文章编号:1005－0523(2007)04－0064－03

海量影像数据的快速浏览设计

占自才

(华东交通大学 电子与电气工程学院,江西 南昌 330013)

摘要:信息技术和传感器技术的飞速发展,使得遥感图像的数据量呈几何级数的递增.当浏览图像或缩放时传统的方法速度都比较慢,特别是缩小图像时要读取整个影像数据,而且还要间隔的抽取数据,速度之慢几乎难以忍受,当对影像数据进行数据矢量化的时候,工作效率就非常低.本文提出了一种在打开文件时花出一定时间对影像文件进行临时文件重建,做成倒金字分层结构,并通过内存映射文件读取数据,简化了文件读取操作.这样每次浏览或缩放图像时就可以无等待了.

关 键 词:内存映射文件技术;倒金字塔结构;缓存技术;图像分层

中图分类号:TU45　　**文献标识码**:A

1 引言

随着信息技术和传感器技术的飞速发展,现代遥感数据源不断丰富,使得遥感影像图的数据量从以前的 10 MB、100 MB 增长到现在的 1 GB,甚至是 10 GB 以上[1].在这种情况下,如果对图像文件的处理仍采用传统的方法,显然是不行的.一方面,计算机硬件环境的发展已经远远不能满足图像处理的需要;另一方面,传统的利用文件指针方式来读取图像文件的方法,只能正确读取文件前面 $2^{31}-1$(约 2 GB)BIT,所以不能满足大于 2 GB 以上的影像文件的读取.在该文中提出的一种格式化海量图像并能使之快速被浏览的方法,实现该方法的主要技术包括内存映射文件技术、初始化影像文件建立临时倒金字塔影像文件、缓存技术、图像分层技术[2].这些技术是 GIS 矢量数据算法可视化的关键[1].

2 建立内存映射文件[4]

在对文件进行读写操作的时候,Win32 API 和 MFC 都提供了支持这些操作的函数和类,如 Win32 API 的 WriteFile(),Read2 File () 和 MFC 提供的 CFile 类.这些函数在文件的数据量小于 $(2^{31}-1)$ 字节的时候,在大多数场合是可以满足用户的需求.但是当文件的数据量大于$(2^{31}-1)$ 字节(约 2 GB) 时候,就不能正确地使用 Seek() 函数定位到要读取数据的位置,这是因为 Seek(LONG Loff, UINT nFrom) 函数提供的 Loff 参数是 LONG 型的,它最多支持$(2^{31}-1)$ 字节的偏移量.假如图像文件的数据量大于 2 GB 的话,就不能定位到 2 GB 以后的部分来读取数据.由于内存映射文件技术的主要函数都提供了两个 DWORD 型参数来分别表示文件大小或偏移量的低 32Bits 和高 32Bits,这两个参数加起来共 64Bits,因此它所支持的最大文件长度为 16EB($2^{64}-1$ 字节),几乎可以满足对任何海量数据文件的处理.下面对两个关键函数进行具体说明:

HANDLE CreateFileMapping(HANDLE hFile,//创建文件内核对象

LPSECURITY-ATTRIBUTES lpFileMappingAttributes,//返回的句柄可否被子程序使用

DWORD flProtect,//指定页面的保护属性(只读、读写、拷贝等)

DWORD dwMaximumSizeHigh,//文件大小的高 32Bits

DWORD dwMaximumSizeLow,//文件大小的低 32Bits,

和 dwMaxmumSizeHigh 组成 64Bits 值,来表示文件的

收稿日期:2007－05－27

作者简介:占自才(1969－),男,江西乐平人,自动化专业讲师,主要从事图像处理与智能控制方面的教学研究工作.

长度

LPCTSTR lpName);//文件映射对象的名字

该函数创建一个文件内存映射对象,以告知系统文件映射对象需要多大的物理存储器.由于内存映射文件的物理存储器实际上是本地硬盘上的一个文件,而不是从系统的页文件中分配的内存,所以 CreateFileMapping() 函数不分配进程的地址空间,不占用内存空间,因此在实际应用中通常是一次性把整个文件创建为内存映射对象.如果要把文件中的数据映射到进程的地址空间中,则需要调用 MapViewofFile() 函数.该函数的具体定义如下:

LPVOID MapViewofFile(

HANDLE hFileMappingObject,//CreateFileMapping ( ) 函数返回的文件内存映射对象句柄

DWORD dwDesiredAccess,//对页数据的访问方式

DWORD dwFileoffsetHigh,//映射开始位置的高 32Bits

DWORD dwFileoffsetLow,//映射开始位置的低 32Bits,和 dwFileoffsetHigh 组成 64Bits 值,来表示文件的偏移地址

DWORD dwNumberofBytesToMap);//映射的字节数,即在进程的地址空间中分配的字节数

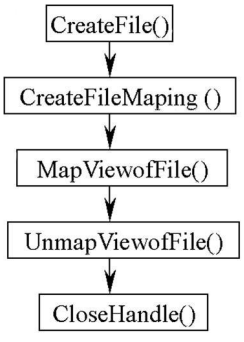

图 1　内存映射文件的步骤

调用该函数后,访问文件中的数据,就如同它位于内存中一样,这样就省去了对文件执行 I/O 操作的时间,所以它比用 CFile 类和 Win32API 的 WriteFile ( ),ReadFile ( ) 速度上要快得多.在使用该函数的时候,由于它的返回值 LPVOID 是一个 32Bits 的指针类型,所以在处理海量数据文件的时候,还要进行分块映射,并且一次映射到进程地址空间的数据量要小于 $2^{31}-1$ 字节数.文件内存映射的开始位置(即文件的偏移地址) 还必须满足是操作系统分配粒度的整数倍(通常操作系统的分配粒度是 64K).图 1 是使用内存映射文件的一般步骤.

3　分层金字塔结构

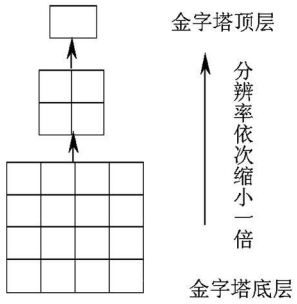

图 2　金字塔结构

为提高海量数据图像的浏览速度,这里将对原始图像建立分块的金字塔结构.根据不同的显示要求,调用不同金字塔等级中的数据来达到快速显示、漫游的目的.图像金字塔结构是一个多分辨率的层次模型,从金字塔的底层到顶层,其分辨率越来越低,相当于放大倍数越来越小.对应层的数据量也越来越小(一般情况下是呈 4^n 递减),所以在不考虑数据压缩的情况下建立金字塔后的图像比原图像增加大约 1/ 3 的数据量.图 2 是一个三层金子塔结构.在对文件建立金字塔的时候,只能是根据实际情况,首先对原始影像图进行分块.分块的规则是随意的,本系统为了避免对文件数据的来回映射,所以采用每 1024 列作为一数据块,这样可以顺序地对文件进行映射,最大限度地避免了对文件某些行、列的重复映射,使映射的次数更少,读取图像数据的速度更快.

4　通过文件映射方式建立金字塔图像文件

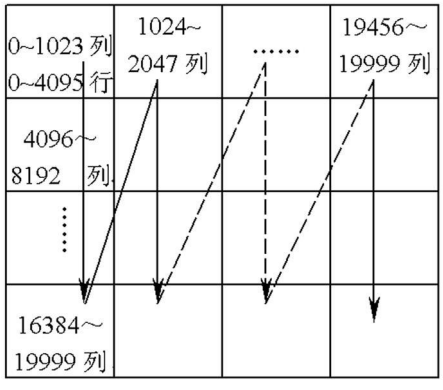

图 3　图像重新存放方式

按照第 3 节的方式建立被读取的图像映射文件与临时分层映射文件,把读取的文件分批映射到内存进程中,根据文件的位置及相应的分辨率把被读取的数据拷贝到临时文件内存映射相应的地址中.

假设每次映射被读取图像文件是4 096行,读取的数据又是以1 024列分块的,那么被读取的图像数据相应放在临时文件的位置以及在临时文件内存映射的位置可以用图示方式更容易说明(如图3所示).并假设图像是20 000×20 000的,为说明简便,并假设图像是8位单波段图像.

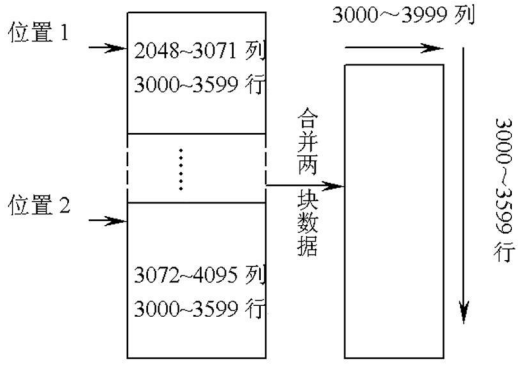

图4 选取被显示部分的数据

如图所示,被读取图像文件每次被映射4096行,依次进行.当每次映射的数据经过处理都要拷贝到相应的临时文件地址中,临时文件的大小可以计算得出.假设临时文件的金字塔顶层是原来的1/64倍分辨率,则临时文件的大小为(1+1/4+1/16+1/64)×20 000×20 000=531250000 bits.为了减少映射次数,每次映射到被拷贝的数据在经过处理后都拷贝到相应的临时文件内存地址中.数据处理方法如图3所示,解释如下:假设被读取的映射首地址是lprmem,并分配一个内存为4 096×1 024的临时存放数据的地址空间为lptmem,临时文件首地址为lpwmem,临时文件移动地址为lpmmem.假设现在是第一映射,把lprmem的每行的前1 024列全部拷贝到lptmem中,lpmmem=lpwmem,再把lptmem拷贝到lpmmem中,接着把lprmem的每行的第1 024列至2 047列的数据拷贝到lptmem中,lpmmem=lpwmem+4096×1 024,把lptmem拷贝到lpmmem中,依次进行处理完底层的前4096行.再把楼盘然么马的数据每隔一个字节提取一个数据组成缩小一倍分辨率的数据放在另一个内存地址中,不妨设为lprmem2中,把lprmem2中的前1 024列的数据整理到lptmem中,lpmmem=lpwmem+20 000×20 000,把lptmem的数据拷贝到lpmmem中.接着把lprmem2第1 024~2 047列的数据整理到lptmem中,lpmmem=lpwmem+20 000×20 000+4 096/2*1 024,把lptmem的数据拷贝到lpmmem中,依次整理到最后一列,再缩小一倍的数据也以此类推处理,这样,前面4096行的数据的1,1/4,1/16,1/64各分辨率的数据都拷贝到了临时文件相应地址中.以同样的方法处理后面各行的数据,并把它拷贝临时文件的相应地址中.这样临时金字塔文件就建立好了.之所以要建立分块分层的临时文件,是为了显示图像不再需要到原图像去读取数据.因为每一帧图像在原文件的跨度太大,读取速度相当慢.而重新组织的临时文件已经分块分层,自然每一帧图像跨度就小,浏览速度非常快[5].

5 图像显示

得到显示范围数据的基本步骤是这样的:根据图像显示的范围(起始点坐标、高度、宽度、缩放系数,其中坐标单位与高度、宽度的单位都为图像元素点),确定要显示的图像在临时文件的哪一层和哪一块(有时可能是两块).结合第4节的各种假设,本文再假设显示图像的起点坐标为(3 000,3 000)、高度为600、宽度为1 000、为简便起见缩放系数为1.根据这些就可以知道在临时文件中相应图像位置以及临时文件中要被映射的数据大小.位置可以计算如下:

根据缩放系数可以判定显示的数据在临时文件最底层,读取数据分两部分,先算第一部分.起始位置1=3 000/1 024×1 024×20 000+3 000×1 024(整型运算),读取大小为1 024×600.再算第二部分.起始位置2=(3 000+1 000/1 024×1 024×20 000+3 000×1 024(整型运算),读取大小为1 024×600.接着把两个位置的数据拼在一起,如图4所示.

通过相应的图像显示技术(VC++提供较好的图像显示函数或类,这里不予介绍)把相应确定要显示部分显示在视图内.每次更换浏览帧不会超过0.1秒.浏览效果相当好.

6 运行的结果

本程序在CPU为P4 2.2G Hz,内存为256MB的PC机上运行,打开一幅1.2GB的影像文件所花的时间大约为40秒.与Photoshop比要快20秒左右.

7 结束语

实践证明,采用内存映射文件技术读取海量图形文件的速度比采用Win32 API和MFC提供的文件操作函数和类读取文件的速度要快(下转第100页)

Research of Web Application Development Based on Ajax

YAN Li-Ping, YUAN Ke-Feng, SONG Kai

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: On the basis of the comparison with traditional model for Web applications, the paper puts forward a kind of model of Web application development based on Ajax, and analyzes the working theory and critical technologies of Ajax, which is then discussed in the realization of a Web voting system.

Key words: ajax; XML http request; DOM; web application

(上接第 66 页)得多,因为经它映射后,文件中的数据直接就存放缓存中了,省去对文件执行 I/O 操作的时间,此外它还可以正确读取大于 2GB 的图像,有效地解决了传统的文件指针只能读取 2GB 数据的限制;其次由于本程序使用了缓存技术,即每次在内存中的数据,只是影像文件中的一块,所以占用内存很小.再之,本文采用了临时文件是分块分层的倒金字塔结构,使得每次需要浏览的数据都可以在文件跨度很小的范围内找到相应的数据,使得图像浏览速度比较快,视觉无需等待.总之,采用本方法进行海量遥感图像的读取具有一定的优势.

参考文献:

[1] 孙家炳,舒宁,关泽群. 遥感原理、方法和应用[M]. 北京:测绘出版社,1997.

[2] 胡伟忠,刘南,刘仁义.基于内存映射文件技术的海量影像数据快速读取方法[J].计算机应用研究,2005,(2):111—112.

[3] 田杰,吴力合,吕建新. 地理信息系统中算法的研究[J].微计算机信息,2003,7.

[4] 希望图书创作室. Visual C++ 6.0 技术内幕[M].北京:北京希望电子出版社,1999,215—218.

[5] 吕京国,黄国满. 用 Visual C++ 实现大数据量的快速存取[J]. 测绘科学,2002,(9):29—31.

The Project of Quick Scanning Means for Large Scale Images

ZHAN Zi-cai

(School of Electrical and Electronic Engineering, East Chiua Jiaotong University, Nanchang 330013, China)

Abstract: The rapid development of information technology and sensor technology has given rise to the geometrically progressive in the case of remote sensing data. When images are scanned or images are zoomed, traditional method is very slow, especially when all images are zoomed, all images are read, images are got at intervals, whose slow speed is oppressive, and when the images are vectored, work efficiency is very low. This paper puts forward a project that it takes much time to rebuild image document, to make it pyramid structure, and to read the images through storage mapping file technique, so it is very simple. Thus, it is not necessary to wait when the images are scanned or zoomed.

Key words: storage mapping file technique; pyramid structure; cache technique; data block