

文章编号: 1005—0523(2005)05—0129—04

一种高效的位图显示方式

占自才

(华东交通大学 电气与电子工程学院, 江西 南昌 330013)

摘要:对于小的位图显示,并没有什么太多的技术问题,而对于大的(几百兆甚至上千兆)的位图显示,为了提高显示速度,在显示之前就必须要进行必要的处理.本文为了提高大位图的显示速度,采用了浏览位图哪里就把哪里的图像放入内存,当有缩小图像时,浏览到的图像范围就很大,就采用在显示之前把要显示的部分进行压缩好,放大时显示范围会很小,就把要显示的图像扩充好再显示.总之,在图像显示之前要进行预处理,这是本文的难点与创新之处.

关 键 词:显示速度;缩放;预处理
中图分类号:TP391.4 **文献标识码:**A

1 引 言

在各种 GIS 开发平台与一些图像显示平台中,图像显示是必不可少的.特别是 GIS 开发平台中,数据的矢量化都是以各种地形位图与卫星遥感位图为背景的,而这些栅格数据的显示速度对于 GIS 的终端用户又是至关重要的^[1].位图的显示有各种方式,各有各的特点. Photoshop 的图像浏览速度是最快的,不管图像的是大是小.但是对于特大图像来说,它要花很多时间进行初始化.在目前的家用计算机上,一幅 500M 的图像它要花上 6~9 分钟的时间进行格式初始化处理,用户显得有时会很不耐烦. Mapinfo 是一种 GIS 开发平台,它对栅格图像的显示是不用什么时间来进行初始化的,但在图像浏览与栅格图像进行矢量化时,对于一幅 50M 以上的图像,每拖动一下几乎都有等上半分钟左右,这对用户来讲是一件非常难受的事情.基于这些,我们在开发 GIS 软件的时候,采用什么样栅格图像显示方法就显得尤为重要了.

2 位图格式简介^[2]

位图结构包括位图文件头结构与位图信息头结构,共 54 字节.位图结构可参阅本节提供参考数目.

3 图像显示速度特点

图像显示的速度一般与显示图像的计算机硬件与软件相关.计算机主频高,显卡内存大,则显示速度快,反之则慢;软件设计方式不同,对图像的显示速度有决定性影响.所以,如何设计一个高效的图像浏览方式是本文创新与重点.

4 显示方式设计思路

我们设计图像显示软件时采用的是 VC++ 开发平台,这是因为在处理数据量大程序时,C++ 有它的速度优势.而 VC++ 有几个位图显示函数^[3],即: StretchDIBits 与 SetDIBitsToDevice.第一个函数

收稿日期: 2005—01—18
作者简介: 占自才(1969—),男,江西南昌人,讲师.

是带自动缩放拉伸的,第二个函数只能 1:1 显示位图.当要显示的位图比较小时,直接把整个位图的信息送入位图显示函数就能很好的显示效果.但对于非常大的数据图像,直接调用位图显示函数或者说把要显示位图信息送入位图显示函数就万事大吉的话,那是远远不够,这样不但有可能造成死机甚至出现 BUG.在浏览图像的时候,不管是浏览或是进行缩放,这些功能都由位图显示函数来处理了,而位图函数是把整个图像数据都写入内存或显存了,显然显存的大小往往是比图像数据要小得多,这就会造成显示的速度相当慢. Photoshop 采用了虚拟内存与重新对图像数据进行一种格式化以提高图像显示部分的索引效率,但它对整个图像进行一种格式化所花的时间有时叫人难以忍受的.本文采用的时在图像浏览到哪个位置的时候,就把这块图像进行位图格式化,并进行缩放处理.因为被显示到显示屏上的图像实际上是一个小位图.把这个进行处理后待显示部分图像做成位图结构送入位图显示函数,显示效率就和显示一个小位图的效率几乎一样了.因为计算机读写文件的速度与数据块的位置有关,所以本文在把要显示部分图像做成位图时比 Photoshop 要慢一些,因为 Photoshop 一开始就对位图进行了一种有利于读数据速度的处理.

总之,本文图像显示的过程是定好图像缩放倍数与位置以及在显示屏中的范围.当然,当有图像放大时,为了显示整数倍像素点,可以显示一定放大倍数且比显示屏大一点的范围.只不过是显示的起始位置可能在显示屏的外面一点.显示图像的原理图可以用图 1 来表示.

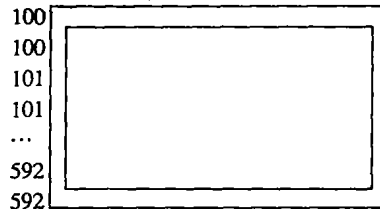
图中内方框表示显示屏,外方框表示截起的要显示图像的范围,数字表示截起图像在源图像的行列号.图像浏览时被浏览范围的图像被截起且被处理成小位图结构(比显示屏行列数稍大)放入内存用位图显示函数进行显示,位图显示函数无需再进行缩放处理,再显示之前都处理好了,具体在下节阐述.

5 图像浏览时预处理

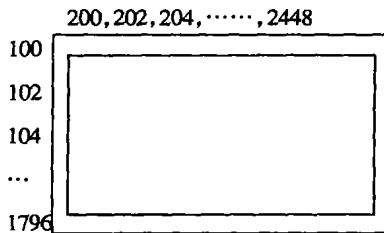
图像浏览时,必须知道在显示屏中显示的图像部分在整个图像中的起始点位置、缩放比以及被截起显示的图像部分在显示屏中的起始点.而计算机屏幕在程序设计时又是有逻辑坐标与设备坐标之分的,逻辑坐标是 VC++ 图像显示函数中所使用数学坐标,设备坐标是屏幕从左上角为(0,0)坐标

起,方向向右与向下而设置的坐标.两坐标的原定不一定是指它们的(0,0)位置,而是可以用原点设置函数可以随时设置的.原点设置函数分别是:

200,200;201,201;...;812,812.



(a)放大一倍时的图像显示



(b)缩小一倍时的图像显示

图 1 图像显示原理图

逻辑原点设置: $pDC \rightarrow \text{SetWindowOrg}(\text{CPoint})$;

设备原点设置: $pDC \rightarrow \text{SetViewportOrg}(\text{CPoint})$;

其中 pDC 是一个与显示设备绑定描述对象, CPoint 是逻辑原点与设备原点对象.但是不管什么时候,两原点在显示屏上都是指同一个位置点.这在后面阐述中会用到这些原理.

5.1 图像拖动处理

图像第一次显示可以是全范围显示或 1:1 显示,为了论述方便,假设第一次显示或拖动前一次显示是 1:1 显示.假设显示的图像是 $8\,000 \times 6\,000$ 像素的,屏幕为 800×600 逻辑原点用 CPoint pl 表示,设备原点用 CPoint pd 表示.假设前一次显示 $pl(100, 100)$ 、 $pd(0, 0)$,显示范围截起如图 2(a)所示.假设当前手抓把图像向右上角移动了 $(2\,000, 1\,000)$ 像素,这实际上是鼠标向左下角移动了 $(2\,000, 1\,000)$.则显示范围截起如图 2(b)所示.

显然,如果每一次都是整幅图像截起,只要把设备原点坐标减上鼠标前后坐标位置变化,重新设置设备原点,显示图像函数各参数无需变化,只要刷新一次就可以了.但实际是每次截起图像仅比显示的横竖方向多 200.这样,每次在源图像中截起图像部分的起始点位置只要改变一下,截起高度、宽度均不用改变.起始点位置只要加上鼠标向右下角移动的相对位置就可以了,而逻辑与设备原点坐标无需改变,显示函数各参数也无需改变,只需刷新就可以.确定了截起范围,就可把新截起的图像做成位图结构,放入位图

显示内存的数据，从新刷新视图，图像拖动浏览就实现了。当移动到边界时，因为截起图像的起始点不能为负数以及右下角的截起也不能出图像边界，这些都是

为负数以及右下角的截起也不能出图像边界，这些都是

是在程序设计时要考虑的。

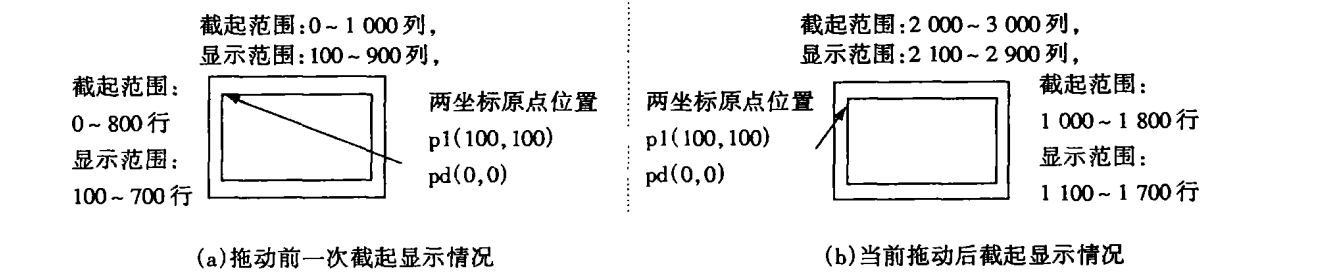


图 2 图像显示范围截起示意图

5.2 图像缩放处理

当图像有缩放时，截起图像起始点位置与范围以及对截起图像的处理方式将要改变。为了说明其原理，不失一般性，这里用缩小一倍与放大一倍来阐述。

5.2.1 图像缩小一倍处理

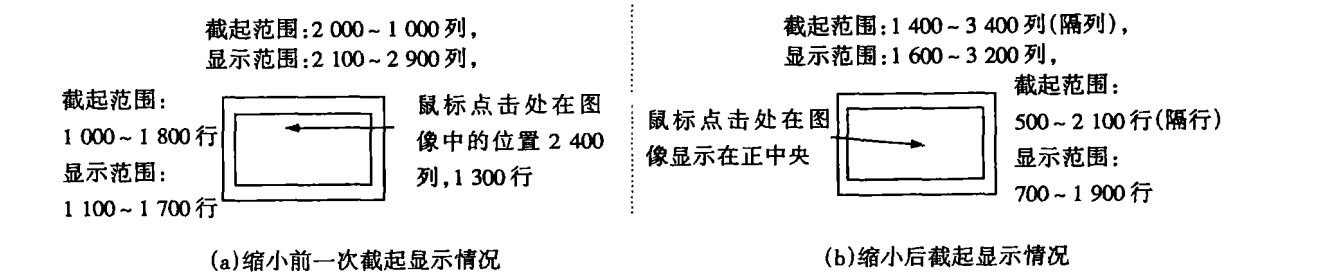


图 3 图像缩小一倍截起图像示意图

定位缩小且确定了截起图像位置及范围后，因为图像是缩小一倍显示，所以只要隔行隔列读取图像就可以了。把读取的数据做成位图结构放入相应内存，重新设置显示函数相应参数，重新设置逻辑与设备原点为 $p1(500, 400)$ 、 $pd(400, 300)$ ，正好是截起后图像中心位置与显示屏中心位置，被点击位置

的图像点已经显示在显示屏正中央且已缩小一倍了。

5.2.2 图像放大一倍处理

同图像缩小处理类似，将鼠标点击处放在屏幕中央进行放大一倍。图像截起位置与范围用图 4 来阐述。

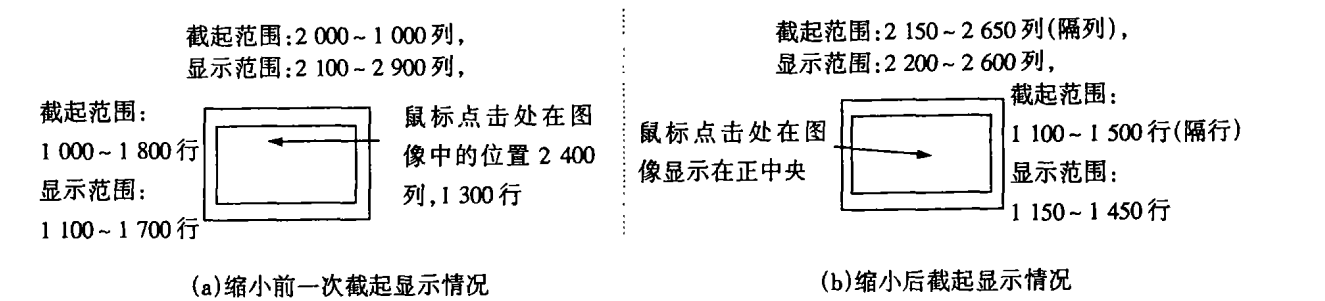


图 4 图像放大一倍截起图像示意图

定位放大且确定了截起图像位置及范围后，因为图像是放大一倍显示，所以被截起的数据每行每

列数据都要重复一遍写入位图内存，这样就可以保证每两行或两列数据一样，达到放大一倍的效果。

重新设置显示函数相应参数,重新设置逻辑与设备原点为 $pl(500, 400)$ 、 $pd(400, 300)$,正好是截起后图像中心位置与显示屏中心位置,被点击位置的图像点已经显示在显示屏正中央且已放大一倍了。

不管缩放多少倍,其原理是一样的,只是取像素的位置或间隔为小数取整后的数据,并且在程序设计方面的细节要考虑周到,当然这不是本文重点。

6 显示效果分析与结论

从上面的显示位图原理可知,浏览到图像哪里,就把能显示在显示屏中的图像略大一点的图像数据读入显示,这样就减轻了内存与显存的负担,显示速度当然就大大提高了。但是,每一次浏览图像都要重新读取数据,这在一定程度上有影响每次的浏览速度。读取同一分区或相近分区的数据速度是很快,我们基本感觉不出来,在图像放大时,要读数据量较小且距离较近,所以图像放大浏览时速度

度较快。而图像缩小浏览时,要读的数据量较大,但不管怎么缩小,在本文假设的显示屏下也不会超过 $3 \times 800 \times 1\,000$ 字节,只不过缩小越大,范围也就越大,数据相距就太大,这样也会大大地影响读取速度,所以缩小浏览图像时,浏览速度会较慢,但一般等待都在 1~2 秒之内。我们实际过程中设计一个导航器,大范围浏览图像时利用导航器,浏览就不会太慢了。

基于本文的原理做成的浏览图像软件,在浏览大图像时效率很高,图像浏览是开发 GIS 平台的基础,这就为我们以后的研究开发提供了保证。

参考文献:

- [1] 陈述彭,鲁学军,周成虎,等.地理信息系统导论[M].北京:科学出版社,1999.
- [2] 周长发.精通 Visual C++ 图像编程[M].北京:电子工业出版社,2000.
- [3] David J. Kruglinski, Scot Wingo, George Shepherd, 等. Visual C++ 6.0 技术内幕[M].北京:北京希望电子出版社,1999.

An Efficient Method to Display Bitmap

ZHAN Zi-cai

(School of Electrical and Eleceronic Engineering, East China Jiaotong Universty, Nanchang 330013, China)

Abstract: As for displaying a small bitmap, there is no technical problem. But as for displaying a big bitmap (several hundred or thousand megabytes), it is necessary to do some pre-process in ordre to improve the displaying speed. In this paper, a new method, which does some pre-process before displaying, is introduced to improve the displaying speed of big bitmaps.

Key words: showing-speeding; zoomout & zoomin; pre-process.