( NOT IsLeaf(node) AND k > k1 ) DO
BEGIN

```

{下层子结点定位}

```

  k1 := k1 + 1; iChild := 1;
  IF ( i ≥ i1 + 2n-k1 ) THEN BEGIN
    i1 := i1 + 2n-k1; iChild := iChild + 1; END;
  IF ( j ≥ j1 + 2n-k1 ) THEN BEGIN
    j1 := j1 + 2n-k1; iChild := iChild + 2; END;
  node := node^.Child[iChild];

```

END;

END;

如果要查找叶结点时,可取 k 为 n 。

3.4 线索化

二叉树线索化的过程可分为两个阶段:首先是遍历建立好的普通二叉树,寻找叶结点。其次是查找层数不低于它的邻接结点。遍历的过程与建立普通二叉树的过程类似,也可用一个递归的过程进行。DEM 二叉树中查找邻接点的方法有很多,如文献[6]的方法。主要有两类:回溯法和遍历定位法。

由于本文所述二叉树结点的相对位置与所代表空间位置一一对应,因此,可用上面所述的 SearchNode 过程直接遍历查找定位,比较的次数最大为树的高度。

以下是 DEM 二叉树线索化的算法:

```

PROCEDURE ThreadQuadtree( node, i, j, k );
{把第 k 层坐标在(i, j)与(i+2n-k-1, j+2n-k-1)之间的
DEM 二叉树线索化}
{调用方法:ThreadQuadtree( root, 0, 0, 0 )}
BEGIN
  IF IsLeaf( node ) THEN {叶结点则线索化}
  BEGIN
    SearchNode( node^.Child [1], i-1, j, k ); {查
找邻接结点作为线索}
    SearchNode( node^.Child [2], i, j-1, k );
    SearchNode( node^.Child [3], i+2n-k, j, k );
    SearchNode( node^.Child [4], i, j+2n-k, k );
  END
  ELSE {非叶结点则遍历查找叶结点}
  BEGIN
    ThreadQuadtree( node^.Child [1], i, j, k+1 );
    {递归以线索化二叉树各子树}
    ThreadQuadtree( node^.Child [2], i+2n-k-1, j,
k+1 );
    ThreadQuadtree( node^.Child [3], i, j+2n-k-1,
k+1 );
    ThreadQuadtree( node^.Child [4], i+2n-k-1, j

```

$+ 2^{n-k-1}, k+1)$;

END;

END;

线索化了的四叉树叶结点的邻接结点查找非常简单,就是叶结点的线索指针所指。

3.5 邻接结点定位

经线索化的四叉树可以很容易获得邻接结点的指针。从本结点定位信息及其邻接结点所在层数信息还可以计算出其邻接结点的定位信息。具体地,假设 DEM 大小为 $2^n \times 2^n$;本结点在四叉树的第 m 层,其所在区域为 $(p * 2^{n-m}, q * 2^{n-m})$ 到 $((p+1) * 2^{n-m} - 1, (q+1) * 2^{n-m} - 1)$,块大小为 $2^{n-m} \times 2^{n-m}$, $(p, q$ 的取值范围从 0 到 $2^m - 1)$;其 i 方向的邻接结点(非空)在四叉树的第 k 层(i 按图 2 从 1 到 4 取值, $k \leq m$)。则这个邻接结点所在区域的计算方法为:

$((p - 2^{m-k}) * 2^{n-m}, (q \text{ DIV } 2^{m-k}) * 2^{n-k})$ 到 $(p * 2^{n-m} - 1, (q \text{ DIV } 2^{m-k} + 1) * 2^{n-k} - 1)$ ($i=1$);

$((p \text{ DIV } 2^{m-k}) * 2^{n-k}, (q - 2^{m-k}) * 2^{n-m})$ 到 $((p \text{ DIV } 2^{m-k} + 1) * 2^{n-k} - 1, q * 2^{n-m} - 1)$ ($i=2$);

$((p + 1) * 2^{n-m}, (q \text{ DIV } 2^{m-k}) * 2^{n-k})$ 到 $((p + 1 + 2^{m-k}) * 2^{n-m} - 1, (q \text{ DIV } 2^{m-k} + 1) * 2^{n-k} - 1)$ ($i=3$);

$((p \text{ DIV } 2^{m-k}) * 2^{n-k}, (q + 1) * 2^{n-m})$ 到 $((p \text{ DIV } 2^{m-k} + 1) * 2^{n-k} - 1, (q + 1 + 2^{m-k}) * 2^{n-m} - 1)$ ($i=4$);

4 在地面可视化中的应用

地面的可视化方法有两类^[5]。一类是前向投影^[7,8]。它把基于 DEM 的纹理(遥感影像)向成像面投影映射形成可视化图象。由于其生成不符合扫描线顺序,因此占用的存储空间大,并要对映射值进行排序和归并;而且由于纹理映射的非线性,某些可视化图象象素内可能得不到任何纹理采样值^[9]。这类方法较适合于地面 DEM 网格本身的可视化。另一类是逆向映射^[5,10]。它是从成像面上每一象素点引出一条视线或光线,然后追踪此光线得到与 DEM 交点的位置及并获得对应的纹理值。这类方法的关键在于被追踪光线与对应视见点的 DEM 快速追踪和求交,亦即应设法减少无效求交的次数。

减少无效求交次数,提高光线追踪效率的途径有:DEM 分层次;为每一子区域建立尽量小的包围

盒。这正是本文所给出的经过改进的四叉树结构所能提供的。DEM 分层次(用四叉树表示)使光线追踪的无效求交次数大大降低;四叉树每一结点包含最大和最小高程信息又相当于给出了一个在 DEM 相应区域进行求交测试的最小包围盒,从而又大大减少了无效求交的次数。

本文用葛洲坝附近的 DEM 数据进行了实验,并得到了较为满意的结果。图 3 是葛洲坝附近之原始 DEM,它采用了右旁所示的斑马调色板(下低上高)。其大小为 256×256 ,高程范围为 $60 \sim 500$ 米,

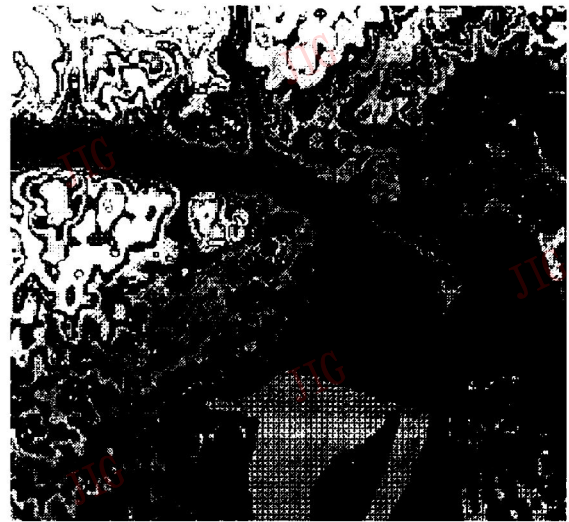


图 3 葛洲坝附近之原始 DEM(采用右边斑马调色板)
Fig. 3 Original DEM of the area of Gezhou Dam, China
(with zebra palette on the right)

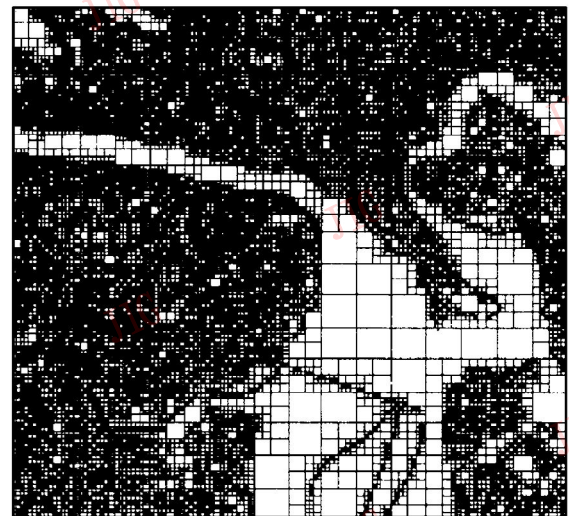


图 4 改进的四叉树结构
Fig. 4 The Improved quadtree

等高距为 10 米。

图 4 为图 3 的 DEM 改进的二叉树结构。其中,二叉树最大为 8 层,叶结点之最大和最小高程值相等。其结点总数为 50 609 个,相当于塔形分层表示的 58%。由图中也可见,在平坦地区(如江面)其二叉树结点数较少,而在起伏较大地区(如坡面)其四

叉树结点分布明显较多。如果把高程容限定为 50 米时,则其对应二叉树结点总数则只有 4 581 个。

图 5 为图 3 表示的 DEM 的地面可视化结果,其纹理采用了相关地区的 TM 影像。

此外,带有一定容限的 DEM 改进二叉树还可用于地面的分形可视化。



图 5 地面可视化结果

Fig. 5 Terrain visualization

5 结 论

改进的 DEM 二叉树比普通二叉树多包含了更多可加速可视化的信息,因此比普通的 DEM 二叉树表示更适于地面的可视化;其最大深度不大于塔形分层表示的层数,因此,在地面可视化应用中,光线追踪的效率比塔形分层表示高;它的结点中包含了最高高程信息,也包含了最低高程信息,这样,进行包围盒测试时,包围盒更小、更贴近 DEM。同时,对二叉树的线索化提高了邻接结点的查找效率。因此,光线求交效率得以大大提高;如果对结点中高程最大、最小值给一个容限,则二叉树深度会降低,存储所占空间也将大大减少,另外它还可用于地面的分形可视化;可对平坦地形的 DEM 进行压缩,但不适合于高程变化频繁的山区的 DEM 的压缩存储。

参 考 文 献

- 1 柯正谊等. 数字地面模型,北京:中国科学技术出版社,1993.
- 2 Sandor J. Octree Data Structures and Perspective Imagery, Computers and Graphics, 1985, 9(4), 393~405.
- 3 Peng Q. et al A Fast Ray Tracing Algorithm Using Space Indexing Techniques, Proceedings of EUROGRAPHICS'87, Amsterdam, 1987, 11~23.
- 4 Samet H. Applications of Spatial Data Structures, Addison Wesley, 1989.
- 5 Cohen D, Shaked A. Photo~Realistic Imaging for Digital Terrains, Computer Graphics Forum, 1993, 12(3), 363~373.
- 6 Samet H. Neighbor Finding Techniques for Images Represented by Quadtrees, Computer Graphics and Image Processing, 1982, 18(1), 37~57.
- 7 Graf K Ch et al : Perspective Terrain Visualization, Computers and Graphics, 1994, 18(6), 795~802.
- 8 Dubayah R O, Dozier J. Orthographic Terrain Views Using Data Derived from Digital Elevation Models, PE&RS, 1986, 52(4), 509~518.
- 9 彭群生,梁友栋. 真实感图形的计算机生成,计算机学报,1989,3, 226~237.
- 10 Paglieroni D W, Petersen S M. Parametric Height Field Ray Tracing, Graphics Interface'92, 192~200.



郝鹏威, 1988 年于西北工业大学计算机系获学士学位, 1994 年毕业于西北工业大学 CAD/CAM 中心, 获硕士学位。现为中国科学院遥感应用研究所博士生。主要从事计算机图形学、科学可视化、数字图象处理等研究。

An Improved Quadtree of DEM for Terrain Visualization

Hao Pengwei, Huang Bo, Zhu Chongguang

(*Institute of Remote Sensing Applications, CAS, Beijing 100101*)

Abstract Quadtree representation of DEM is of importance in DEM compression and terrain visualization. An improved quadtree is proposed, which has the advantages of both an ordinary quadtree and a pyramidal structure. Each node of the improved quadtree is added with a maximum height value and a minimum height value, and the whole quadtree is threaded, thereby making raytracing efficient in terrain visualization.

Keywords DEM, Quadtree, Threaded tree, Terrain visualization

(上接 577 页)

Ellipse Matching and Its Application to 3D Registration of Serial Cell Images

Zhang Yujin

(*Department of Electronic Engineering, Tsinghua University, Beijing 100084*)

Abstract In the process of 3-D cell reconstruction of serial sections through medical tissues, the registration of cell profiles in consecutive sections is a critical step. This paper presents an automatic registration method that builds up the equivalent ellipses of cell profiles with the help of the moment of inertia and then matches these ellipses to obtain the required geometric transformations. This method has been applied to the 3-D registration of megakaryocyte cells in human bone marrow specimens with satisfactory results.

Keywords 3-D Reconstruction, Image registration, Geometric transformations, Serial sections, Ellipse of inertia