Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Terrain-based Adaptive Compression and Real-time Visualization

Xiaodong Wang
Image Processing and Pattern Recognition Laboratory
Beijing Normal University
Beijing, China
xd_wang@mail.bnu.edu.cn

Xin Zheng
Image Processing and Pattern Recognition Laboratory
Beijing Normal University
Beijing, China
zhengxin@bnu.edu.cn(corresponding author)

Figure 1. Image decomposition (yellow arrow) and reconstruction (blue arrow) using WT.

***Abstract*—We propose a large scale terrain adaptive compression method with high compression rate in this paper by using integer wavelet transform combined with restricted quadtree triangulation based on the terrain variation characteristic. By taking advantage of the efficient computing and multi-resolution attributes of our algorithm, we can decompress the wavelet coefficients which indicate the terrain variation with multi-resolution efficiently at any viewpoint and render the terrain in real time by fast updating strategy. Simulation results demonstrate that by our method we can compress the data efficiently and render the terrain scene smoothly in real time.**

***Keywords*—integer wavelet transform, restricted quadtree triangulation, terrain compression, real-time visualization**

I. INTRODUCTION

Terrain rendering is an important component of virtual reality. As virtual reality techniques have been used further more and the resolution of devices for data acquiring has been much larger, the number of data to be rendered in virtual environment has enlarged more and more. It exceeds the loading ability of the RAM in PC or graphics cards. We have to piece the large data into smaller patches and load one patch for a time. The rendering efficiency is much influenced by the process of Data IO.

In order to debase the frequency of Data IO process, many people propose their own data compression methods, such as cache-oblivious mesh layouts [1], out-of-core compression [2,3], streaming meshes [4], C-BDAM [5], geometry clipmap [6] and so on.

In recent years, wavelet transform [7] has obtained great success in image compression such as JEPG2000 etc., because of its good attributes which have more advantages than traditional compressing methods. Especially, integer wavelet transform (IWT), which is based on second-generation wavelets, lift scheme [8,9], has the advantages of simplicity and low computational complexity. So our previous work [14] use this method to construct a lossless terrain compression and visualization algorithm. But in that paper, we don't consider the terrain variations especially in some complex area and render them with the same resolution. To solve this problem, we use the restricted quadtree triangulation (RQT) scheme [12] which performs hierarchical triangulation based on a quadtree structure. So in this paper, we propose an efficient method for large scale terrain adaptive compression and rendering based on integer wavelet transform combined with restricted quadtree. Except the high compression ratio and low complexity, real-time and nice view are also in our algorithm.

The organization of this paper is as follows: In the next section, we will introduce the IWT-based RQT. Then in section 3, we will talk about the terrian compression, view-depended multi-resolution decompression and the scene-updating process in real time rendering. In the last 2 sections, we will give some results of our experiments and make a brief conclusion of this work.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

II. IWT-BASED RQT

A. Integer Wavelet Transform

The wavelet transform is a mathematical tool that can be used to describe 1D/2D signals (images) in multiple resolutions. Using it an image can be decomposed into one low-frequency and three high-frequency sub-band images through a sequence of low-pass and high-pass filters, alternated with down samplings. Because most information of original image belongs to low-frequency part, so the low-frequency part can be decomposed in the same way. And its reconstruction can be considered as the contrary process of decomposition as shown in Fig.1.

In comparison with traditional wavelet transform, IWT has more advantages except the characteristic of multi-resolution decomposition and reconstruction.

First, IWT uses lifting scheme. The lifting scheme is a technique for both designing wavelets and performing the discrete wavelet transform. And the way to start the lifting scheme is via a one step wavelet which called predict wavelet. Predict wavelets are a good model for how wavelet algorithms can approximate a set of data with a function like polynomial interpolation. It starts with a split step, which divides the data set into odd and even elements. Then the predict step, where the odd value is predicted from the even value. Finally the update step replaces the even elements with an average. A simple lifting scheme forward transform is diagrammed in Figure 2.

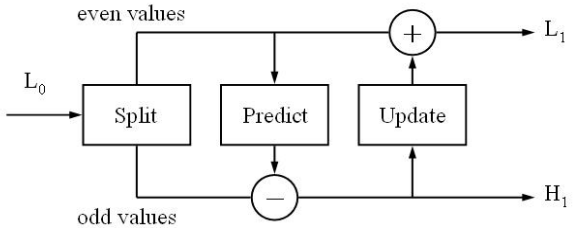

Figure 2. Lifting scheme forward WT.

Second, every transform by the lifting scheme can be inverted. That means the data can be reconstructed without any error. The 5/3 [10] integer wavelet used for our compression can explain this attribute. The 5/3 lifting scheme wavelet transform algorithm is

$$\chi(2\nu + 1) = \xi(2\nu + 1) - [\xi(2\nu) + \xi(2\nu + 2)] / 2, \quad (1)$$

$$\delta(2\nu) = \xi(2\nu) + [\chi(2\nu - 1) + \chi(2\nu + 1) + 2] / 4. \quad (2)$$

The contrary transform algorithm is

$$\xi(2\nu) = \delta(2\nu) - [\chi(2\nu - 1) + \chi(2\nu + 1) + 2] / 4, \quad (3)$$

$$\xi(2\nu + 1) = \chi(2\nu) + [\xi(2\nu) + \xi(2\nu + 2)] / 2. \quad (4)$$

Now we can see that this IWT not only can be performed immediately in memory of the input data with only constant memory overhead, but also has perfect reconstruction and efficient attributes, of course the multi-resolution. So it's very suitable for our compression and rendering.

B. Restricted Quadtree Triangulation

An example of the recursive triangulation of a plain quadtree is shown in Figure 3. The difference between the height of a vertex and the average height of the two neighbors located perpendicular to its dependency graph is defined as a delta value, δ (see Fig.4). If δ is larger than a given threshold, the corresponding vertex is split. However, cracks can occur in the corresponding three-dimensional surface as shown on the right. To guarantee a matching triangulation, cracks have to be avoided. The way is to represent the restriction of the RQT as a dependency graph on the grid. Every vertex depends on two other vertices of the same or the next higher level in the quadtree hierarchy. This means that if the vertex is selected for triangulation then the related must be selected too. The dependency graph is depicted in Figure 4.

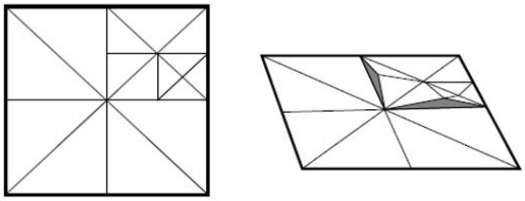

Figure 3. Nonrestricted quadtree triangulation.

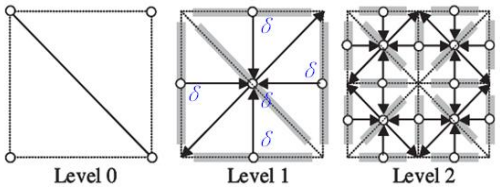

Figure 4. Dependency graph.

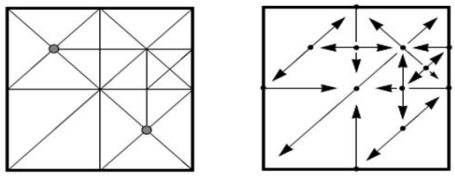

Figure 5. Restricted quadtree riangulation.

The dependency graph prevents cracks by consistently restricting the selection of points such that a matching triangulation results. Figure 5 shows the restricted version of the quadtree triangulation from Figure 3, with the dependency graph shown on the right. Note that the triangulation can efficiently be constructed because it is given implicitly.

C. IWT-based RQT

Terrain data is often stored as DEM which we can consider it as one gray image. As shown in the Fig.1, each scale is

composed of the three high-frequency subbands, and wavelet coefficients in each subband indicate the local surface variations along the direction of the corresponding arrow. Thus, a wavelet coefficient in a more complex surface region has a high magnitude and, conversely, a low magnitude in a less complex region.

As already mentioned, a wavelet coefficient has both spatial and frequency domain information. Its frequency-domain position corresponds to the scale (or resolution level), and its spatial-domain position can be estimated from the position in the corresponding subimage (or subband) of the transform domain. Figure 6 shows the 1D wavelet decomposition result (WD) of eight-point data and effective positions (EP) of wavelet coefficients obtained from the Fig.2.

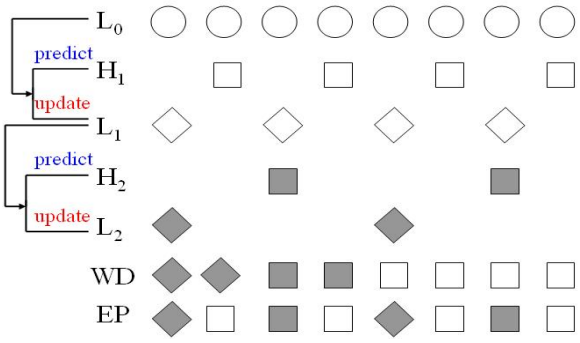

Figure 6. WD result and EP of wavelet coefficients.

Now from the (1), (2), (3) and (4), we can see that two-dimensional wavelet coefficients can be obtained by performing one-dimensional WT along the row and column directions, successively. The result of the wavelet decomposition in the 2D domain and the effective positions of wavelet coefficients obtained from Fig.1 are given in Fig. 7. Here ○, □, and △ represent coefficients of subimage HL_1, LH_1, and HH_1 for scale 1, respectively, and ◆, ●, ■, and ▲ represent coefficients of subimage LL_2, HL_2, LH_2, and HH_2, respectively, for scale 2.

After this step, each relocated coefficient has a one-to-one correspondence to a vertex in the terrain surface, and its magnitude represents the local variation of the vertex.

The square cell unit which make five coefficients centered at the location of a diagonal coefficient construct the whole structure. Notice that the coefficients of HL, LH, and HH represent the local variation of horizontal, vertical, and diagonal directions of the corresponding vertices as in Fig.8. Careful consideration of Fig.4 and Fig.8 reveals that δ and the magnitude of the relocated wavelet coefficient at the same position have similar surface complexity information.

Therefore, for triangulation, we adopt the RQT algorithm based on wavelet coefficients to utilize the efficiency of WT for data compression. In the algorithm, we first determine whether a vertex is split or merged by checking if the magnitude of the corresponding wavelet coefficient is larger than a given threshold. Then, a terrain mesh model is updated by splitting or merging the vertex using RQT.

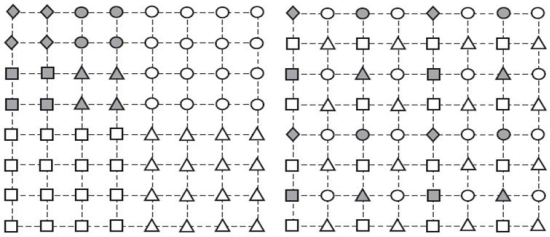

Figure 7. 2D WT result (left) and EP of wavelet coefficients (right).

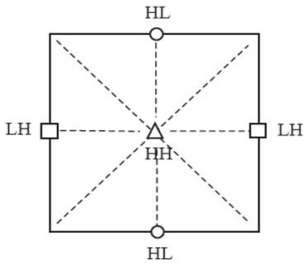

Figure 8. A square cell unit consisting of five coefficients

III. TERRAIN COMPRESSION AND VISUALIZATION

A. *Terrain Compression*

Considering the terrain data is very large, we use partitioning compression because it enable a random access to a compressed bitstream and also regulate the memory usage and the amount of transmitted data. For solving the boundary problem, our compression algorithm performs IWT n times for the entire terrain before dividing it into small blocks and then separates the whole wavelet coefficients into each block (Fig. 9). Here we store the last 2 resolution data after IWT first, because when the viewpoint is very high and we need to see the whole scene, we can render these data (have the most information of the surface) directly by contrary transform (3) and (4) mentioned in section 2. Then divide the (n-2)th transform wavelet coefficients into small blocks for compression.

In the IWT-based RQT described above, wavelet coefficients are the only data need to be stored. Many WT-based compression algorithms have been proposed in the image processing field. Here we use SPIHT (set partitioning of images into hierarchical trees) [13] to encode the wavlet coefficients for each small block after IWT. Because it has been known as an outstanding algorithm from the viewpoint of high compression ratio, so we explain it simply in our paper.

The SPIHT algorithm is composed of a sorting pass and a refinement pass. In the sorting pass, significant coefficients are selected by comparing the magnitude of each wavelet coefficient with a threshold, and locations of the significant coefficients are encoded by using a spatial orientation tree structure. In the refinement pass, bit-plane coding is performed

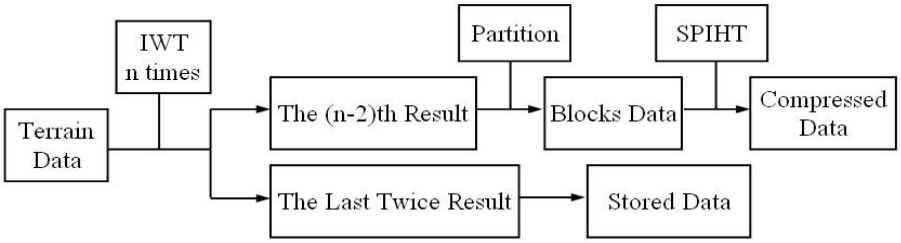

Figure 9. Flow chart of the compression algorithm.

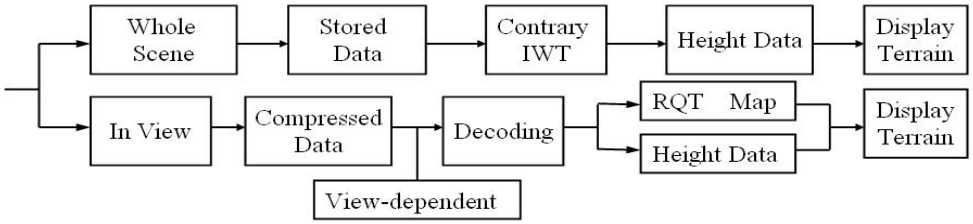

Figure 10. Flow chart of the visualization algorithm.

for the significant coefficients selected in the previous sorting passes. These two passes are progressively repeated by reducing the threshold by half. Meanwhile, the algorithm utilizes the observation, known as a decaying spectrum, that children in lower scales of an insignificant coefficient may also be insignificant with a high possibility. So it adopts a tree structure to represent insignificant coefficients over several subbands at once.

In implementing the SPIHT algorithm, the three queues of a list of insignificant sets (LIS), list of insignificant pixels (LIP), and list of significant pixels (LSP) are adopted. In the sorting pass, all trees and pixels in the LIS and LIP are compared with the current threshold. Then, new significant pixels are moved into the LSP queue and bit-plane coding is applied to them during the refinement pass.

B. Terrain Rendering

In this section, we describe a real-time terrain visualization algorithm to display large terrain data. Fig.10 shows the overall flow chart of the proposed algorithm.

If you want the whole scene, render the stored data after performed contray IWT. As the viewpoint close to the surface, the terrain grids have a finer resolution, so choose the terrain blocks which in our view and read them into memory. Then perform the decompression and contray IWT of the corresponding bitstreams we can acquire the restricted quadtree structure and height data for each visible block, so the rendering begins. When we move, the previous triangular terrain mesh is then updated by splitting or merging the vertices based on RQT.

IV. RESULTS

We implement our system by using one terrain data with the size of 130175×130175 of Shenzhen city in China and the program is written with VC 6.0 and OpenGL. Experiment is taken on a Pentium D 3.0GHz machine with 1024MB of RAM.

Firstly we perform the IWT 10 times for the terrain, then store the LL_{10} ($2^7\times2^7$) and LL_9 ($2^8\times2^8$). So the stored data is 80KB. Then partition the 16GB wavelet coefficients into 127× 127 pieces with the size of 1025×1025, for each block we use SPIHT to finish the compression. As error measures for the figures, we use A error over all blocks, which is defined as follows:

$$A = (\varepsilon_0 + \varepsilon_1 + \varepsilon_2 + \cdots + \varepsilon_N) / N, \qquad (5)$$

where N is the total number of pixels and ε_N is an error of nth pixel. Table 1 shows the time for each step costs.

TABLE I. TIMES COSTS IN COMPRESSION

IWT	SPIHT	Total
4h26m	22m	4h48m

After compression the data size is 286MB, nearly 60 times smaller than the original data. The comparisons with C-BDAM [5] and geometry clipmap [6] are shown in Table 2. Then we start our rendering using the algorithm in Fig.10.

Notice that althrough [6] is better than our method in compression time and raito, but it uses regular meshes. Table 3 shows the rendering comparisons between the proposed method in this paper with [6] and our previous work [14] by flying through the terrain about 3 mins.

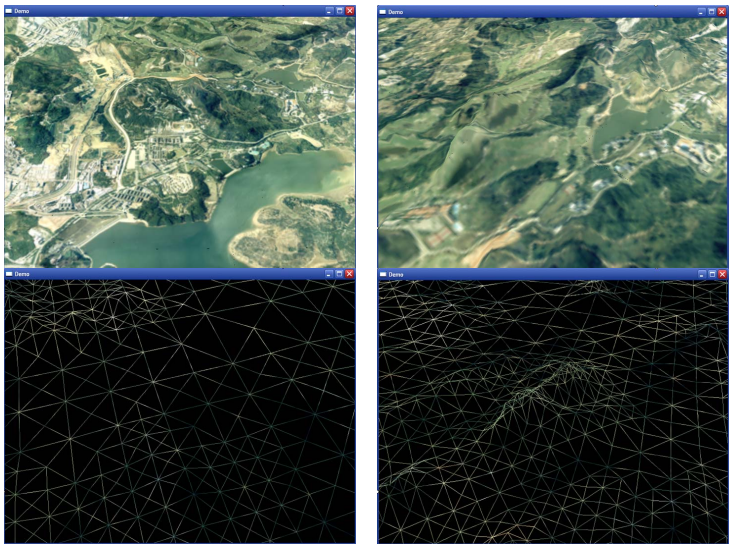

Figure 11. Terrain scenes and meshes.

TABLE II. COMPARISONS WITH THE OTHER METHODS

Method	Compression Time	Compression Ratio
C-BDAM	6h34m	39:1
Geometry Clipmap	2h30m	67:1
Our Method	4h48m	60:1

TABLE III. COMPARISONS WITH OUR PREVIOUS WORK

Method	Average Frames/sec	Average Triangles/sec	Average Required Data Rate kbits/sec
Geometry Clipmap	60	35000	216
Previous Work	75	24000	189
Proposed Method	88	42000	63

So from these results we can say that our algorithm is efficient and real-time with nice look.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose an efficient method for terrain compression and real-time rendering based on integer wavelet transform. In the future, we want to make some more improvements for the large terrain compression and real-time rendering through wavelet technique.

ACKNOWLEDGMENT

This paper is supported by grants from the National Natural Science Foundation of China (Project No.60703070).

REFERENCES

[1] S. E. Yoon, P. Lindstrom, V. Pascucci, and D. Manocha, “Cache-oblivious mesh layouts,” ACM SIGGRAPH, pp. 886-893, 2005.

[2] P. Lindstrom, “Out-of-core simplification of large polygonal models,” In SIGGRAPH’00 Conference Proceedings, pp. 259-262, 2000.

[3] P. Lindstrom, “Out-of-core construction and visualization of multiresolution surfaces,” ACM Symp on Interactive 3D Graphics, pp. 93-102, 2003.

[4] M. Isenburg, and P. Lindstrom, “Streaming meshes,” In Visualization ’05 Proceedings, pp. 231-238, 2005.

[5] E. Gobbetti, F. Marton, and F. Ganovelli, “CBDAM—Compressed batched dynamic adaptive meshes for terrain rendering,” Computer Graphics Forum, pp. 333-342, 2006.

[6] F. Losasso, and H. Hoppe, “Geometry clipmaps: terrain rendering using nested regular grids,” ACM Trans. Graph 23, pp. 769-776, 2004.

[7] S. Mallat, “A Theory for multi-resolution signal decomposition: The wavelet representation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, pp. 674-693, 1989.

[8] W. Sweldens, “The lifting scheme: A custom-design construction of biorthogonal wavelets,” Technical Report 1994:7, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina.

[9] W. Sweldens, “The lifting scheme: A construction of second generation wavelets,” Technical Report 1995:6, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina.

[10] M, D, Adams, and F. Kossentini, “Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis,” IEEE Transaction on Image Processing, pp. 1010-1024, 2000.

[11] H. Hoppe, “Smooth View-Dependent Level-of-Detail Control and its Application to Terrain Rendering,” IEEE Visualization, pp. 35-42, 1998.

[12] R. Pajarola, “Large scale terrain visualization using the restricted quadtree triangulation,” In: Proceedings of Visualization ’98, pp. 19-26, October 1998.

[13] A. Said, W. A. Pearlman, “A new fast and efficient image codec based upon set partitioning in hierarchical trees,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, pp. 243-250, June 1996.

[14] Xiaodong Wang, Xin Zheng, Qian Yin, "Large Scale Terrain Compression and Real-Time Rendering Based on Wavelet Transform," 2008 International Conference on Computational Intelligence and Security, pp. 489-493, December 2008.