

由计算机产生的等距离区域分界线

赖宝珍* 费立凡

摘 要

本文介绍一种最近由作者在美国研制的,由计算机产生等距离区域分界线的程序及其数据结构,同时给出初步的试验结果。

【关键词】 等距离;迭代;局部搜索;对分搜索

引 言

过去,为建立两个在制图定义上对等的区域之间的等距离区域分界线,尤其当分界线地带二侧区域轮廓变化复杂时,只能由制图员用手工方法在地图上反复寻找、量测和描绘,工具主要是圆规、直尺和铅笔。这种所谓试误方法不仅速度慢、精度低、成本高,而且又不能留下坐标档案,谈不上作进一步的处理。因此,本文提出了一系列算法,并进行了力求用计算机解决问题的若干探讨。

1 对问题的分析及基本设想

如图1,设 A , B 为待绘界线两侧两个不同的行政区域。 P 点介于 A 与 B 之间。若 U 点是 P 点到 A 侧的距离最近点, V 点是 P 点到 B 侧的距离最近点,且 $PU = PV$,则 P 点是等距离分界线上的一点。

经分析,还可从动态的角度进一步描述:所求的等距离分界线,实际上是介于两侧之间、随时分别与两侧轮廓线(包括属于一侧的岛态小区域)相切的“弹性圆”在贯穿待划界区的运动中其圆心的轨迹。

本文1987年9月18日收到。

*赖宝珍博士,美国马里兰大学地理系助理教授。

据此，我们对总体方案有了一个基本的设想：用手扶式高精度数字化器，分别将两侧轮廓线数字化，形成矢量式的数字化文件。计算机根据轮廓线类型，先确定一个起始等距离点（简称“起点”），从“起点”出发，朝某一方向，以步长 DENSITY 试探性地跨出一步，作为新的等距离点的最初试探位置，分别搜索两侧轮廓线，各找出离等距离点试探位置距离为最近的点，然后，计算比较这两个距离之差是否小于容差 EPSILON，若是，则这个圆心就被吸收为等距离分界线上的点；否则，重新调整试探位置，并重新搜索到两侧的距离最近点，再比较距离之差与 EPSILON……，如此迭代往复，直到试探位置符合等距离点的条件为止。吸收此点后，继续跨步试探，直到“弹性圆”

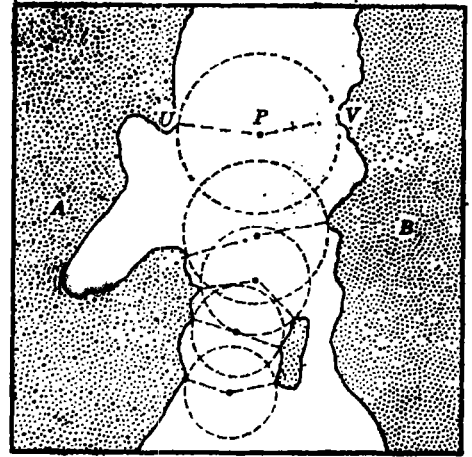


图 1 等距离点和弹性圆

在图内周游完毕。在“弹性圆”圆心运动过程中，随时有序地记录下被吸收为等距离分界线上的点的 X, Y 坐标值，建立坐标串文件，并用绘图子程序将两侧轮廓线及等距离分界线图形绘制在各种图形显示设备上（图形终端显示器、硬拷贝机、笔式绘图机等）。

2 贯彻基本设想的一些主要算法及其依据

下面就程序设计和调试过程中采用的一些主要算法、其理论依据、数据结构、困难所在及其对策，作一简单介绍。

2.1 3种不同的区域轮廓线的处理

对于需要作等距离区域分界线的行政区域，存在着3种不同类型的轮廓线：开放型、封闭型和半封闭型，如图2所示。



a 开放型

b 封闭型

c 半封闭型

图 2 3种不同区域轮廓线的类型

程序首先对原始数据进行判断，区分是哪一类的轮廓线。若是开放型，使程序一开始就在 A、B 间的区域中找到一个等距离点作为“起点”。其算法是：首先选定 A 侧的第 1 点作为一

个端点，接着轮流选取 B 侧中的各点作为第 2 个端点，依次计算并比较两点间的距离，记下使此距离为最小的 B 侧上的点 P_i ，然后选定 P_i 为一个端点，轮流取 A 侧坐标串中各点作为第 2 个端点，依次计算并比较两点间的距离，记下可使此距离进一步缩小的 A 侧上的点 P_i ……，依次迭代下去，直到取 A 、 B 两侧的其他数据点都不能使此距离进一步缩小为止。此时在 A 、 B 两侧分别寻找到了在局部范围内距离最近的原始数据点。因为轮廓线总可以看成是由短直线链段来表示的，而一点到一条线段的最近距离是从此点到此点在该线段上垂足的连线长（若垂足落入此线段内部），或从此点到该线段的某一端的连线长（若垂足位于此线段外部），因此，可进一步精确地计算出分别位于两侧、局部范围内为距离最近的两个点 p_u 和 p_v 。取这两点连线的中点，作为“起点”，“起点”被吸收为等距离线上的点。下一个等距离点的试探位置可通过“起点”，在垂直于 $p_u p_v$ 的直线上截取步长 $DENSITY$ 而得到。经若干次测试从试探位置到两区域各自的最近距离及调整跨步点的位置，使跨步点演变为等距离点而被吸收。第 3 个及以后各个跨步点的最初位置可以利用前两个等距离点的连线定向，截取、调整、吸收过程同前所述。运动至图边后，返回“起点”，向反方向跨步，也运动到图边。若两区域分界是封闭型，则从某一个分界点出发，向分界点左右相邻的两个线段的角平分线方向（显然这种方向有相差 180° 的两个）之一跨出一步，并根据从跨步点向下所作铅垂线与封闭多边形交点数的奇偶，自动判断出该跨点究竟是处于划界区域中还是主区域本身。若在主区域中，则取相反方向重新跨步。确认在划界区域中后，就可同样量测到两主区的距离最近点，调整跨步点位置，直到跨步点运动到图内的另一个分界点为止。若是半封闭型，至于哪一个方向是待划界区，单凭结构简单的输入数据是不能判别的，也就是说，存在着二义性。此时，计算机绘出原始数据图形以及两种可能的跨步方向，向用户发出询问，通过人机对话，找到合适的跨步方向，其余各步类似于封闭型，只不过跨步点的运动不是止于另一个分界点，而是止于图廓边。

2.2 全局搜索还是局部搜索

在迭代过程中，从试探位置出发，搜索两侧轮廓线上的距离最近点，是最经常而基本的过程。由于原始数据点的数量可能很大（例如，每侧上千个点是完全可能的），若采用全局性的顺序检测法，浪费的运算量可想而知。在寻找搜索算法时，我们从手工作业中得到了某种启示。手持圆规的作业员决不会为此进行全图的扫视，他的眼光总是在等距离点试探位置周围的某一局部范围内移动。如何使计算机在搜索两侧距离最近点的过程中也能自动确定出类似的一个局部范围呢？

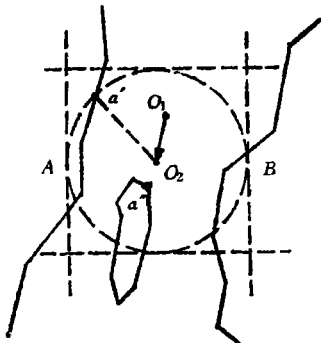


图 3 局部搜索范围

如图 3，假设当“弹性圆”的圆心在 o_1 位置时， a' 为 A 侧上到 o_1 的距离最近点。当圆心的试探位置为 o_2 时，为了重新搜索 A 侧上的一个距离最近点，可以 o_2 为圆心， $a'o_2$ 为半径作一个圆，只要在这个圆内搜索，定能发现 A 侧上的距离最近点。这是因为假设对 o_2 来说，在 A 侧上存在着异于 a' 的距离最近点 a'' ，则 $a''o_2$ 必然小于或等于 $a'o_2$ 。而凡与 o_2 的距离小于或等于 $a'o_2$ 的所有点都在该圆范围内， a'' 当然不例外。

为了更迅速地确定范围，避免不必要的计算，实际搜索时宁肯稍稍放宽这个尺度，而改为用此圆的外切正方形框作为搜索范围。凡落入此框内的线段都被取来计算一下，看 o_2 与它的距离最近点是在线段内部还是其某一端点，记下这个距离最近点。逐一比较正方形框内的所有线段的最近距离，从而找出离 o_2 最近的A侧上的点 a'' 。若没有异于 a' 的 a'' ，则至少可找到 a' 作为 o_2 到A侧的距离最近点。关于B侧距离最近点的搜索方法，与A侧完全相同。

2.3 实际搜索时采用的数据结构及其算法

为了方便用户，输入数据的结构应尽量简单。本程序所用的数据结构如图4所示。 XA 、 YA 、 $ICODEA$ 以及 XB 、 YB 、 $ICODEB$ 是对两轮廓线进行数字化后得到的一系列一维数组。与每段分离链段首点对应的 $ICODE$ 元素为0，与其余点对应的 $ICODE$ 元素均为1。将数字化得到的两区域轮廓线数据文件分别命名，程序在读入这两个文件的过程中，还要同时做3件事：（1）对各侧的链段数进行计数（因任何一侧都可能含有若干岛状小区），计数结果分别存入工作单元 AK 和 BK ；（2）将各侧各链段的首点在原始数据文件中的序号记入数组 IA 和 IB ；（3）将各侧各链段所含有的原始数据点的点数记入数组 IAN 和 IBN 。

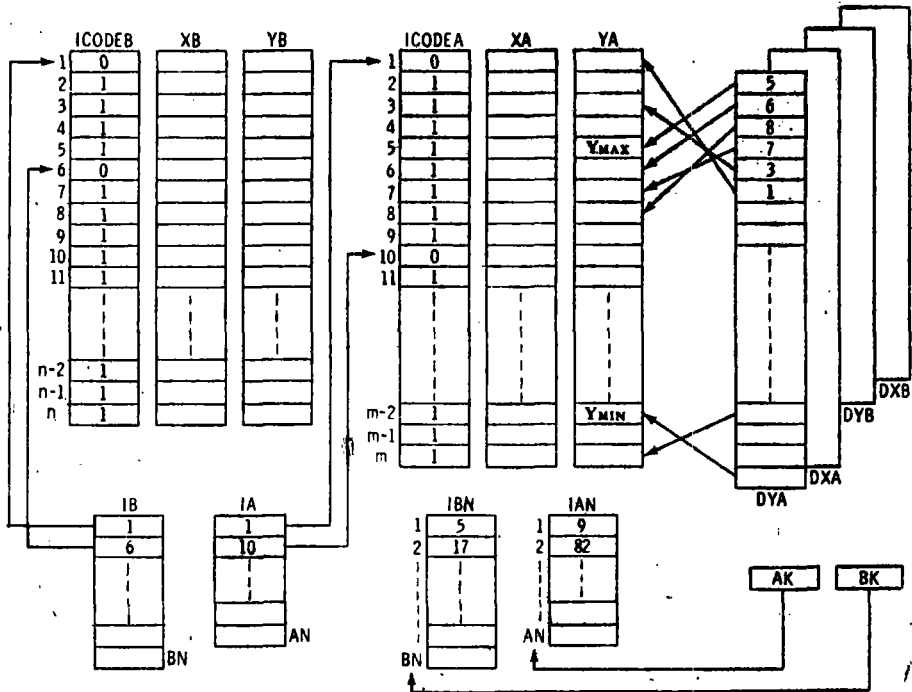


图4 数据结构

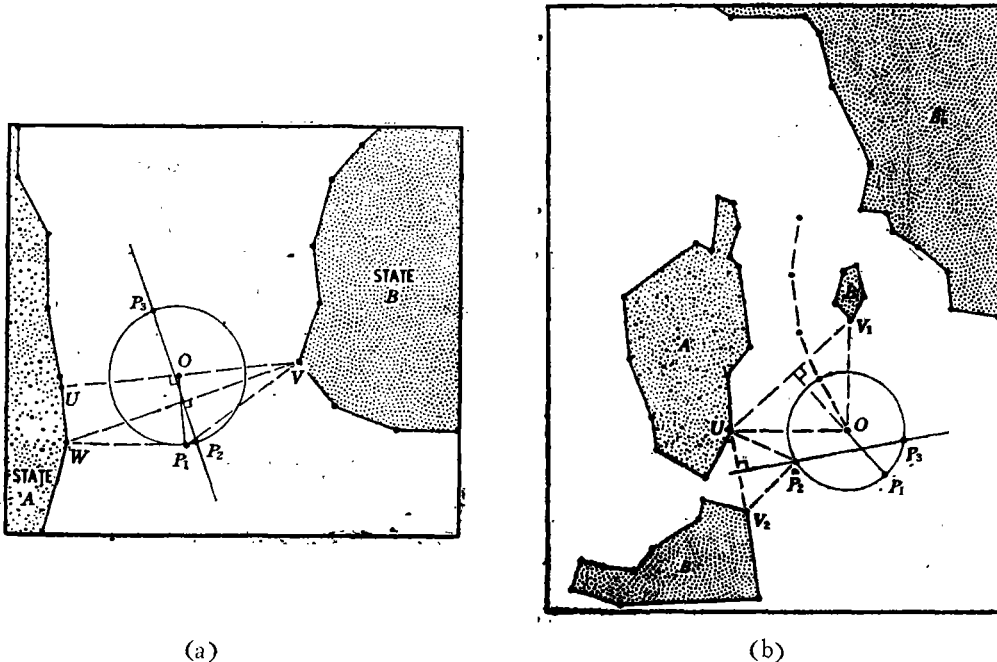
为了能在对大数组进行频繁搜索的过程中节约大量时间，采用了对分搜索法。为此，分别将 XA 、 YA 、 XB 、 YB 4个数组进行由大到小的排队。要得到的只是排队后反映相应坐标值在原始数据中序号的4个整型量数组： DXA 、 DYA 、 DXB 、 DYB 。例如，在图4中， YA 这个数组中的第5个分量最大，所以在对应的数组 DYA 中，第1个分量是“5”， $YA(6)$ 次之，故 $DYA(2) = 6, \dots$ ，依此类推。这里，对分搜索的目的是要求以搜索范围（上述正

方形框)的某一边(例如是 $YMAX$ 值)为上界,迅速确定上界在 DYA 中的位置。这与在一数表中寻找某一确定的值在算法上略有不同。上界位置确定后,下界位置的确定就采用从上往下的线性搜索,以简化程序。一旦遇到小于 $YMIN$ 的 Y 坐标值,则立刻停止向下的搜索。在上、下界之间的序号所对应的所有原始数据点,其 Y 坐标值均满足搜索框的上、下界。对这些点,再用 $XMIN$ 和 $XMAX$ 这两个左、右界加以筛选。对于筛选出的、位于搜索框内的所有点,在 DYA 中都在相应元素上作标记(加负号来区别)。然后只对涉及到这些端点的所有线段,计算从 o_2 到它们的最近距离点,并选出其中的一个最小距离点,作为 a'' 。处理完的线段,其端点序号恢复成正号。

那么在搜索 A 侧时,可否不用 DYA ,而用 DXA 呢?同样可以。至于究竟用哪一个为好,应当视 o_2 点附近划界区域的大致走向而定。如果大致是南北走向,用 DYA 进行搜索,效率较高。因为这样可以迅速筛选掉不在上、下界内的大量点,只要在剩下的少数点中再筛选出介于左、右界内的点。若大致是东西走向,则用 DXA 进行搜索效率较高。我们在程序设计中,利用了 o_1, o_2 的方向角正切值(看其绝对值是否大于1),作为判别附近区域大致走向的依据,由程序自动决定究竟搜索哪一个数组。

2.4 如何调整跨步点的位置

如图5(a)所示,设 o 为已找到的一个等距离点, $op_1 = \text{DENSITY}$, p_1 是跨步点的最初位置。在搜索两侧区域之后,分别得到与 p_1 距离最近的点 W 和 V ,此时,一般地说, $|p_1W - p_1V| > \text{EPSILON}$ (容差),在这种情况下,连接 W, V ,并作 WV 的垂直平分线;同时以 o 为圆心, op_1 为半径作圆,该圆与 WV 的垂直平分线有两个交点 P_2 与 P_3 。按某一种原则选取其中一点(例如,取与 P_1 的距离较近的那一点 P_2),作为调整后的跨步点试探位置。搜索两侧,再寻找与 P_2 距离最近的两点,重复上述过程,以跨步点的最新位置到两侧距离最近点(均为最新最近距离点)的距离之差小于、等于容差 EPSILON 作为迭代的终止条件。



这种确定及调整跨步点的迭代方法目的性强，收敛快，曾一度被认为是可行的。但经多次试验，发生过两个问题：（1）当直线 p_2p_3 接近竖直时，有时得不到 p_2 、 p_3 的精确位置，从而产生死循环；（2）如图5（b）所示，等距离点 o 到A侧及B侧的最近距离点分别为 U 和 V_1 ；而跨步点 p_1 到A侧及B侧的最近距离点分别为 U 和 V_2 ，若 $Up_1 \approx V_2p_1$ ，则连 U 、 V_2 ，并作其垂直平分线，与 $\odot o$ 交于 p_2 、 p_3 。若根据前例所述原则，因 p_3 离 p_1 比 p_2 离 p_1 近，而选择 p_3 作为下一个调整位置，则A侧及B侧到 p_3 的最近距离点又为 U 和 V_1 ，连 U 、 V_1 ，并作其垂直平分线，与 $\odot o$ 交于两点，根据就近选取的原则，将选取很接近 p_1 的那一点，而那一点到A侧及B侧的最近距离点又分别为 U 和 V_2 ……，也就是说，发生了死循环。

在这种死循环中，前者不是理论上的问题，解决办法是采用双精度变量，存放联立方程组

$$\begin{cases} y = k(x - x_0) + y_0 & (1) \\ (x - a)^2 + (y - b)^2 = R^2 & (2) \end{cases}$$

的系数以及在解方程组过程中所用的各中间结果。将（1）代入（2）后，可以求得两个 x 根。为了避免由于 k 的绝对值太大而得不到 p_2 、 p_3 的精确位置，宁可将 x 值代入方程（2）求 y 。此时，将出现4对坐标值均为方程根的多值情况。然后由程序选取出位于方程（1）所描述的直线上的两个点。

上述的第2种死循环表明：当求得 p_2 、 p_3 两个交点之后，仅按离原跨步点的距离远近来决定下一个试探位置的选取是带有一定盲目性的。其症结是忽略了等距离分界线与两侧轮廓线之间内在的拓扑关系。解决办法是在跨出第1步之前，记下两侧距离最近点分别在前进方向的哪一侧，由方位角差的符号记之。在求得 p_2 、 p_3 之后，看哪一个可较好地保持前进方向与两侧的拓扑关系，从而决定选取对象。

3 程序的总体结构

本程序用F77语言编写，包括一个主程序及14个子程序。程序的总体结构流程图见图6。

4 初步试验结果

图7中的6幅等距离分界线图都是在PRIME小型计算机上运行本程序得到的。其中3幅的区域轮廓线数据是模拟的，另3幅的轮廓线取自真实地图的数据。运行时间取决于数据量的大小和图形结构，所采用的容差 $EPSILON = 0.05$ 毫米（0.002英寸）， $DENSITY = 2.5$ 毫米（0.1英寸）。每幅图下边都注明了人工绘制及机助产生所需要的大致时间，其中机助处理的时间包括数据采集、处理及图形绘制。

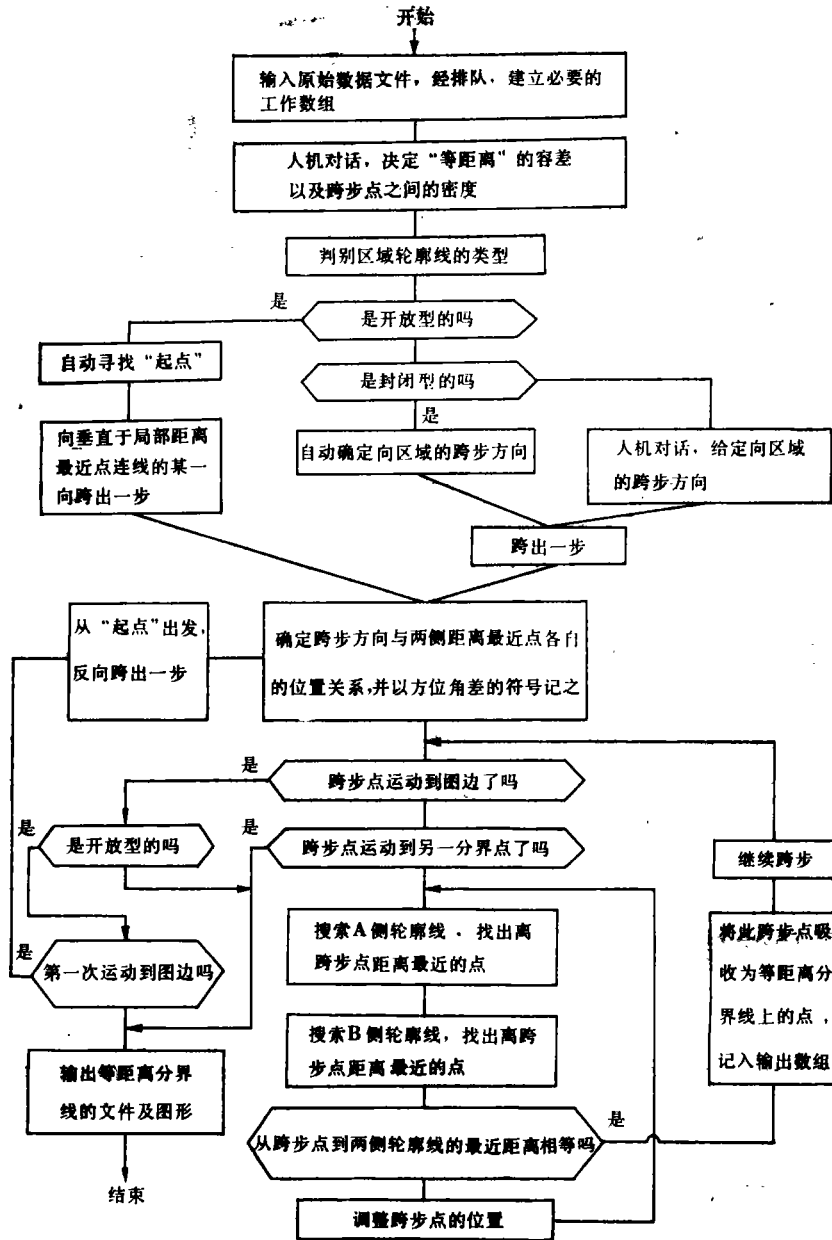
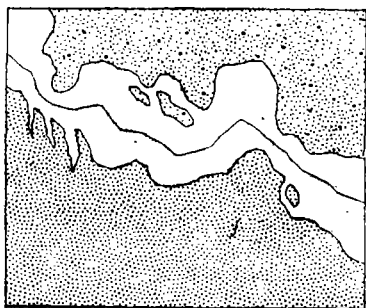


图 6 程序的总体设计

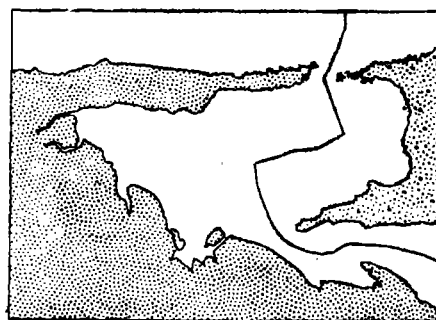
模拟地图数据

真实地图数据

开放型

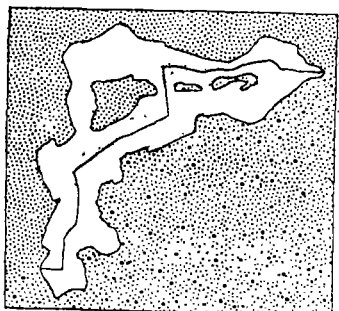


(a) 人工绘制: 4.25小时
机助处理: 1.25小时

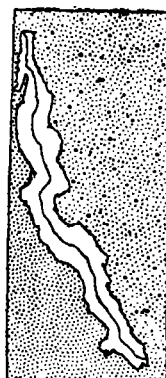


(d) 人工绘制: 8.5小时
机助处理: 2.25小时

封闭型

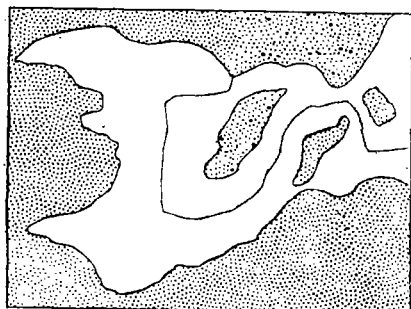


(b) 人工绘制: 4.0小时
机助处理: 1.25小时

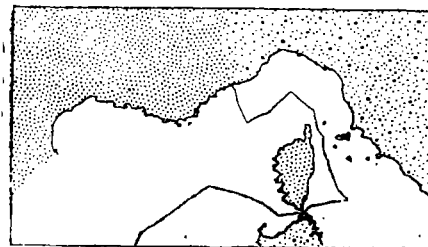


(e) 人工绘制: 7.0小时
机助处理: 1.75小时

半封闭型



(c) 人工绘制: 4.5小时
机助处理: 1.5小时



(f) 人工绘制: 8.0小时
机助处理: 2.5小时

图 7

5 結 語

由于计算机产生等距离区域分界线的解决方法缺乏统计制图或专题制图 (Statistical or Thematic Mapping) 的规律性, 故必须采用类似人工智能 (Artificial Intelligence) 的对策。如计算机自动判别待划区域轮廓线类型、寻找适当的起始点、在一定范围内对两岸进行搜索, 寻找距离最近点、用迭代法调整跨步点位置、自动判别跨步点落在主区域还是落在待划界地带之中等等。为了方便用户, 保持输入数据的简单结构, 在本文介绍的程序运行过程中, 采用了少量的人机对话。

以上实验结果还表明, 与手工作业方法相比, 计算机等距离分界线绘制系统在精度、速度、成本等方面的优越性是明显的。此外还有一些潜在的优越性, 如保留了图形的 $x-y$ 平面坐标档案, 便于进一步对数据作各种分析处理。例如, 可以根据给定的参数, 对等距离线进行综合, 使其成为更便于在实地定位的明显的折线状。也可以按不同比例尺, 整幅或分幅输出绘图结果 (这一点目前已经实现); 同时, 可以快速、精确、廉价地重现及复制结果, 作业员的劳动条件也大为改善。

参 考 文 献

- [1] Beazley, P. B., *Developments in Maritime Delimitation*, *The Hydrographic Journal* No. 39: 5—9, 1986.
- [2] *International Court of Justice, Delimitation of the Maritime Boundary in the Gulf of Maine Area, United States Counter-Memorial, Analytical-Annexes, Vol. IV, Annex 8, Chap. II, 1983.*

Computer-Generated Equidistance Line to Delimitate an Irregular Zone for Two Areas

Lai Poh-Chin Fei Lifan

Abstract

This paper presents an algorithm for the automation of cartographic construction of equidistance lines. The organizational structure of the digital data and unique characteristics of the application program are highlighted. Some working cases are provided to illustrate the results of this pioneering effort.

[Key words] equidistance; iteration; local search; binary search