LOSSLESS VOLUMETRIC MEDICAL IMAGE COMPRESSION WITH PROGRESSIVE MULTI-PLANAR REFORMATTING USING 3-D DPCM

*Vishram Nandedkar, S. V. Bharath Kumar and Sudipta Mukhopadhyay**

Imaging Technologies Lab
General Electric - Global Research
JFWTC, Bangalore - 560066, INDIA
{*nandedkar.vishram@ge.com, bharath.sv@geind.ge.com*}

ABSTRACT

In this paper, we propose a novel lossless volumetric medical image compression scheme using three-dimensional differential pulse code modulation (3-D DPCM), which provides an efficient procedure to achieve progressive multi-planar reformatting (MPR) of large 3-D medical data sets. Being separable and commutative in the order of its application, 3-D DPCM provides an opportunity to generate MPR before decompressing the whole volume, unlike the case with other transforms. We argue that the proposed method suits radiology applications better than JPEG-2000 as it improves the diagnose capability and productivity of radiologists by providing progressive MPR, apart from providing a low complexity volumetric compression scheme with compression performance comparable to that of JPEG-2000.

1. INTRODUCTION

Several of today's imaging techniques produce three-dimensional (3-D) data sets. Medical imaging techniques, such as computed tomography (CT) and magnetic resonance (MR), generate multiple slices in a single examination, with each slice representing a different cross section of the body part being imaged. The growth in data volume directly translates to the need for compression schemes with high compression ratio so that the storage costs are kept at a minimum and speeding up of transmission across low bandwidth channels (e.g. teleradiology applications) is achieved. Simultaneously, fast encoding and decoding algorithms are desirable to reduce the waiting time of radiologists thus aiding to their productivity.

Coding of two-dimensional (2-D) images independently on a slice-by-slice basis of a 3-D volume using 2-D compression schemes do not exploit the dependencies that exist among pixel values in all three dimensions. Because the pixels are correlated in all three dimensions, the volumetric coding of the data set would provide higher compression ratio than that of 2-D coding. This correlation among pixels mainly depends on the scan resolution, overlap between adjacent frames and anatomy of the area being scanned. Hence, in our proposed method, we perform 3-D compression by considering the stack of 2-D slices as a 3-D volume, which is shown in Figure 1. This provides compression ratio comparable to that of JPEG-2000 [1]. Although, the proposed method is a 3-D compression scheme, it provides faster encoding and decoding performance than that of JPEG-2000, because of the inherent simplicity of differential pulse code modulation (DPCM).

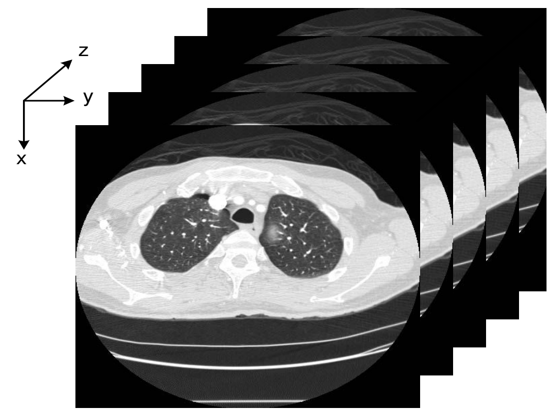

Fig. 1. Stack of 2-D CT images forming a 3-D volumetric data set.

Many a time, for the purpose of visualization and clinical diagnosis, radiologists would like to view the scan along other orientations, e.g. coronal, sagittal or any other oblique plane apart from axial, which requires multi-planar reformatting (MPR) of the original data. Typically, MPR operation requires the data to be available in raw format. But the original image data obtained from scanners is archived in compressed format, owing to storage constraints, which means that the data has to be decompressed before performing MPR. Due to the serial nature of this process, lot of waiting time is involved for radiologists, which if reduced, would definitely help in improving their productivity. The proposed scheme exactly addresses this issue by providing progressive MPR in different directions before completely decoding the compressed bit stream, thus parallelizing the process.

The paper is organized as follows. In Section 2, we present the concept of 3-D DPCM along with its encoding, decoding algorithms and some of its distinctive features. We present the progressive MPR technique, which is achieved within the framework of volumetric compression in Section 3. In Section 4, we present our proposed 3-D compression scheme and compare its performance with JPEG-2000 in terms of compression ratio. We conclude by pointing out that the proposed 3-D compression scheme suits radiology applications better than JPEG-2000 by providing comparable compression performance and faster encoding/decoding along

*The author was a former employee of GE Global Research, Bangalore-560066, India.

with progressive MPR capability, thus improving the productivity of radiologists.

2. 3-D DIFFERENTIAL PULSE CODE MODULATION

Differential pulse code modulation is a waveform coding technique, which is used to convert discrete-time analog signals to digital signals. This involves generating an error sequence, which is the difference between the actual sequence and its predicted sequence. The error sequence is quantized and then coded to obtain a digital signal. Each sample of the error sequence is computed based on the past samples of its corresponding sample in the actual sequence. The motivation to code the error sequence rather than the actual sequence is to exploit the correlation between neighboring samples. Because of correlation, the entropy of the error sequence is small compared to that of the actual sequence and hence lesser number of bits are required to code the error sequence than the actual sequence. The aforesaid philosophy of DPCM is applicable to n-D sequences as long as there exists significant correlation in the data.

In our study, we extend the same philosophy to volumetric data. Since, there exists significant correlation between the adjacent slices of volumetric data, 3-D DPCM can be used to exploit it, thus achieving low bit rates. DPCM compression method can be used for both intra-frame and inter-frame coding. Intra-frame coding exploits spatial redundancy and inter-frame coding exploits temporal or axial redundancy. In the intra-frame coding, the error sequence is generated from the samples of a given frame, while in the inter-frame coding, it is computed using the samples from neighboring frames. In both of these coding schemes, the value of target sample is predicted using the previously coded neighboring samples.

2.1. Mathematical Formulation

We provide the mathematical formulation for encoding (computation of error sequence) and decoding (reconstruction of actual sequence from error sequence) of volumetric data using 3-D DPCM. Since the encoding and decoding operations need to be of low complexity, the predicted value of a given sample is computed based on its previous sample along a particular orientation, rather than assuming an auto-regressive (AR) model of order greater than 1 and computing the prediction coefficients by solving the Wiener-Hopf equation. In such a case, 3-D DPCM becomes a separable transform and can be implemented as three 1-D DPCM operations, each along x, y and z orientations.

2.1.1. Encoder

In the 3-D DPCM method, we perform the 1-D DPCM operation along three orientations i.e. x (row-wise), y (column-wise) and z (frame-wise). As aforementioned, we consider the first order difference of a given sample with its previous sample to compute its predicted value. Let us consider a 1-D sequence $\{x(n)\}$, $x(n) \in \mathbb{Z}$, of length l. The encoding operation is given as

$$d(n) = \begin{cases} x(n) - x(n-1) & \text{for } 0 < n < l, \\ x(n) & \text{for } n = 0. \end{cases} \tag{1}$$

where $\{d(n)\}$ is the error sequence, which we term as the encoded version of $\{x(n)\}$. Extending from Eq. (1), the encoding algorithm for volumetric data, $\{x(n_1, n_2, n_3)\}$, using separable 3-D DPCM implementation is given as

$$d_x(n_1,n_2,n_3) = \begin{cases} x(n_1,n_2,n_3) - x(n_1-1,n_2,n_3) \\ \qquad \text{for } 0 < n_1 < l_x, \\ x(n_1,n_2,n_3) \ \text{for } n_1 = 0 \end{cases} \tag{2}$$

$$d_y(n_1,n_2,n_3) = \begin{cases} d_x(n_1,n_2,n_3) - d_x(n_1,n_2-1,n_3) \\ \qquad \text{for } 0 < n_2 < l_y, \\ d_x(n_1,n_2,n_3) \ \text{for } n_2 = 0 \end{cases} \tag{3}$$

$$d_z(n_1,n_2,n_3) = \begin{cases} d_y(n_1,n_2,n_3) - d_y(n_1,n_2,n_3-1) \\ \qquad \text{for } 0 < n_3 < l_z, \\ d_y(n_1,n_2,n_3) \ \text{for } n_3 = 0 \end{cases} \tag{4}$$

where $\{d_x\}$, $\{d_y\}$ and $\{d_z\}$ represent the error sequences computed along x, y and z directions respectively. l_x, l_y and l_z are the respective dimensions of $\{x(n_1, n_2, n_3)\}$ along x, y and z. Except the sample at $(0, 0, 0)$, all other samples are modified in the above encoding operation. From Eq. (4), sequence $\{d_z\}$ is the 3-D DPCM encoded version of $\{x(n_1, n_2, n_3)\}$ and can be coded to achieve low bit rates. Also, this sequence will be the input to the decoding algorithm to reconstruct $\{x(n_1, n_2, n_3)\}$ in a lossless manner.

2.1.2. Decoder

The decoding algorithm for the volumetric data involves the reconstruction of $\{x(n_1, n_2, n_3)\}$ from the encoded sequence $\{d_z\}$ in a lossless manner. Using Eq. (1), the decoding algorithm for 1-D error sequence $\{d(n)\}$ is given as

$$x(n) = \begin{cases} x(n-1) + d(n) & \text{for } 0 < n < l, \\ d(n) & \text{for } n = 0. \end{cases} \tag{5}$$

which can be extended to 3-D sequence $\{d_z\}$ as follows.

$$d_y(n_1,n_2,n_3) = \begin{cases} d_y(n_1,n_2,n_3-1) + d_z(n_1,n_2,n_3) \\ \qquad \text{for } 0 < n_3 < l_z, \\ d_z(n_1,n_2,n_3) \ \text{for } n_3 = 0 \end{cases} \tag{6}$$

$$d_x(n_1,n_2,n_3) = \begin{cases} d_x(n_1,n_2-1,n_3) + d_y(n_1,n_2,n_3) \\ \qquad \text{for } 0 < n_2 < l_y, \\ d_y(n_1,n_2,n_3) \ \text{for } n_2 = 0 \end{cases} \tag{7}$$

$$x(n_1,n_2,n_3) = \begin{cases} x(n_1-1,n_2,n_3) + d_x(n_1,n_2,n_3) \\ \qquad \text{for } 0 < n_1 < l_x, \\ d_x(n_1,n_2,n_3) \ \text{for } n_1 = 0 \end{cases} \tag{8}$$

Since $x(n) \in \mathbb{Z}$, Eq. (1) and Eq. (5) provide integer implementation with lossless reconstruction, which is essential in medical applications, as it does not introduce unknown artifacts affecting the diagnosis. Similarly, it is clear from Eq. (2)−(4) and Eq. (6)−(8) that $\{x(n_1, n_2, n_3)\}$ is perfectly reconstructed. The aforementioned encoding and decoding operations can be implemented as in-place computation, thus reducing the memory footprint. Since, 3-D DPCM is a separable and orientation independent transform, the decoding operation need not necessarily follow the same order as the encoding operation. For example, if the encoding is performed first along y, followed by x and then z, the decoder need not have to follow $z \to x \to y$ path for perfect reconstruction. Even the decoding in $x \to y \to z$ or $y \to z \to x$ would provide lossless reconstruction of the original volumetric data.

2.2. Features

2.2.1. *Integer operation & Lossless reconstruction*

Since the input volumetric data is integer valued and the encoding and decoding operations are linear, the error and the reconstructed sequences are also integer valued. This feature is critical for lossless reconstruction, which is not possible with other transforms like Fourier/DCT/wavelet because of the floating-point nature of transformed data. The floating-point data is quantized to make it integer valued so as to enable encoding, which results in lossy reconstruction. Though, integer wavelet transforms provide a framework for lossless reconstruction, they are not order independent like DPCM because of the non-linearity involved in computing the wavelet coefficients, and are also more complex than DPCM. In Section 3, we show that order independent transform is required to generate progressive MPR, which is provided by 3-D DPCM.

2.2.2. *Computational complexity*

3-D DPCM is computationally less complex and requires minimal CPU time because it involves either addition or subtraction depending on the encoding or decoding operation. Using 1-D DPCM, $\{x(n)\}$ requires $(l-1)$ operations to be performed to compute the error sequence or the reconstructed sequence. 3-D DPCM is implemented as three separable 1-D operations and it requires $C_{dpcm,3}$ operations (additions/subtractions), where

$$\begin{aligned} C_{dpcm,3} &= (l_x-1)l_yl_z + l_x(l_y-1)l_z + l_xl_y(l_z-1) \\ &= 3l_xl_yl_z - (l_xl_y + l_yl_z + l_zl_x) \end{aligned} \quad (9)$$

Since JPEG-2000 uses wavelet transform, we are interested to determine the computational complexity of 3-D integer wavelet transform and compare it with $C_{dpcm,3}$. Let the integer implementation of a given 1-D wavelet transform requires p operations, where the operations include addition, subtraction and flooring. Using 1-D wavelet transform, $\{x(n)\}$ would require $pl(1-\frac{1}{2^k})$ operations, where k is the number of levels of wavelet decomposition, with the assumption that $\frac{l}{2^k} = 2^h$, $h \in \mathbb{Z}^+$. The 3-D integer wavelet transform can be implemented as separable 1-D transforms, which requires

$$C_{w,3} = \frac{12}{7}pl_xl_yl_z\left(1-\frac{1}{8^k}\right) \quad (10)$$

where $C_{w,3}$ represents the complexity of the given wavelet, for k levels of decomposition, with the assumption that $\frac{l_x}{2^k} = 2^q$, $\frac{l_y}{2^k} = 2^r$, $\frac{l_z}{2^k} = 2^s$, where $q, r, s \in \mathbb{Z}^+$. The computational complexity of 3-D Fourier transform or 3-D DCT is $O(l_xl_yl_z\log(l_xl_yl_z))$. Consider 1-D $(1,1)$ wavelet [2] for which $p = 4$. For this value of p, it is clear from Eq. (9) and Eq. (10) that $C_{dpcm,3} < C_{w,3}$ for any l_x, l_y and l_z. JPEG-2000 uses 2-D $(2,2)$ wavelet [2] with $p = 10$, which is more complex than 2-D $(1,1)$. To apply JPEG-2000 on volumetric data, 2-D $(2,2)$ wavelet has to applied on each slice of the data independently. The computational complexity of 2-D DPCM and 2-D wavelet (with k levels of decomposition) applied independently on each slice of 3-D data are given as

$$C_{dpcm,2} = 2l_xl_yl_z - l_xl_z - l_yl_z \quad (11)$$

$$C_{w,2} = \frac{4}{3}pl_xl_yl_z\left(1-\frac{1}{4^k}\right) \quad (12)$$

Table 1 shows the complexity estimates for 2-D and 3-D versions of $(1,1)$, $(2,2)$ (with $k = 1$) and DPCM for volumetric data of dimensions $512 \times 512 \times 16$. The 2-D operations are performed independently on each of 16 slices of dimensions 512×512. Though JPEG-2000 is a 2-D compression scheme using 2-D $(2,2)$ wavelet applied independently on each slice, it is more complex than 3-D DPCM as is evident from Table 1. Again, it is interesting to note that 3-D DPCM is 26.6% *less complex* than 2-D $(1,1)$, thus suiting to our requirements.

Transform	Complexity, C	C_I
2-D $(1,1)$	16777216	$-$
2-D $(2,2)$	41943040	150%
2-D DPCM	8372224	-50.1%
3-D $(1,1)$	25165824	50%
3-D $(2,2)$	62914560	275%
3-D DPCM	12304384	-26.6%

Table 1. Comparison of computational complexities of 2-D and 3-D versions of $(1,1)$, $(2,2)$ wavelets with one level of decomposition and DPCM on volumetric data of $512 \times 512 \times 16$. C_I represents the percentage increase in complexity of a given transform w.r.t. 2-D $(1,1)$. Refer to Eq. (9)$-$(12).

2.2.3. *Memory footprint*

The unique flow of encoder and decoder of 3-D DPCM can be implemented as in-place computation, thus requiring only one buffer (for original data) to perform these operations, leading to less memory footprint, unlike the case in other aforesaid transforms.

2.2.4. *Non-dyadic operation*

Dyadic operation is applicable only to those data whose dimensions are integer power of 2. Since most wavelet implementations are inherently dyadic, it requires preprocessing of non-dyadic data. Generally, Ultrasound (US) and Computed Radiography (CR) images are non-dyadic. The 3-D DPCM operation is independent of data dimension and hence does not require any preprocessing of data, unlike wavelets.

3. PROGRESSIVE MULTI-PLANAR REFORMATTING USING 3-D DPCM

As aforementioned, many of today's medical imaging modalities produce volumetric data sets, which are very large to be transferred over low bandwidth networks to remote physicians. Hence, there is a need for compression schemes with high compression ratio so that the waiting time of radiologists is reduced along with the storage costs involved in archiving the data. Within this compression framework, any additional feature, which does not increase the waiting time of radiologists would help in increasing their productivity and diagnostic capability. Recently, a smart navigation scheme has been proposed in the compression framework to navigate through large 3-D data sets [3].

In this paper, we propose a method to generate progressive MPR of volumetric data within the compression framework using 3-D DPCM. The conventional method to compute MPR in compression framework is shown in Figure 2a, which involves decoding the bit stream to generate full data followed by MPR computation. Being serial, this process increases radiologists' waiting time. The stall in the process can be removed if MPR can be generated while decoding the bit stream itself. This necessitates the

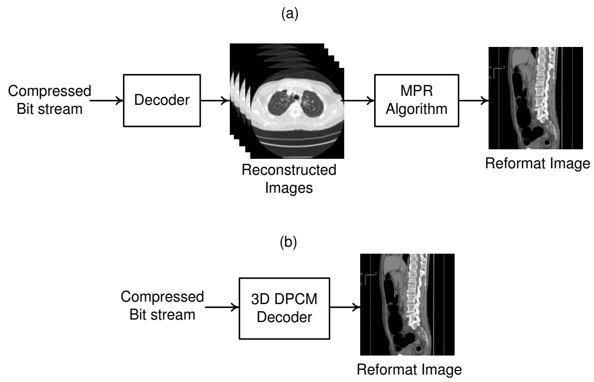

Fig. 2. (a) Conventional method to compute MPR in compression framework (b) Progressive MPR using 3-D DPCM.

order of application of 1-D transform along different directions to be commutative. 3-D versions of Fourier/DCT/wavelet transform are commutative if implemented as separable 1-D transforms in floating point format. But, this does not provide lossless reconstruction because of quantization errors. Though, integer wavelets provide lossless compression, the order of its application is not commutative. Hence, no codec offers progressive MPR before decoding the full bit-stream.

Since, 3-D DPCM is commutative and involves integer operations, it can be used to generate MPR progressively from the compressed bitstream. Because of commutative nature, the order of encoding and decoding operations need not be the same, nevertheless, lossless reconstruction is guaranteed. Hence, different order of decoding would provide different reformats of the original data volume. Though the original data volume shown in Figure 1 is the axial view, different order of decoding provides the sagittal view which is shown in Figure 2b. For example, assume that a radiologist requests for the sagittal view of the scanned series, which is available as compressed bit stream in the server. In the decoder, we perform the inverse 1-D DPCM operation first along z (axial) direction and then along x (coronal) direction, which provides the first reformatted frame along y (sagittal) direction. Consecutive frames can then be decoded by performing inverse 1-D DPCM along y direction. Thus the first reformatted frame is obtained before decoding that direction, thus delivering the radiologist with an image reformat faster than the conventional method.

4. LOSSLESS VOLUMETRIC COMPRESSION USING 3-D DPCM

Having discussed the need for 3-D compression and progressive MPR feature provided by 3-D DPCM in compression framework, we present the experimental results of compression ratio provided by our proposed volumetric compression scheme. The study is conducted on 3-D CT data sets with each 2-D slice being 512×512. To avoid memory footprint, the data set is divided into sub-volumes, each with 8 slices. As aforesaid, 3-D DPCM is implemented as three separable 1-D DPCM operations, each along x, y and z directions and the error sequence is computed. Huffman coding is used to code the error sequence and the bit-stream is generated. The compression ratio performance of our proposed volumetric compression scheme is compared to that of JPEG-2000 in Table 2. The performance comparison is made with respect to JPEG-2000 because it is the new state-of-the-art image compression standard designed for broad range of applications, including compression and transmission of medical images. Table 2 shows that the proposed lossless volumetric compression scheme performs similar or better than JPEG-2000 in terms of compression ratio. JPEG-2000 uses adaptive arithmetic coding which is more complex than the Huffman coding used in our proposed scheme. It is to be noted that better compression ratios can be achieved with our proposed method if more complex entropy coding schemes are adopted as in JPEG-2000, but with a trade-off in decoding speed. Compression schemes with fast decoder provide minimal waiting time for the radiologists when the data transferred over the network is progressively decoded, thus improving their productivity. Since radiology applications prefer fast compression schemes with good compression performance, we argue that the proposed method suits these applications better than JPEG-2000.

Data	N_f	S_t	S_o	CR_{3D}	CR_J
CT-1	232	5.000	75%	3.68	3.61
CT-2	384	2.500	50%	4.98	3.44
CT-3	1510	1.250	28%	3.74	3.86
CT-4	1920	1.250	36%	3.83	3.90
CT-5	2144	0.625	50%	3.13	3.11

Table 2. Compression ratio performance of proposed 3-D compression scheme w.r.t. JPEG-2000. N_f represents number of slices in the data volume, S_t and S_o represent the slice thickness (in mm) and slice overlap respectively. CR_{3D} and CR_J represent the compression ratio achieved by the proposed volumetric compression scheme using 3-D DPCM and JPEG-2000 respectively.

5. DISCUSSION & CONCLUSION

We have proposed a 3-D DPCM based lossless volumetric compression scheme, which provides progressive MPR for large 3-D medical data sets. The conventional compression schemes lack the ability to achieve lossless compression along with MPR and fast decoding speed, unlike our proposed method. But, the proposed method does not provide multi-resolution capability, which is inherent in wavelet based compression schemes. The proposed method is intended to improve the productivity and diagnostic capability of radiologists and suits radiology applications better than JPEG-2000 by providing comparable compression ratio and faster decoding along with progressive MPR. Though the results of the proposed method are provided only on CT data sets, the method is also applicable to 3-D medical data sets of other modalities.

6. REFERENCES

[1] D. Taubman and M. Marcellin, *JPEG-2000: Image Compression Fundamentals, Standards and Practice*, Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Boston, November 2001.

[2] A. R. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo, "Lossless image compression using integer to integer wavelet transforms," in *Proc. IEEE ICIP*, October 1997, pp. 596–599.

[3] S. V. Bharath Kumar, Sudipta Mukhopadhyay, and Vishram Nandedkar, "A Novel Progressive Thick Slab Paradigm for Volumetric Medical Image Compression and Navigation," in *Proc. IEEE ICIP*, Singapore, October 2004, To Appear.