

基于提升小波的地形数据混合熵编码压缩与实时渲染

郭浩然* 庞建民

(解放军信息工程大学计算机科学与技术系 郑州 450002)

摘要: 高分辨率地形高程和影像数据给交互式3维地形可视化应用带来沉重压力,主要体现在数据存储、调度传输及实时渲染等方面。该文设计一种基于提升小波变换与并行混合熵编码的地形数据高性能压缩方法,并结合图形处理器(Graphics Process Unit, GPU)Ray-casting实现大规模3维地形可视化。首先建立多分辨率地形块的小波变换模型来映射其求精和化简操作;其次,基于提升小波变换分别构建格网数字高程模型(Digital Elevation Model, DEM)和地表纹理的多分辨率四叉树,对量化后的稀疏小波系数引入并行游程编码与并行变长霍夫曼编码相结合的混合熵编码进行压缩;将压缩数据组织成多序列层进码流进行实时解压渲染。在GPU上基于统一计算设备构架(Compute Unified Device Architecture, CUDA)实现该文的提升小波变换与混合熵编码。实验表明,在压缩比、信噪比与编解码的数据吞吐量综合指标方面,该文方法优于其它类似方法。实时渲染的高帧率满足了交互式可视化的要求。

关键词: 数据压缩;提升小波;并行熵编码;图形处理器;地形可视化

中图分类号: TP274

文献标识码: A

文章编号: 1009-5896(2012)12-3013-08

DOI: 10.3724/SP.J.1146.2012.00652

Terrain Data Hybrid Entropy Coding Compression Based on Lifting Wavelet and Real-time Rendering

Guo Hao-ran Pang Jian-min

(Department of Computer Science and Technology, PLA Information Engineering University, Zhengzhou 450002, China)

Abstract: High resolution terrain Digital Elevation Model (DEM) and orthophoto bring severely load including data storage, schedule and real-time rendering, etc.. A high performance terrain data compression method is proposed based on lifting wavelet transform and parallel hybrid entropy codec, and combined with Graphics Process Unit (GPU) Ray-casting to achieve large-scale 3D terrain visualization. First, the multi-resolution wavelet transform model of terrain tile is constructed to map the refinement and simplification operation. Then the multi-resolution quadtree of DEM and terrain texture is built separately based on lifting wavelet transform, the sparse wavelet coefficient generated from quantization is compressed by a hybrid entropy codec which combined with parallel run-length coding and variable-length Huffman coding. The compressed data are organized into progressive stream to do real-time decoding and rendering. The present lifting wavelet transform and hybrid entropy codec is implemented by Compute Unified Device Architecture (CUDA) in GPU. Experiment results show that the data compression ratio is effective with this method, PSNR and code-decode data throughput. High Frames Per Second (FPS) in real-time rendering satisfied the demand of interactive visualization.

Key words: Data compression; Lifting wavelet; Parallel entropy coding; Graphics process unit; Terrain visualization

1 引言

大规模地形可视化系统广泛应用于军事领域的战场态势模拟、作战训练仿真和民用领域的虚拟游园、3D游戏等方面。当前对3维地形场景模型的需求不再限于模拟地貌和晕渲着色,更多依托于现代测绘技术所生产的海量高分辨率DEM(数字高程模

型)和正射影像地表纹理来构建真实感大规模3维地形场景。对渲染真实感和交互实时性的要求,使得海量数据集(包括DEM和地表纹理)与有限的计算机硬件处理能力之间的矛盾日益成为当前交互式大规模地形可视化系统面临的关键问题。地形数据高性能压缩、高效传输与快速解压缩渲染成为近年来该领域研究热点。

小波变换方法已经引入到对地形数据的处理。小波的局部化分析特性、多分辨率特性和紧支性等属性“天然”适用于多分辨率模型的构建^[1,2]、3维几

2012-05-29 收到, 2012-09-13 改回

国家863计划项目(2009AA012201)和信息工程大学博士学位论文创新基金(BSLWCX201103)资助课题

*通信作者: 郭浩然 haoran006@yahoo.com.cn

何数据的压缩^[3,4]以及模型的层进传输^[5,6]。其中,文献[4]使用小波变换系数的位面编码技术结合最佳截断嵌入码块编码(Embedded Block Coding with Optimized Truncation, EBCOT)算法压缩DEM数据,编码计算复杂度很高,不利于实时渲染。文献[3]将小波作为指导精细化层次细节(Level-of-detail, LOD)建模的标准,指导地形LOD渲染,但未引入对地形数据的压缩。文献[5]基于提升小波建立地形块累进模型,在GPU上实现数据压缩和累进传输,并将小波变换与粗粒度LOD渲染结合,具备较高的实时解压效率。但基于间接查找表的压缩方式不能有效消除数据相关性,压缩效率不高。此外,当前基于GPU的通用计算(GPGPU)发展迅猛,CUDA架构使得GPU的可编程性及流处理器特性进一步增强。文献[7]给出的CUJ2K是基于CUDA实现的JPEG2000编码器的例子,能达到22 Mpix/s的图像编码压缩吞吐量,文献[8]给出了CUDA实现的GPU并行霍夫曼编码器,比CPU实现的串行方法有30~50倍的速度提升,但遗憾的是都未给出相应的解码器。

本文基于提升小波,建立地形数据的小波变换模型来映射地形构网的求精和化简操作;通过提升小波变换分别构建数字高程和正射影像的多分辨率四叉树,并将量化小波系数存储在四叉树节点中;基于CUDA实现了并行游程编码器对量化小波系数进行压缩,进而采用并行的霍夫曼编码器对游程编码后的数据进一步压缩。混合熵编码后地形数据组织成多条流层进递交给解码和渲染进程,最终实现大规模3维地形场景的高性能数据压缩、调度及交互式渲染。

2 地形数据的小波变换模型

2.1 提升小波机制

小波变换本身并不能进行数据压缩,在其对数据进行分解成空间和频率不同的子集之后,对其系数进行量化与特定编码能实现压缩/解压缩的目的。假设 $\{\dots V_n \subset V_{n-1} \subset \dots \subset V_1 \subset V_0 \subset \dots \subset V_{-n} \dots\}$ 为2维空间的一系列嵌套函数子空间序列集合,小波分析可获得函数 $f(x,y)$ 在任意子空间 V_i 上的近似表示 $P_i f(x,y)$ 。两个相邻空间的近似差(approximation difference)可在相应的补子空间 C_i 中表示。从子空间 V_0 开始经过有限层次的小波分解,函数 $f(x,y)$ 可表示为

$$f(x,y) = \sum_i \sum_j \delta_{i,j}^M \cdot \phi^M(x,y) + \sum_{m=1}^M \sum_i \sum_j (\lambda_{i,j}^{m,1} \cdot \varphi^{m,1}(x,y) + \lambda_{i,j}^{m,2} \cdot \varphi^{m,2}(x,y) + \lambda_{i,j}^{m,3} \cdot \varphi^{m,3}(x,y)) \quad (1)$$

对应的尺度函数记为 $\phi(x,y)$, 小波函数记为 $\varphi^1(x,y)$, $\varphi^2(x,y)$ 和 $\varphi^3(x,y)$, M 为分解尺度。式(1)表示函数 $f(x,y)$ 在任意空间的近似表示与相邻空间近似差的累积之和。尺度系数 δ 能表征函数的大致形态,小波系数 λ 表示对应方向上的细节变化。数据(或信号)经小波变换后,能量集中在低频(逼近)系数上,并使高频(细节)部分大量系数的绝对值趋于零,分别对低频系数和高频部分进行量化、编码即可达到压缩的目的,这就是基于小波变换对数据进行压缩的原理。

小波提升机制基于某个基本小波分析,在主尺度函数不变的情况下,对式(1)中的小波空间进行改进,使之具备更高消失矩和平滑性,以保证变换后的数据集能量分布更集中并兼顾数据的平滑重建。基于相应的尺度函数和小波系数,在主尺度函数不变的前提下,经过1维提升的小波分解公式可表示为 $\lambda_{j+1,l} = \sum_k \tilde{\mu}_{k-2l} \delta_{j,k}$, $\delta_{j+1,l} = \sum_k \tilde{\eta}_{k-2l}^0 \delta_{j,k} + \sum_k s_{l-k} \lambda_{j+1,k}$ 合成表示为

$$\delta_{j,k} = \sum_l \eta_{k-2l} \left(\delta_{j+1,k} - \sum_k s_{l-m} \lambda_{j+1,m} \right) + \sum_k \mu_{k-2l}^0 \delta_{j+1,k}$$

进一步地,系数间的关系可由分裂、预测和更新3个步骤实现。即在系数分解过程中, δ_i 首先按照基本小波变换关系分裂成两个集合 δ_{i+1}^0 和 λ_{i+1}^0 , 然后由 δ_{i+1}^0 预测 λ_{i+1} , 得

$$\lambda_{i+1} = \lambda_{i+1}^0 - P(\delta_{i+1}^0) \quad (2)$$

由式(2)计算 λ_{i+1} 的更新系数 δ_{i+1} , 得

$$\delta_{i+1} = \delta_{i+1}^0 + U(\lambda_{i+1}) \quad (3)$$

相应的小波逆变换过程表示为

$$\left. \begin{aligned} \delta_{i+1}^0 &= \delta_{i+1} - U(\lambda_{i+1}) \\ \lambda_{i+1}^0 &= \lambda_{i+1} + P(\delta_{i+1}^0) \end{aligned} \right\} \quad (4)$$

任意数据集经过小波变换都可以得到一系列不同尺度的近似表示,每个尺度都可等价于相邻上个尺度的简化。

2.2 变换模型建立及小波选取

地形区域的规则格网数字高程可表示为 $\{E_{i,j}^0, 0 \leq i < N_1, 0 \leq j < M_1\}$ 。对该数据集进行2维小波分析,高程采样点与初始尺度系数相对应,根据小波变换特征分析,尺度系数集合 $\{E_{i,j}^M, 0 \leq i < 2^{-M} N_1, 0 \leq j < 2^{-M} M_1\}$ 可视为原数据集 M 尺度的逼近模型,小波系数集合 $\{c_{i,j}^{m,d}, 0 \leq i < 2^{-m} N_1, 0 \leq j < 2^{-m} M_1, d = 1, 2, 3\}$ 包含 M 尺度简化的2个相邻层次间损失的信息。

地形模型的化简和求精过程可映射为对应数据集的小波变换及其逆变换。图1表达了本文地形块小波变换模型,对原始地形块进行一级小波变换过

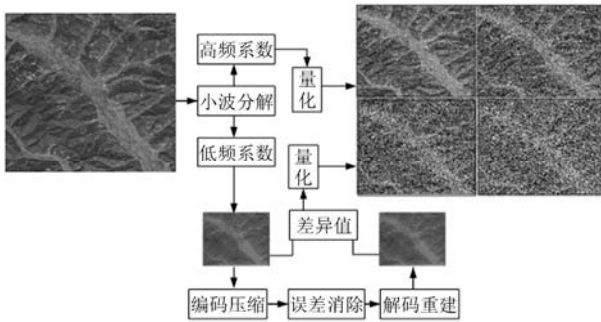


图1 地形块的小波变换模型

程中，经过行和列两个方向上的高通滤波和低频滤波，得到低尺度上的概貌信息和行、列方向上的细节信息。对高频系数和低频系数进行量化，生成一系列包含不同粗糙层级的原始地形块近似表达。简化的细节信息包含在小波系数矩阵中；相反，粗糙的地形块结合细节信息，通过小波逆变换进行求精，合成精细的地形层次，甚至原始地形。该过程是完全可逆的。

规则格网DEM和对应区域的遥感正射影像是构建真实感3维地形场景的基础数据，目前它们大多以高分辨率的图像格式发布。学术界对基于小波的高效图像压缩方法研究由来已久。但不同与普通的静态图像压缩，DEM要准确反映地形地貌和高度变化，一般不允许压缩有较大的信息损失，要求无损压缩或近无损压缩；此外，考虑到规则格网DEM数据的特点，选取的小波应对称而平滑，同时具有一定的消失矩。因此，本文选用以Lazy分裂为提升基础的5/3整数提升小波^[9]。遥感正射影像在进行提升小波变换之前首先要将其RGB色彩空间转换到合适的色彩空间。这是因为原始正射影像的RGB色彩空间具有很强相关性，而理想的色彩空间应使得能量尽可能集中在一个分量上，图像各分量之间的相关性尽可能小，以利于更有效地对其压缩编码。本文将正射影像的RGB分量转换到YCoCg色彩空间，对亮度信息(Y分量)和色度信息(Co, Cg分量)分别进行小波变换。遥感正射影像具有纹理复杂的特点，对其进行小波变换需选取方向敏感性更高的变换基，从而更适合于其的纹理特征处理。基于此考虑，本文选取方向ADL(Adaptive Directional Lifting)小波^[10]对遥感正射影像进行小波变换。

3 基于小波变换的地形数据多分辨率模型

3.1 地形数据多分辨率四叉树的构建

在原始地形区域被划分成一系列地形块(tile)的集合之后，针对每2×2邻接地形块，应用图1模型生成它们的低分辨率近似表示，并建立对应的小波系

数矩阵。2×2邻接地形块作为低分辨率近似表示的子节点，低分辨率的近似表示被合并成新的父亲节点，多分辨率地形块四叉树就以这种自底向上的方式递归建立，直到达到自定义的深度为止。图2所示的四叉树即地形数据多分辨率模型，针对DEM和正射影像分别应用该模型建立各自的四叉树。小波系数矩阵包含有近似表示与原始数据之间的差异值，存储在四叉树每一层的节点中。

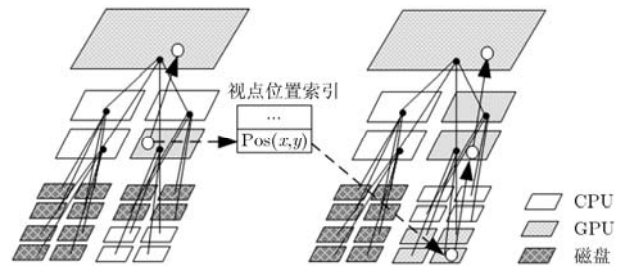


图2 地形数据的多分辨率模型及视相关的数据层进传输

3.2 层进传输

以视点改变为制导进行多分辨率模型的数据层进调度与传输。视点改变时，观察相机挂载的视锥剔除函数进行可见性判断，处于视锥中的地形块引起多分辨率模型的构网变化。当视点快速改变时，构网引发的频繁和大量地形块调度与传输会占用很高的缓存及计算器件传输带宽。层进结构可有效解决这种问题。小波变换后的四叉树每一层节点中都存储有包含其低层的细节信息，假设四叉树第L层的地形块被视锥选中，那么可以将第L+1层的父节点作为索引递交给相机，父节点的近似表示结合子节点的细节信息完成满足分辨率要求的构网。如图2所示。这样的拉动式层进处理使得传输的数据量大大减少，而且流式数据传输也提高了“磁盘-CPU-GPU”之间的数据吞吐量。

4 并行混合熵编码

小波变换之后要对系数进行量化，使之满足某种稀疏特性，之后对量化后的系数稀疏矩阵进行编码以实现数据压缩。四叉树多分辨率地形模型构建之后，节点的浮点型细节系数 C_i 需要通过标准的标量定步长量化方法量化成整型数值 c_i ， $c_i = \text{sign}(C_i) \cdot \lfloor |C_i| / \sigma \rfloor$ ，式中 σ 是量化步长。量化后的小波系数组织成包含大量0或者很少实数的流数据，采用游程编码对这种具备稀疏特性的流数据进行编码；对游程编码后的数组作为信源符号再采用霍夫曼编码。在GPU流处理器架构下基于CUDA充分利用数据并行，实现上述混合熵编码的并行化。

4.1 引入并行前缀和扫描的游程编解码器

游程编码的基本思想是将具有相同数值且连续出现的信源符号组成的符号流串用其数值及串长度表示。把量化后的稀疏小波系数作为信源时，连续的 L 个0值对其作游程编码只需传送一个数组 $(L, 0)$ 即可代替传送这一串的0值，这实现了对数据的压缩传输，这些连续的0值称为游程。本文关注输入流中0值符号的游程，在编码阶段，对输入流中的每一个非0符号，由一对包含原始符号及该符号之前的0值计数来代替。如图3(a)所示，首先扫描输入流，其中的0值符号标记为0，非0值符号标记为1；然后，针对标记数组进行游程单元的数据划分，构造游程的原始符号及其输出索引。

上述过程中，扫描输入流并划分游程单元是运算密集的，对一个 n 元输入数组，最坏时间复杂度能达到 $O(\lg 2n)$ 。当前NVIDIA推出的主流GPU的Femi, Tesla硬件体系结构和CUDA软件架构为基于GPU的通用计算提供了优良的支持。所以，基于当前GPU的并行处理能力，本文引入并行前缀和(parallel prefix sum)扫描原语^[11]实现对输入流游程单元数据划分的加速计算。

给定一组有序元素 $[a_0, a_1, \dots, a_{m-1}]$ ，按顺序计算这些元素的部分和 $[0, a_0, (a_0 + a_1), \dots, (a_0 + a_1 + \dots + a_{m-1})]$ 。并行前缀和构造一个 $\log_2 m$ 层的平衡树，包括从根节点向叶节点的剪枝扫描过程和相反的向下扫描过程。在剪枝扫描阶段，将每一层两个相邻叶节点相加并将值存在后一个叶节点中；而在第2阶段，在每一层将两个相邻节点值互换，然后将二者的和存在后一个叶节点中。这一算法共需要 $m/2$ 个线程，计算复杂度是 $O(n)$ 。

运行并行前缀和扫描之后，标记数组中的非0符号的游程被计算出来，如图3(a)中的(1,4,3)及其索引(1,7,8)组成游程(1,1), (4,7)和(3,8)，即非0符号及其在新数组中的下表索引；对于索引(1,7,8)，通过下标算子 $A[i] = A[i-1] + 1$ 计算出索引所指示的非0符

号之前的0计数，即(1,5,0)。为充分利用后续霍夫曼编码，0计数不能太大，否则会增大霍夫曼编码的最大码字长度。为了避免这种情况，需要在经过游程编码后的符号流中0计数超过给定门限(255)的位置插入额外的0；对每组索引计算出需要额外插入的0值数目，再次对0计数运行并行前缀和扫描，获取新的索引，最后输出压缩编码。

解码器如图3(b)所示。编码器的输出数据流是解码器的输入序列。首先对0计数运行并行的前缀和扫描，得到非0符号对应的临时位置向量，然后以下标算子 $A[i] += i$ 计算出非0符号在输出向量中的目的地址；对非0符号运行分散操作将它们写到输出向量对应的地址中，完成解码。

4.2 并行霍夫曼编解码器

为进一步压缩游程编码后的流，本文基于CUDA在GPU上实现了一种细粒度数据并行的霍夫曼编解码器，其主要包含以下3个并行步：

- (1)为输入数据流分配码字；
- (2)为码字计算输出比特位置；
- (3)将码字存入压缩数据流。图4(a)解释了编码3个步骤的主要操作。具体操作如下：

第1步 为输入流数据分配变长码字。不失一般性，假设码字存储在查找表里，表中的每个实体包含两个值：码字的二进制代码和码字长度(bit)，记为 $(cw, cwlen)$ 。在压缩过程中，输入数据流的每一个符号被相应的变长码字替换，线程束(warp)中每个线程处理一个输入数据单元，线程装载输入数据流单元与运行码字查找是并行进行的。当前的GPU计算架构对32位数据类型提供高效的支持，输入数据被划分为32位无符号整型装载入共享存储，在共享存储中由线程块对其进行处理。32位数据的值被分成4字节符号，然后被分配给查找表中相应的可变量码字。这些码字通过本地连接的方式组成聚合码字，码字的字节长度也被计算出来。

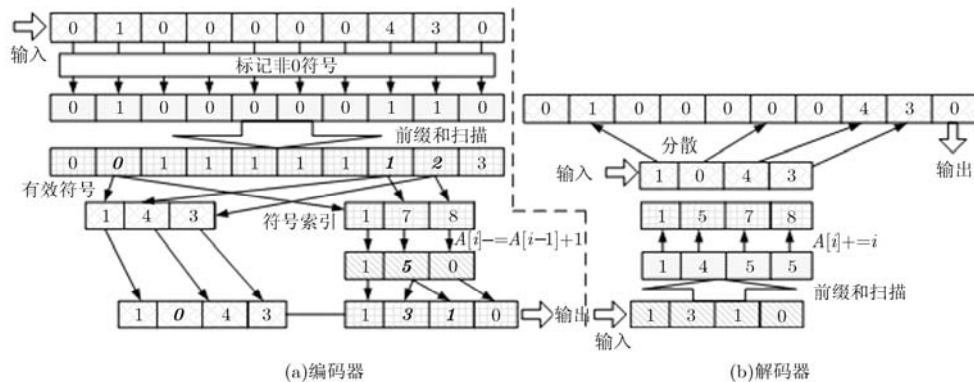


图3 基于RLC的流压缩编解码器

第2步 计算码字在数据在共享存储中的目的地址及数据起始位置。第1步中码字已经被分配给输入数据符号，码字输出位置之间的相关性已经由码字长度这个先验知识消除，可有效避免写地址冲突。每个码字的输出参数(目的地址和起始地址)由其在目的存储空间中占用的比特数及其与前一码字之间的偏移量来确定，即下标算子 $A[i]=A[i-1]$ 。并行前缀和扫描可在 $O(\lg n)$ 个并行步中以 $O(n)$ 个操作快速完成这一计算。

第3步 将码字写入压缩数据流。该步骤包含两个阶段，首先将第2步生成的码字在目的存储空间的偏移地址写入压缩数据流，然后将这些地址指示的目的存储空间对应位置设置为第1步生成的聚合码字的值。如图4(a)所示。将这两阶段结合以并发方式写入压缩数据流输出。当前GPU提供对“按位写(bitwise)”原子操作的硬件支持(put_bits_atomic)，这使得对同一地址的比特级写操作能以并发线程的形式高效执行，避免了“写竞争”的出现。在该机制的保证下，对同一存储地址的并发“读-改-写”操作序列能无中断地完成，产生正确的输出结果而不用担心写顺序。

设可编址的码字目的存储空间为32位，记为wordsize。变量 k 记为线程号，同时也关联于由第 k 号线程所处理的码字在输入流中的索引。kc记为压缩的数据流中目的存储空间的索引，startbit记为目的存储地址的起始位置。表1给出并行霍夫曼编码算法的伪码。

霍夫曼解码器如图4(b)所示。其接收编码器输出的压缩码字流和位置偏移，通过并行前缀和扫描和重排序等步骤实现数据无损解码。本文不再赘述。

5 渲染

编码压缩后的地形数据组织成多条序列流递交

表1 并行霍夫曼编码算法的伪码

```

(1)  $k \leftarrow \text{tid}$ 
(2) for threads  $k = 1$  to  $N$  in parallel
(3)    $\text{symbol} \leftarrow \text{data}[k]$ 
(4)    $\text{cw}[k], \text{crlen}[k] \leftarrow \text{cwtable}[\text{symbol}]$ 
(5) end for
(6) for threads  $k = 1$  to  $N$  in parallel
(7)    $\text{bitpos}[1, \dots, M] \leftarrow \text{prefixsum}(\text{crlen}[1, \dots, M])$ 
(8) end for
(9) for threads  $k = 1$  to  $N$  in parallel
(10)   $\text{kc} \leftarrow \text{bitpos}[k] \text{ div } \text{ws}$ 
(11)   $\text{startbit} \leftarrow \text{bitpos}[k] \text{ mod } \text{ws}$ 
(12)  while  $\text{crlen}[k] > 0$  do
(13)     $\text{numbits} \leftarrow \text{crlen}[k]$ 
(14)     $\text{cwpart} \leftarrow \text{cw}[k]$ 
(15)    if  $\text{startbit} + \text{crlen} > \text{wordsize}$  then
(16)       $\text{overflow} \leftarrow 1$ 
(17)       $\text{numbits} \leftarrow \text{wordsize} - \text{startbit}$ 
(18)       $\text{cwpart} \leftarrow \text{first numbits of } \text{cw}[k]$ 
(19)    end if
(20)     $\text{put\_bits\_atomic}(\text{out}, \text{kc}, \text{startbit}, \text{numbits}, \text{cwpart})$ 
(21)    if overflow then
(22)       $\text{kc} \leftarrow \text{kc} + 1$ 
(23)       $\text{startbit} \leftarrow (\text{startbit} + \text{numbits}) \text{ mod } \text{wordsize}$ 
(24)       $\text{remove first numbits from } \text{cw}[k]$ 
(25)       $\text{crlen}[k] \leftarrow \text{crlen}[k] - \text{numbits}$ 
(26)    end if
(27)  end while
(28) end for

```

渲染进程，渲染进程的前端是混合熵解码器，解码后的数据重建地形多分辨率四叉树。当前视口内需要渲染的地形块集合通过对四叉树深度优先遍历所得，遍历到的节点被标记为可视节点。遍历停止的条件是表示地形块的当前节点被完全剔除出当前视

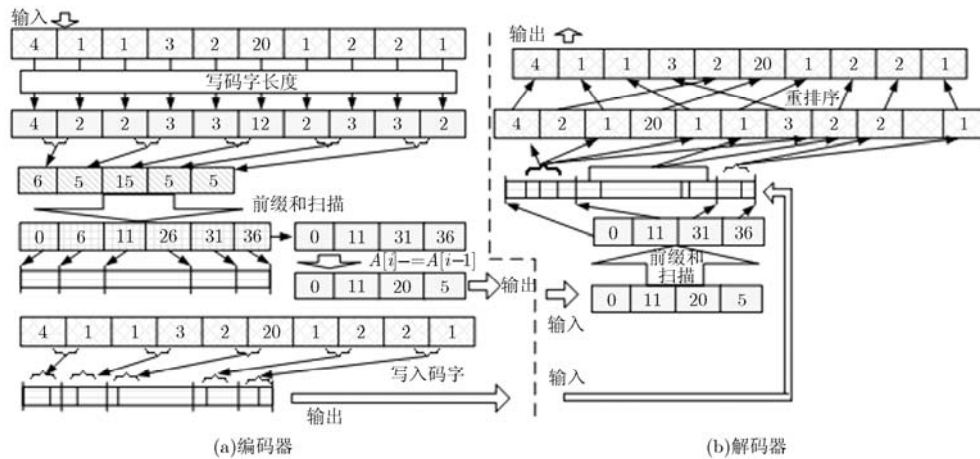


图4 Huffman编解码器

锥, 视锥剔除由视点制导的地形块精细度评价因子来决定。

结合课题组此前的工作, 本文定义视点制导的地形构网变化触发条件: 地形块精细度综合评价函数^[12]为

$$f = \frac{d \cdot v}{w \cdot C_1 \cdot \max(C_2 \cdot \varepsilon_2, 1) \cdot C_3};$$

其中 d 为地形块到相机的距离, v 为相机移动速度, w 为地形块大小, C_1 是地形块动态误差控制阈值, 用以调节地形块到相机距离对多分辨率二叉树节点屏幕贡献度的影响; C_2 是地形块起伏程度的度量阈值, C_3 为相机移动速度控制阈值, 以评价相机移动速度对多分辨率二叉树节点屏幕贡献度。比较 f 与指定的屏幕空间误差阈值 τ , 当 $f > \tau$ 时, 地形块需要精细构网, 即二叉树需要进行深度遍历。本文设定 $\tau = 0.7$ 。

对于选中的二叉树节点, 首先进行离散遍历, 遍历停止条件是获取某高程采样点的位置, 针对该高程采样点, 调入对应的正射影像块进行各向异性插值, 所得的色彩值赋予该采样点进行渲染, 这就是GPU Ray-casting方法的基本思想。

6 实验结果及分析

实验硬件平台为intel i5 750 2.67 GHz四核CPU, 4 GB系统内存, GPU为1 GB显存的Nvidia Geforce GTX285。软件平台是基于Windows XP SP3的CUDA 3.2套件和Visual studio 2008开发环境。利用CUDA实现懒惰插值的5/3小波变换、自适应方向提升小波变换和混合熵编解码器。测试数据集为空间分辨率1 m的Austria Vorarlberg正射影像, 分辨率为447000×677000, 为便于处理, 将其分为一系列2048×2048的分块子影像集合; 对应区域的格网高程数据采样点间隔10 m, 每单位为0.1 m, 数据保存格式为Tiff。实验选择压缩比(Compress Ratio, CR)、峰值信噪比(PSNR)和编解码吞吐量(百万像素/s, Mpix/s)作为本文数据处理和编解码方案的主要评价指标, 帧率作为渲染性能的评价指标。

6.1 压缩比和PSNR

为评估压缩速度和重建质量, 对一组包含100张2048×2048子影像块的Vorarlberg正射影像集运行自适应方向提升小波变换, 小波级数定为4; 对高程数据采用懒惰插值的5/3小波变换。通过指定命令参数为算法指示载入的数据是影像纹理还是高程。使用本文的混合熵编码方案压缩小波分解系数。为了对比压缩比和重建质量, 对同样的数据集, 本文使用JPEG, JPEG2000和S3TC DXT1分别对它们进行压缩。JPEG和JPEG2000的实现是基于跨平台

的ImageMagick6.7.7库¹⁾, S3TC DXT1的实现是基于libSquish1.11²⁾跨平台库。由于S3TC DXT1不支持可变比特率的图像处理, 因此为了尽可能客观地进行指标对比, 约束图像比特率(bit per pixel)固定为1.30 bpp。

对Vorarlberg的正射影像集合进行测试, 本文方法达到了18.4:1的压缩比, 重建图像的峰值信噪比达到PSNR 37.1 dB; 对于规则格网高程数据的测试结果显示, 压缩比和信噪比更优。基于离散小波变换和EBCOT编码的JPEG2000在压缩比和峰值信噪比两项指标上与本文接近, 而JPEG和S3TC压缩方法在压缩比和重建质量方面与本文方法存在较大差距, 表2给出了每种方案的性能指标测试结果。

表2 压缩比与重建质量对比

方法	压缩比(CS)		PSNR (dB)	
	正射影像	数字高程	正射影像	数字高程
本文方法	18.4:1	26.1:1	37.1	41.7
JPEG	11.2:1	17.3:1	30.3	36.2
JPEG2000	16.1:1	22.4:1	34.6	38.1
S3TC DXT1	9.7:1	14.1:1	18.4	22.6

6.2 编解码吞吐量

编解码吞吐量是度量codec性能的另一重要评价指标。高吞吐量意味着单位时间内方法可编解码更多的数据, 对于地形模型的构建及渲染、交互的性能有重大影响。采用本文的方案对包含100张2048×2048像素的Vorarlberg正射影像集进行编解码吞吐量测试, 小波等级设定为11。全部数据编码完毕耗时1.56 s, 编码吞吐量达到260 Mpix/s, 解码时间消耗0.6 s, 吞吐量达700 Mpix/s。对该区域对应高程数据的编解码吞吐量分别为720 Mpix/s和1770 Mpix/s。

为了与JPEG和JPEG2000的编解码性能进行对比, 我们分别下载编译了libjpeg-turbo 1.2.0³⁾和Kakadu⁴⁾, 前者是高性能的JPEG编解码开源方案, 后者是声称最完整的JPEG2000编解码标准的实现。基于libjpeg-turbo 1.2.0中的jpgtest例程对影像测试集和高程数据进行编解码, 程序的色度选项设定为4:4:4, 质量为90。测试影像数据的编码吞吐量为212

1) ImageMagick6.7.7: <http://www.imagemagick.org/script/download.php>

2) libSquish1.11: <http://code.google.com/p/libsquish>

3) libjpeg-turbo 1.2.0: <http://sourceforge.net/projects/libjpeg-turbo/files/1.2.0/>

4) Kakadu: <http://www.kakadusoftware.com/>

Mpix/s, 解码达到280 Mpix/s; 高程的编解码吞吐量分别为438 Mpix/s和610 Mpix/s。需要指出的是, 对JPEG编解码吞吐量的测试还不包括构建多分辨率四叉树模型的操作, 而在对本文方法所进行的性能测试中, 这些操作是考虑的。此外, libjpeg-turbo 1.2.0和Kakadu都是基于CPU的实现方案, 并未引入GPU加速。CUJ2K^[7]是基于CUDA实现的JPEG2000编码器方案, 本文也用其对测试数据进行了实验, 结果如表3所示。但CUJ2K并未实现解码器。

表3 编解码吞吐量(Mpix/s)

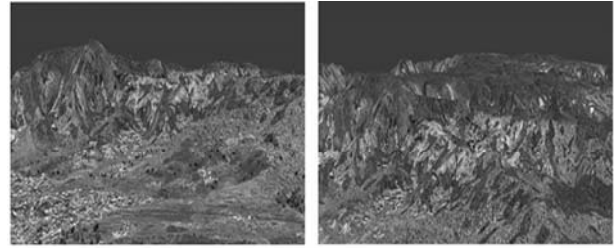
方法	编码吞吐量		解码吞吐量	
	正射影像	数字高程	正射影像	数字高程
本文方法	270	700	720	1770
JPEG(libjpeg-turbo)	212	438	280	610
JPEG2000(Kakadu)	190	345	210	580
JPEG2000(CUJ2K)	206	390	N/A	N/A

6.3 渲染帧率和效果

基于GPU的地形数据小波变换与混合熵编码, 对构建的多分辨率地形模型进行GPU Ray-casting渲染。渲染视口为1920×1080, 定义漫游视角为45°, 令视点沿预定义路径进行漫游, 图5给出了测试数据集构建的地形模型漫游截图。从图6可以看出, 本文的大规模地形数据可视化的漫游帧率主要集中在100–160 fps之间, 即使到漫游数据量最大的区域(包含3DS格式的建筑、房屋、树木、道路等), 其帧率也在50 fps以上, 远高于交互式参考漫游帧率30 fps, 确保了实时交互可视化的要求。与传统的先构建地形多分辨率金字塔模型、再结合可变粒度的LOD构网及纹理映射渲染的方案相比, 本文的方法在GPU上进行小波变换的同时即完成地形多分辨率模型构建, 数据的压缩和解压缩以流处理模式, 并支持数据的层进调度传输, 这些处理都是提高地形数据处理性能的有效技术手段。基于GPU的Ray-casting渲染避免了纹理映射所需的大量坐标和像素计算, 节省了GPU处理资源, 能保证实时交互式可视化的渲染帧率。

7 结束语

本文设计实现了一种基于提升小波变换对大规模地形数据进行多分辨率模型构建, 并使用混合熵编码进行数据压缩的地形可视化方案。阐述了小波选取原则, 给出了小波变换的地形数据多分辨率模型, 将地形数据的化简和求精操作映射为相应的小



(a)漫游时间为20 s时的地形模型渲染及光照效果 (b)漫游时间为60 s时的地形模型渲染及光照效果

图5 地形模型的渲染和光照效果随漫游时间的变化

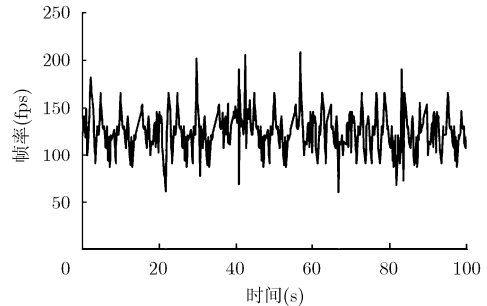


图6 帧率随漫游时间的变化

波变换和逆变换, 数据的传输调度以层进的方式进行。实现一种并行游程编码首先对量化小波系数进行编码压缩, 进一步地, 设计和实现并行可变长并行霍夫曼编码对游程编码后的数据进一步压缩。在渲染时, 以视点为引导, 对地形多分辨率模型以GPU Ray-casting的方法进行实时渲染。基于CUDA实现本文的小波变换和混合熵编码, 实验结果分析表明, 本文方法能实现高压缩比的数据压缩和高吞吐量的编解码, 有效降低了大规模地形可视化系统运行时的数据处理、传输和交互渲染的压力, 具有优良的交互实时性。

本文的小波变换和压缩编码方法仍属于经典压缩技术范畴。压缩感知^[13]理论是近年来新兴的研究热点, 它可以在不丢失逼近原信号所需信息的情况下, 用最少的观测次数来采样信号, 得到信号的压缩表示, 且不需要进行N次采样的中间阶段, 从而在节约采样和传输成本的情况下, 实现采样与压缩同步的目的。压缩感知目前广泛应用于雷达信号重建^[14]、遥感图像处理等领域。在压缩感知理论中, 信号越稀疏, 恢复信号越准确。而本文中小波变化后的数字高程和正射影像纹理系数都具有稀疏特性, 所以在后续的研究工作中, 探索引入压缩感知理论实现地形数据的压缩及重建, 值得深入研究。

参考文献

- [1] Wang X, Zheng X, and Yin Q. Large scale terrain compression and real-time rendering based on wavelet

- transform[C]. Proceedings of Computational Intelligence and Security (CIS), Suzhou, China, 2008, 2: 489-493.
- [2] Marc Treib, Florian Reichl, Stefan Auer, *et al.* Interactive editing of gigasample terrain fields[J]. *Computer Graphics Forum*, 2012, 31(2): 55-71.
- [3] Kim J K and Ra J B. A real-time terrain visualization algorithm using wavelet-based compression[J]. *The Visual Computer*, 2004, 20(2): 67-85.
- [4] Celine Roudet, Florent Dupont, and Atilla Baskurt. Semi-regular 3D mesh progressive compression and transmission based on an adaptive wavelet decomposition[C]. Proceedings of SPIE Wavelet Applications in Industrial Processing VI, San Jose, USA, 2009, 7248: 37-48.
- [5] 张燕燕, 黄其涛, 韩俊伟. 基于提升小波的大地形累进压缩及实时渲染[J]. *计算机辅助设计与图形学学报*, 2010, 33(6): 1352-1359.
- Zhang Yan-yan, Huang Qi-tao, and Han Jun-wei. Lifting wavelet-based progressive compression and real-time rendering to large terrain[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2010, 33(6): 1352-1359.
- [6] 施松新, 张引, 叶修梓, 等. 大规模地形场景流式渐进传输[J]. *浙江大学学报: 工学版*, 2008, 42(11): 1862-1867.
- Shi Song-xin, Zhang Yin, Ye Xiu-zi, *et al.* Progressive streaming of massive terrain data[J]. *Journal of Zhejiang University: Engineering Science*, 2008, 42(11): 1862-1867.
- [7] Furst N, Weiss A, Heide M, *et al.* CUJ2K library, version 1.1[OL]. <http://cuj2k.sourceforge.net>, 2012.
- [8] Balevic A. Parallel variable-length encoding on GPGPUs[C]. Proceedings of Parallel and Distributed Computing, Delft, Holland, 2009: 26-35.
- [9] Taubman D S and Marcellin M W. JPEG2000: Image Compression Fundamentals, Standards and Practice[M]. Norwell: Kluwer Academic Publishers, 2002: 262-293.
- [10] Ding Wen-peng, Wu Feng, Wu Xiao-lin, *et al.* Adaptive directional lifting-based wavelet transform for image coding[J]. *IEEE Transactions on Image Processing*, 2007, 16(2): 416-428.
- [11] Sengupta S, Harris M, Garland M, *et al.* Scientific Computing with Multicore and Accelerators[M]. Florida: USA, CRC Press Inc, 2011: 413-442.
- [12] Guo Hao-ran, Pang Jian-min, Yu Jin-tao, *et al.* GPU-oriented hybrid quadtree based multiresolution policy for large-scale terrain visualization[C]. Proceedings of the 2011 International Conference on Mechanical Engineering and Technology, London, UK, 2011: 143-146.
- [13] 石光明, 刘丹华. 压缩感知理论及其研究进展[J]. *电子学报*, 2009, 37(5): 1070-1081.
- Shi Guang-ming and Liu Dan-hua. Advances in theory and application of compressed sensing[J]. *Acta Electronica Sinica*, 2009, 37(5): 1070-1081.
- [14] 王伟伟, 廖桂生, 吴孙勇, 等. 基于小波稀疏表示的压缩感知 SAR 成像算法研究[J]. *电子与信息学报*, 2011, 33(6): 1140-1146.
- Wang Wei-wei, Liao Gui-sheng, Wu Sun-yong, *et al.* A compressive sensing imaging approach based on wavelet sparse representation[J]. *Journal of Electronics & Information Technology*, 2011, 33(6): 1140-1146.
- 郭浩然: 男, 1987年生, 博士生, 研究方向为虚拟现实、战场训练模拟、高性能计算与并行处理.
- 庞建民: 男, 1964年生, 博士, 教授, 博士生导师, 研究方向为虚拟现实、高性能计算与并行处理、计算机体系结构.