

基于高斯金字塔的海量超大图像快速漫游算法^{*}

冯 莎, 卢选民, 陶旺林, 刘李娟

(西北工业大学 电子信息学院, 西安 710029)

摘 要: 针对敦煌壁画海量超大数字图像显示时打开速度慢、刷新困难等问题,提出了一种基于高斯金字塔分块分层数据组织算法。该算法在不损害图像的情况下,可以对图像中任意一块区域进行快速显示和浏览。实验表明,该算法能够实现海量超大数字图像的实时漫游,并且具有最佳分辨率及较高的平稳性。

关键词: 高斯金字塔; 图像分块; 图像分层; 图像漫游

中图分类号: TP391 文献标志码: A 文章编号: 1001-3695(2012)03-1141-02

doi:10.3969/j.issn.1001-3695.2012.03.093

Rapid roaming algorithm for massive image based on Gaussian pyramid

FENG Sha, LU Xuan-min, TAO Wang-lin, LIU Li-juan

(School of Electronics & Information, Northwestern Polytechnical University, Xi'an 710029, China)

Abstract: To resolve the problems of open slowly and flush difficulty for the massive Dunhuang fresco image, this paper put forward an algorithm of blocks and layers structure based on Gaussian pyramid. This algorithm could rapid browse and process any area of image but not damaging original image. Experiment results show that the algorithm can achieve real-time massive image data roaming, and can realize the best resolution and have higher stability.

Key words: Gaussian pyramid; image block; image layer; image roaming

敦煌莫高窟有 735 个洞窟,4.5 万平方米的壁画,是我国的艺术文化遗产。敦煌壁画数字图像的一个主要特点是数据量大,单幅图像的数据大小都为兆,有的甚至超过几百兆,如 tiff 图像。面对如此庞大的数据,要进行快速平滑的显示浏览、图像预处理、图像测量等将是十分困难的。这对计算机的硬件和软件都提出了非常高的要求。对于海量图像数据处理来说,最大的问题是要显示的图像总是会比计算机的内存大,也就是说图像数据不可能都放在内存中进行处理。那么如何组织在硬盘中的数据就成为了一个非常关键的问题。

目前比较成熟的技术是对影像进行分层、分块,即影像金字塔技术。该技术大量应用于海量图像数据库的建立和遥感影像图的浏览显示,国内外众多的 GIS 软件都采用了这种技术。例如,美国的 ERDAS 公司研制的 ERDAS IMAINE 以及国内超图地理信息技术的 SuperMap 等。

1 高斯金字塔图像构造算法

1.1 图像数据分层

高斯金字塔是以块为单位的一系列图像序列,其实现方法是将原始图像按照一定的压缩比例,将图像抽样生成出一系列图像序列,然后将图像序列按照固定的像素大小分块。分块后的数据按照一定的方式在磁盘上组织成便于系统查找和存取的文件。

图像的分层采样一般要考虑数据读取方式和图像采样方法。如果以原始图像为分层处理的数据源,每往上一层,需要映射到内存的数据是下一层的两倍,如从第 0 层到第 1 层,是

将两行数据中的 4 个像素合成 1 个像素,从第 0 层到第 2 层,是将四行数据中的 16 个像素合成 1 个像素。不但每层读取的数据量在加大,而且数据处理的复杂度也在增加。以上一层数据为数据源进行处理,只需要每次读取两行数据中的四个像素值。

因为涉及到 4 个像素,在图像上的横向跨度为 2 行数据,可以考虑使用内存映射文件将源图像的两行数据映射到内存,获得内存的起始地址指针 Src,设目标像素在目标图像里所在行的横坐标为 s,源图像每行的字节宽度为 Width,则在图像中的四个像素值为 Src[2 × s]、Src[2 × s + 1]、Src[Width + 2 × s]、Src[Width + 2 × s + 1]。然后采用某种采样算法将它们合成 1 个像素。对于图像的采样方法,大部分文献资料所采用的采样比例为 4:1,即将 4 个像素合成 1 个像素,同时也不乏 2:1 或者 3:1 等采样比例。采样比例的选择要根据系统应用的需要进行选择。一般不作图像处理,只用于显示的系统,采样比例都为 4:1,这样既可以保证采样时方便计算,也可以方便上下层之间建立父子关系。采样的计算方法一般有线性插值法、平均值法、最近邻域法等。结构如图 1 所示。

1.2 图像层数据分块

根据影像金字塔原理,除了对原始图像进行分层,还必须对分层后形成的图像序列进行分块。图像的分块一般要考虑数据读取方式、分块方式和分块大小。对于读取方式,可以每次读取图像层的一行数据,然后将属于该块的数据提取出来。这种方法增加了数据读取次数,减少了内存占用量。也可以一次读取多行数据,然后将属于该块的数据提取出来。这种方法减少了从磁盘读取的次数,但是加大了内存占用量。

收稿日期: 2011-07-26; 修回日期: 2011-09-01 基金项目: 国家文物局数字遗产保护基金资助项目(2009140-22/05)

作者简介: 冯莎(1989-),女,陕西渭南人,硕士研究生,主要研究方向为计算机图像学(fengsha700301@126.com);卢选民(1972-),男,副教授,博士后,主要研究方向为智能信息处理、多媒体通信与计算机网络等;陶旺林(1986-),女,硕士研究生,主要研究方向为计算机图像学;刘李娟(1986-),女,硕士研究生,主要研究方向为计算机图像学。

对于分块方式,目前主要有两种分块方式,如图 2 所示。一种是带状分块方式,这种方法是将图像数据的若干行作为一个单元形成一个带状。带状条可以顺序地编址并且可以单独索引。还有一种是块状划分,其方法是影像数据按照网格划分成小块,图像块通常是方形的,有时也可以是矩形的。相对于带状划分,块状划分更适合图像索引和镶嵌,而且划分后的图像块具有良好的聚集特性。

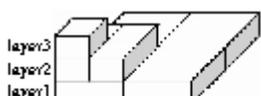


图 1 金字塔结构

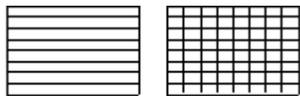


图 2 带状分块方式和块状分块方式

对于分块大小,通常取 2 的幂次方,具体的大小在选择时要考虑到操作系统数据存储格式的特点以及内存管理的最小单位。一般采用的分块大小为 128×128 、 256×256 、 512×512 、 1024×1024 。

本文采用的方法是对数据源中的一行数据创建内存映射对象,所要读取的块位于哪个行数据中,就将该行数据映射到内存,然后从该行数据取出需要的数据。因为是对一行数据读取,整个源图像有多少行数据就要创建多少个内存映射对象,读取完后又要释放多少次。但是内存占用会小很多。在计算中间文件头的时候,计算得到了每层图像的块高度和块宽度,在进行分块时分别对该层的每块图像进行处理,如第 i 层:

```
for (k = 0; k < TierHeightInBlock [ i ]; k ++)
```

```
for (s = 0; s < TierWidthInBlock [ i ]; s ++)
```

将第 (k, s) 块图像数据从该层中提取出来时,先利用内存映射文件,将该层图像映射为内存映射对象。然后在堆中申请一块图像块大小的内存,接下来就是从该层数据中读取属于当前块的每一行数据或 BlockPixelHeight 行数据,然后对 BlockPixelHeight 行数据进行处理: `for (int m = 0; m < BlockPixelHeight; m ++)`; 读取 BlockPixelHeight 行中数据当前块的数据,该行数据在层里的位置为

```
ReadPosition = ReadPosition + iWidth * (k * BlockPixelNumHeight + m)
```

只需要从这个位置提取属于当前块的那段数据,写入到分配的内存里面。遇到超出源图像宽度或高度,将该像素置为 0,完成 BlockPixelNumHeight 行数据读取后,就形成了一个数据块,总共要形成 $k \times s$ 个数据块。

1.3 建立影像金字塔模型

获取了分层分块的数据后,需要对数据按照一定的方式进行组织,使得数据的获取和查找变得方便容易。很多文献按照四叉树的方式组织数据,但是不能很快地完成数据的存取。图像在层中的排列方式如图 3 所示。

按照影像金字塔原理,形成影像金字塔后的文件大小将比原始影像大 33% 左右,因此它的数据也是海量的,在将图像块写入影像金字塔文件时,也需要将影像金字塔文件映射为内存映射对象,要写多大数据,就映射多大空间到内存中,按照文件头、各层头、各块头的顺序完成数据的写入。在文件结构的层头里,设置了 TierResolution 和 TierOffsetBits 字段分别记录当前层的分辨率和在整个影像金字塔文件中的偏移量。在块头里设置了 BlockOffsetBits 记录当前在整个文件中的偏移量。在将图像块写入到影像金字塔文件的过程中,需要计算每层的分辨率、偏移量和每块的偏移量。这些字段用于图像块的内存调度时,方便数据的查找和读入。

2 系统实现

本文提出的算法已经在敦煌数字拼接系统中实现。系统

针对敦煌图像的大容量数据测试效果如图 4 所示。系统基于 Visual Studio 6.0 开发,采用 VC ++ 做开发语言,支持的后台数据库包括 GDAL 函数库以及 OPENCV 数据库,大大提高了系统对大数据图片的访问速度。

(3,0)	(3,1)	(3,2)	(3,3)	(3,4)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)
(0,0)	(0,1)	(0,2)	(0,3)	(0,4)

图 3 图像在层中的排列方式



图 4 图像浏览

对于图像中感兴趣的区域,可以采用框选的方式进行选择。所谓框选,是指在原视区内,点击鼠标左键并移动鼠标,在桌面上出现随鼠标移动的以虚线为边界的矩形框,鼠标左键抬起,矩形框内图标被选中。所不同的是在本文中通过拖动鼠标对图像区域进行选中。在 VC 6.0 的平台下,本功能的实现主要是捕捉鼠标的左键落下、抬起、移动这三个消息。当按下鼠标左键时,通过消息函数获得矩形选框的起点坐标。移动鼠标,进行框选。当鼠标左键弹起时,通过消息函数获得矩形窗口的终点坐标。

框选只是在视区对感兴趣区域进行了标志,要进一步实现放大和处理功能就必须记录该区域在图像存储区域的位置和大小。在视区内,有一个标志感兴趣区域的框选区域,要求其对应的存储区域内的参数就要首先定义一个矩形 rc , 分别以鼠标落下点和抬起点作为矩形的左上点和右下点, $rc.left = start.x, rc.right = end.x, rc.top = start.y, rc.bottom = end.y$ 。这样就可以计算 ROI 区域的各个参量,图像的首地址用 m_pImage 表示,图像的宽度用 m_iX 表示,图像的高度用 m_iY 表示。这三个参量均为已知参量。则 ROI 区域的像素首地址 $p = m_pImage + (m_iY + rc.top) \times m_iX + rc.left$, 宽度 $Width = rc.right - rc.left + 1$, 高度 $Height = rc.bottom - rc.top + 1$ 。

从测试结果中看出,数据的传输速率比较高,系统稳定。实验证明,使用高斯金字塔模型对大数据图像进行显示在效率上是可行的,而且系统的平滑性和实时性也都达到了预期的效果。

3 结束语

本文针对“数字敦煌”中壁画数字图像的大容量特性,提出了采用高斯金字塔技术实现无缝漫游的算法,并利用 GDAL 函数库,最终以较高的效率来保证漫游的平滑性以及稳定性,使得在不损失图片分辨率的情况下,实现了图片的实时平稳漫游。它可以在大型关系数据库中高效地存储管理海量栅格数据,为后续需要的处理过程提供了一个基本框架与平台。

参考文献:

- [1] 冯宏华. C++ 应用程序性能优化[M]. 北京: 电子工业出版社, 2007.
- [2] 胡金星, 马照亭, 吴焕萍, 等. 基于格网划分的海量地形数据三维可视化[J]. 计算机辅助设计与图形学报, 2004, 16(8): 1164-1168.
- [3] 吕京国. 用 Visual C++ 实现大数据量的快速存取[J]. 测绘科学, 2000, 27(3): 29-32.
- [4] 李勇. 图像分割技术探究[J]. 科技咨询导报, 2007(5): 56-58.
- [5] 李军. 海量影像数据库的研究设计与实现[D]. 武汉: 中国地质大学, 2000.
- [6] 张琼声, 刘冬萍. 操作系统内核内存分配算法的分析与性能评价[J]. 计算机系统应用, 2007(1): 40-43.
- [7] HU Rong. Regulation of osteoclast differentiation by transcription factors MITF, PU. 1 and Eos[R]. [S. l.]: OHIO State University, 2007: 142-145.