STRIPE-BASED SPIHT LOSSY COMPRESSION OF VOLUMETRIC MEDICAL IMAGES FOR LOW MEMORY USAGE AND UNIFORM RECONSTRUCTION QUALITY

Youngseop Kim and William A. Pearlman

Rensselaer Polytechnic Institute
Electrical, Computer and Systems Engineering Dept.
Troy, NY 12180, USA
kimys@cipr.rpi.edu, pearlman@ecse.rpi.edu

ABSTRACT

This paper presents a low memory implementation of efficient lossy volumetric medical image compression using the Set Partitioning in Hierarchical Trees (SPIHT) algorithm. The coding units in this three-dimensional wavelet transform and compression method are short sequences of horizontal stripes cut from the sequence of slices in the volumetric image. As the compression degree increases, the boundaries between adjacent coding units became increasingly visible. Unlike video, where image frames are viewed under dynamic conditions, the volume image is examined under static conditions and must not exhibit such boundary artifacts. In order to eliminate them, we utilize overlapping and averaging both at the intra-slice and inter-slice boundaries between adjacent and successive stripe sequences. Our low memory implementation smooths out considerably the variation in mean squared error among different slices and suffers only an insignificant loss in performance from that of a full memory implementation.

1. INTRODUCTION

Recent years have seen a geometric increase in the volume of medical image data generated in hospitals. In the average sized hospital, many tera-bytes (10^{15}) of digital image data are generated every year, almost all of which have to be kept and archived. Many of these images are volumetric, consisting of a series of sequences of slices through a given part of the body. Since successive slices change slowly in their density profile, three-dimensional (3D) compression techniques seem to offer greater potential for data reduction than two-dimensional (2D) techniques which operate on each slice individually. We and others have reported previously on efficient 3D wavelet transform compression that provides reversible reconstruction [1, 2, 3]. No less important than the storage problem is the problem of transmission for off-site consultation, diagnosis, and related applications. In such scenarios, the need for rapid diagnosis and the limitations of bandwidth of the communications medium may necessitate the use of lossy compression. This kind of compression allows much greater data reductions and hence transmission speed. Although the reconstructions are not mathematically perfect, the compression is kept to a degree that maintains diagnostic accuracy.

Some excellent results for lossy compression of medical image volumes have been achieved using wavelet transform zerotree (EZW) [2] and set partitioning in hierarchical trees (SPIHT) [3] coding techniques. However, to obtain these results, a 3D wavelet transform of the full slice sequence or a segment of 16 slices was required. The former requires very large memory, but the latter, in an effort to save memory, showed deterioration in quality at the boundaries between the 16-slice segments. In this paper, we present 3D wavelet transform compression of volume medical images by the SPIHT method that utilizes a small memory. We divide the full sequence of slices into short sequences of small horizontal stripes stacked on top of each other. Each stripe sequence is encoded in turn and sent to the output bit stream, so that the memory needed for encoding a single stripe sequence is required in the encoder. Likewise, the decoder decodes each stripe sequence in turn, flushing its memory before the bit stream of the next stripe sequence arrives. While the stripe is attractive in its simplicity, it also has a drawback of introducing boundary artifacts, as well as reducing the coding efficiency if the stripe is small. To solve the boundary problem, we overlap the adjacent intra-slice stripes and the stripes between adjoining strip sequences in the axial dimension. Each stripe sequence comprises a coding unit that receives the same quality factor in the coding algorithm, so as to minimize abrupt distortion discontinuities between adjacent coding units. We describe our low-memory coding scheme using the SPIHT algorithm and different wavelet transforms for transaxial-axial direction. We also show extensive results and comparisons of methods in coding of two medi-

THIS WORK WAS SUPPORTED IN PART BY NATIONAL SCIENCE FOUNDATION GRANT NO. NCR-9523767. THE GOVERNMENT HAS CERTAIN RIGHTS IN THIS MATERIAL.

cal volume images.

2. STRIPE TRANSFORM

First we describe the partition and encoding of a single image when partitioned into stripes. Consider the division of the image uniformly into horizontal stripes with a small number of rows and compressing each stripe independently, using the SPIHT algorithm. Typically, the stripes contain 32, 64, or 128 rows, but throughout this work we used 64 rows. Now we consider the wavelet transform of each stripe and its spatial orientation tree. The wavelet subbands of the first stripe are shown in Figure 1 inside a full image wavelet transform of two decomposition levels. Our spatial orientation tree is defined in a pyramid constructed with recursive four-subband splitting. Each node of the tree corresponds to a pixel, and is identified by the pixel coordinate. Its direct descendants (offspring) correspond to the pixels of the same spatial orientation in the next finer level of the pyramid. The tree is defined in such a way that each node has either no offspring or four offspring, which always form a group of 2x2 adjacent pixels.

A common problem arises in regard to independent lossy coding of stripes that does not occur with a full slice transform. As the rate becomes lower, there occurs an increasing degree of visibility of the discontinuity in the reconstruction at adjacent stripe boundaries. We alleviate this problem by transformation and encoding of four rows in common between adjacent stripes, as shown in Figure 2, and forming a weighted average of the reconstructed values of this overlapping area. We utilize the S+P filter [4] for the vertical direction of the stripe transform , because of its small support, perfect reversibility, and known efficacy in compression of medical images. The S+P transform is not unitary, so needs to be scaled before application of the SPIHT algorithm for lossy compression. The proper scaling is described in [4, 3].

In the horizontal transform, when the image width at any decomposition level is 16 or more, we used biorthogonal 9/7 tap filters, otherwise we used S+P. Therefore, we use a recursive hybrid filtering scheme that uses either S+P or 9/7 tap filters in the horizontal direction, depending on the image width at the given decomposition level, and an S+P filter for the vertical direction.

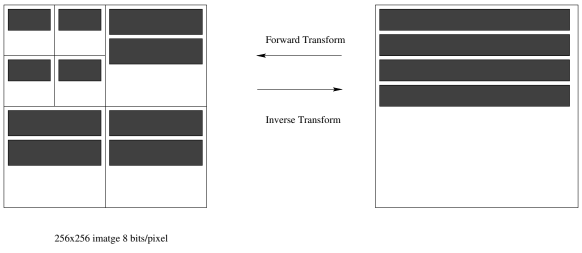

Figure 1: 2D Stripe Parent-Offspring

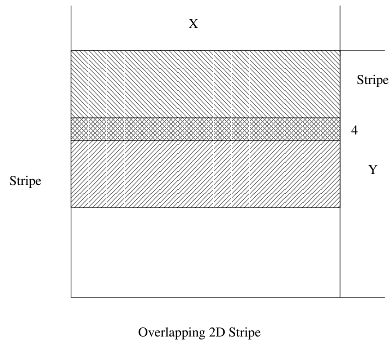

Figure 2: 2D overlap

3. CODING OF STRIPE SEQUENCES

We now present the extension of the stripe-based SPIHT to segments of N slices of a volume image. We envision now that each slice of the segment is divided into the same size and number of horizontal stripes, forming sequences of stripes stacked one upon another. To each stripe sequence is applied separable wavelet filtering in each of the three dimensions. The order of recursive splitting and subsampling of the created subbands is shown in the splitting tree diagram in Figure 3. Also shown is the final 21 subbands produced by two levels of dyadic axial splitting followed by two levels of non-dyadic intra-slice vertical and horizontal splitting. The fact that the splitting in each dimension does not follow in turn and that high frequency intra-slice subbands are further split results in a so-called wavelet packet decomposition. We found better results with packet decompositions than with the usual dyadic ones. In this work, we use three-levels of packet decomposition in both axial and transaxial domains to create 28 subbands and process 16 slice segments.

The different filtering operations are performed in both axial and transaxial domains with reflection extensions both at every image or slice boundary and at every boundary between 16-slice segments. In the horizontal transform, when the row width drops below 16 in the course of the recursive decompositions, we use an S+P filter, otherwise we use the 9/7 tap filter. In the axial direction, we use either an S+P filter or the integer I(5,3) filter. The filters used in the decompositions are summarized in Table 1

The spatio-temporal (transaxial-axial) dependence trees follow the filter tree diagram, but are hard to visualize. An illustration of three-dimensional dyadic wavelet filtering and dependence relationships is shown in Figure 4 for a segment of N = 16 slices. The image (stripe) sequence S(x,y,z) of length N is first filtered along the axial dimension and downsampled by 2, resulting in a lowpass sequence $S_l(x, y, z)$ and a highpass sequence $S_h(x, y, z)$, each of length N/2. Each slice in these sequences is then vertically and horizontally filtered and subsampled by two. As seen in Figure 4, this procedure results in eight three-dimensional subbands, denoted LLL, LLH, LHL, LHH, HLL, HLH, HHL, and HHH, where L and H denote low and high frequency in order of axial, horizontal and vertical, respectively.

In the dependence tree, each node (coefficient) has either no offspring or eight offspring which is a group of 2x2x2 adjacent coefficients. Therefore, parent-offspring relationship can be established as shown in Figure 4. The relationships for the two-level packet decomposition of 21 subbands in Figure 3 follow similar rules along the filter tree, except that when a high frequency band is further split, the eight offspring of a node in that band are dispersed among the newly generated subbands.

After packet decomposition of a stripe sequence, the SPIHT algorithm is applied to the resulting 3D dependence tree data structure [5]. Then, the output bit stream is further compressed with an arithmetic encoder. Sets of 2x2x2 coordinates are kept in the list to increase the coding efficiency and their significance values are coded as a single symbol by the arithmetic encoder.

In order to avoid artifacts caused by discontinuities in reconstructions between adjacent coding units, we overlap the stripe sequences both vertically and axially. In the vertical direction, the overlap is four rows as in Figure 2 and in the axial direction it is one slice, as shown in Figure 5. The reconstructed pixels in the overlapping regions are then treated to weighted averaging to produce the final values. In order to maintain uniform quality across all stripe sequences, we encode each one with the same quality factor setting, in effect terminating coding at the same point in the same bit plane in every coding unit. This strategy allows the rate to vary among the stripe sequences to meet the distortion constraint and often results in an average rate savings.

Our implementation requires that only a stripe sequence and its overlap area be buffered. We encode a single stripe sequence and send its compressed bit stream to the decoder. Both the encoder and decoder can flush their buffers for the next stripe sequence. When the encoder or decoder finishes with a 16 slices in a segment, it shifts to process the next segment of 16 slices.

Direction	Decomposition I	Decompostion II
Axial	S+P	I(5,3)
vertical	S+P	S+P
horizontal	9-7 tap (size $\geq$ 16), otherwise S+P	9-7 tap(size $\geq$ 16), otherwise S+P

Table 1: Filter choices in 3D stripe-based SPIHT.

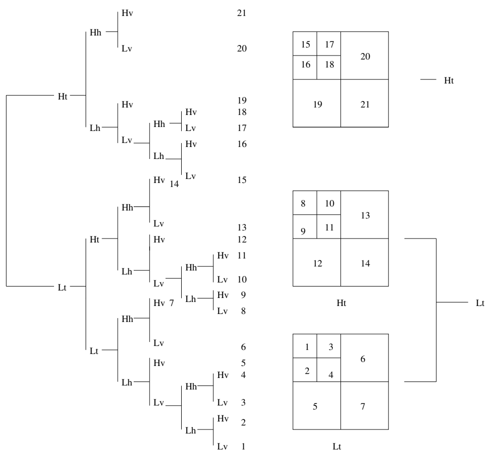

Figure 3: 3D Stripe Transaxial-Axial Decomposition

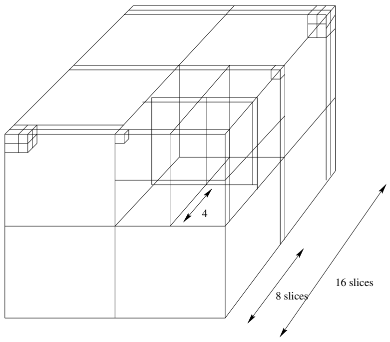

Figure 4: 3D Stripe Parent-Offspring Dependency

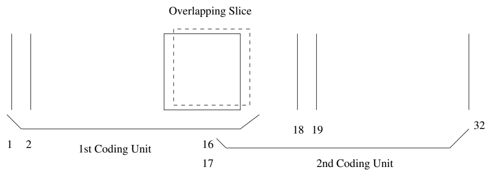

Figure 5: 3D overlapping method for axial direction

4. CODING RESULTS

We test the stripe-based 3D SPIHT coding method against original 3D and 2D SPIHT on an 8 bit per pixel 256x256x64 MR chest image and an 8 bit per pixel 256x256x128 CT skull image [6]. The graphs of PSNR (peak signal to noise ratio) versus slice number for all the coding comparisons are given in Figure 6 for the MR chest image and in Figure 7 for the CT skull image at the same code rate of 0.10 bits per pixel. In Figures 6 and 7, we can see that stripe-based 3D SPIHT no longer exhibits the PSNR dips between adjacent 16-slice segments, as does the traditional 3D SPIHT. In fact, the PSNR variation across the slices follows that of the 2D SPIHT reference, whose performance is about 6 dB inferior. We can see that the rate-distortion performance in Figure 7 is increased by an average of 0.85 dB by use of the same quality factor in every stripe sequence And even though we used smaller coding units, its efficiency is close to that of the tradional 3D SPIHT [5], being on the average only 0.61 db inferior for MR and 1.65 dB for CT, but much steadier in PSNR across the different slices.

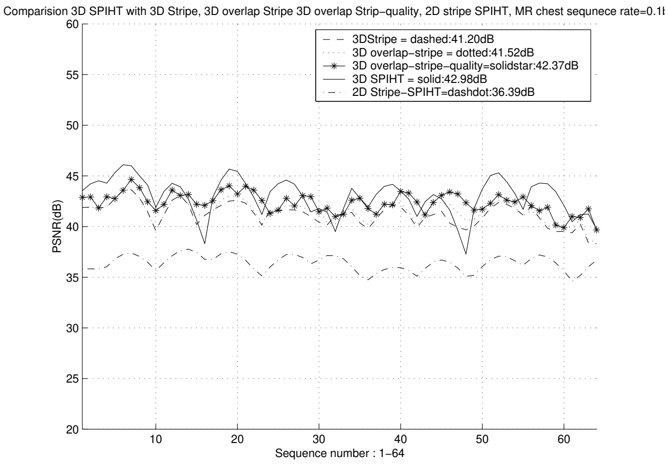

Figure 6: Comparision 3D SPIHT with 3D Stripe-SPIHT with quality factor for MR Chest image at rate = 0.1bpp

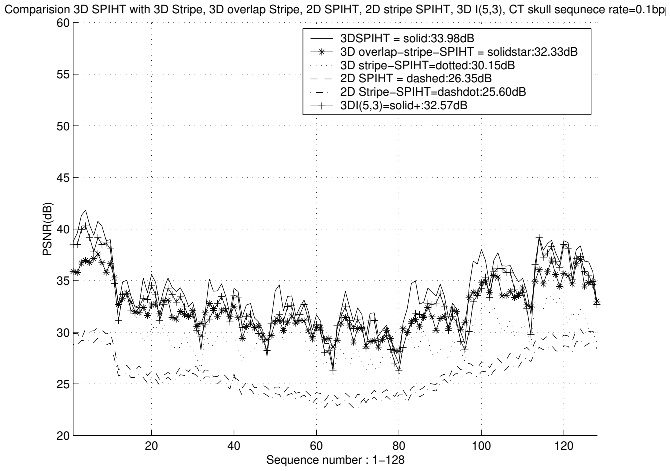

Figure 7: Comparision 3D SPIHT with 3D Overlapping-Stripe-SPIHT, 3D Stripe-SPIHT, 2D SPIHT, 2D Stripe-SPIHT size=64 for CT skull image at rate = 0.1bpp

5. SUMMARY AND CONCLUSIONS

The goal of this our work has been to provide good rate versus distortion performance using low memory and to suppress quality losses at boundaries between the coding units of stripe sequences. In this paper, we implemented a new idea of extracting segments of slices, dividng these segments into stripe sequences, taking 3D wavelet packet transform of these sequences, and encoding them with the 3D SPIHT algorithm. We were also able to achieve smooth transitions in quality between the slices both within a segment and between segments. A stripe based-SPIHT saves considerably in dynamic memory usage and does not cause noticeable degradation and may even improve rate-distortion performance when assigning the same quality factor to every stripe.

6. REFERENCES

[1] Y. Kim and W.A.Pearlman, "Lossless volumetric medical image compression," in *Proc. of SPIE, Applications of Digital Image Processing*, vol. 3808, pp. 305–312, 1999.

[2] A. Bilgin and M. W.Marcellin, "Efficient lossless coding of medical image volumes using reversible integer wavelet transforms," in *Proc. of Data Compression Conference*, pp. 428–437, March 1998.

[3] Z. Xiong, X. Wu, and D. Y.Yun, "Progressive coding of medical volumetric data using three-dimensional integer wavelet packet transform," in *IEEE Workshop on Multimedia Signal Processing*, pp. 553–558, Dec. 1998.

[4] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy image compression," in *IEEE Trans. on Image Processing*, vol. 5, pp. 1303–1310, Sept. 1996.

[5] B. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical tree," in *Proc. of Date Compression Conference*, pp. 252–260, 1997.

[6] *ftp://carlos.wustl.edu*. Mallinckrodt Institute of Radiology Image Processing Lab.