Advances in medical image compression: Novel schemes for highly efficient storage, transmission and on demand scalable access for 3D and 4D medical imaging data

by

Victor F. Sanchez Silva

B.S., Instituto Tecnológico y de Estudios Superiores de Monterrey, 1999
M.Sc., University of Alberta, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIRMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

June 2010

Abstract

Three dimensional (3D) and four dimensional (4D) medical images are increasingly being used in many clinical and research applications. Due to their huge file size, 3D and 4D medical images pose heavy demands on storage and archiving resources. Lossless compression methods usually facilitate the access and reduce the storage burden of such data, while avoiding any loss of valuable clinical data. In this thesis, we propose novel methods for highly efficient storage and scalable access of 3D and 4D medical imaging data that outperform the state-of the-art. Specifically, we propose (1) a symmetry-based technique for scalable lossless compression of 3D medical images; (2) a 3D scalable medical image compression method with optimized volume of interest (VOI) coding; (3) a motion-compensation-based technique for lossless compression of 4D medical images; and (4) a lossless functional magnetic resonance imaging (fMRI) compression method based on motion compensation and customized entropy coding.

The proposed symmetry-based technique for scalable lossless compression of 3D medical images employs wavelet transform technology and a prediction method to reduce the energy of the wavelet sub-bands based on a set of axes of symmetry. We achieve VOI coding by employing an optimization technique that maximizes reconstruction quality of a VOI at any bit-rate, while incorporating partial background information and allowing for gradual increase in peripheral quality around the VOI.

The proposed lossless compression method for 4D medical imaging data employs motion compensation and estimation to exploit the spatial and temporal correlations of 4D medical images. Similarly, the proposed fMRI lossless compression method employs a motion compensation process that uses a 4D search, bi-directional prediction and variable-size block matching for motion estimation; and a new context-based adaptive binary arithmetic coder to compress the residual and motion vector data generated by the motion compensation process.

We demonstrate that the proposed methods achieve a superior compression performance compared to the state-of-the-art, including JPEG2000 and 3D-JPEG2000.

Table of Contents

List of Tables

List of Figures

Acronyms

2D	Two Dimensional
3D	Three Dimensional
4D	Four Dimensional
AVC	Advanced Video Coding
BOLD	Blood Oxygenation Level-dependent
bpp	bits per pixel
bpv	bits per voxel
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-adaptive Variable-length Coding
CT	Computed Tomography
DCMV	Differential Coding of Motion Vectors
DCT	Discrete Cosine Transform
DICOM	Digital Imaging and Communications in Medicine
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimized Truncation
ESCOT	Embedded Sub-band Coding with Optimized Truncation
EZW	Embedded Zerotree Wavelet
fMRI	Funtional Magnetic Resonance Imaging
GOP	Group of Picures
GOS	Group of Slices
IEEE	Institute of Electrical and Electronics Engineers
IWT	Integer Wavelet Transform
JPEG	Joint Photographic Experts Group
JPIP	JPEG2000 Interactive Protocol
LPS	Least Probable Symbol

MAXSHIFT	Maximum Shift
MB	Mega Byte
MF-MC	Multi-frame Motion Compensation
MPEG	Motion Picture Experts Group
MPS	Most Probable Symbol
MR	Magnitude Refinement
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
MV	Motion vector
NLM	National Library of Medicine
PACS	Picture Archiving and Communication Systems
PET	Positron Emission Tomography
PSNR	Peak Signal-to-Noise Ratio
RLC	Run-length Coding
ROI	Region of Interest
SAD	Sum-of-absolute Differences
SBHP	Sub-band Block Hierarchical Partitioning
SC	Sign Coding
SMVD	Spatial Motion Vector Difference
SNR	Signal to Noise Ratio
SPECT	Single Photon Emission Computed Tomography
SPIHT	Set Partitioning in Hierarchical Trees
SQP	Square Partitioning
TMVD	Temporal Motion Vector Difference
TR	Time of Repetition
US	Ultra Sound
VLC	Variable-length Coder
VOI	Volume of Interest
VSBM	Variable-size Block Matching
ZC	Zero Coding

Acknowledgements

I would like to thank my supervisors, Dr. Panos Nasiopoulos and Dr. Rafeef Abugharbieh, for their valuable advice, support, time and patience throughout my degree.

I would also like to thank my colleagues at The Signal and Image Processing Laboratory and The Biomedical Signal and Image Computing Laboratory. Thank you for the discussions and good vibe around the lab.

To Dr. Boris Sobolev, thank you for your support, valuable discussions and advice on how to perform research. Working together has helped me develop valuable skills not only for the academic world, but also for life in general.

To my parents, whose encouragement and support helped me throughout my degree. To my siblings, Luis and Rebeca, thank you for always staying positive and supportive.

Co-authorship Statement

This thesis presents research work conducted by Victor Sanchez, in collaboration with Dr. Panos Nasiopoulos and Dr. Rafeef Abugharbieh.

This is a Manuscript based thesis constructed around the four manuscripts described below:

Chapter 2: "Symmetry–Based Scalable Lossless Compression of 3D Medical Image Data," *IEEE Transactions on Medical Imaging*, vol. 28, no. 7, pp. 1062–1072, 2009. The identification and design of the research program, the research and data analysis were performed by Victor Sanchez. The manuscript is the work of Victor Sanchez who received suggestions and feedback from Dr. Nasiopoulos and Dr. Abugharbieh.

Chapter 3: "3D Scalable Medical Image Compression with Optimized Volume of Interest Coding," submitted to the *IEEE Transactions on Medical Imaging*, August 2009. The identification and design of the research program, the research and data analysis were performed by Victor Sanchez. The manuscript is the work of Victor Sanchez who received suggestions and feedback from Dr. Nasiopoulos and Dr. Abugharbieh.

Chapter 4: "Efficient Lossless Compression of 4D Medical Images Based on the Advanced Video Coding Scheme," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 4, pp. 442–446, 2008. The identification and design of the research program was performed by Victor Sanchez and Dr. Nasiopoulos. The research and data analysis were performed by Victor Sanchez. The manuscript is the work of Victor Sanchez who received suggestions and feedback from Dr. Nasiopoulos and Dr. Abugharbieh.

Chapter 5: "Novel Lossless fMRI Image Compression Based on Motion Compensation and Customized Entropy Coding," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 645–655, 2009. The identification and design of the research program, the research and data analysis were performed by Victor Sanchez. The manuscript is the work of Victor Sanchez who received suggestions and feedback from Dr. Nasiopoulos and Dr. Abugharbieh.

Chapter 1

1. Introduction and Overview

1.1 Introduction

Recent years have seen three dimensional (3D) and four dimensional (4D) medical image acquisitions becoming a staple in healthcare practice and research [1]. These types of images are increasingly being collected and used in many clinical and research applications including functional brain imaging and computer assisted intervention. The availability of such exquisite non-invasive *in vivo* high resolution data have thus practically revolutionized medicine with 3D and 4D medical images now being an integral part of patients' records.

Current 3D medical imaging technologies include many *structural* imaging modalities such as magnetic resonance imaging (MRI), computed tomography (CT), positron emission tomography (PET), 3D ultrasound (US), and single photon emission computed tomography (SPECT) [2]. Volumetric medical images typically comprise cross sections of the anatomy of a region of interest (ROI) as a collection of two dimensional (2D) image slices.

Current dynamic volumetric, or 4D imaging technologies acquisitions, include *structural* imaging modalities, such as dynamic MRI, dynamic CT, and dynamic (3D-US); and *functional* imaging modalities, such as dynamic PET, dynamic SPECT, and functional MRI (fMRI) [2]. Four dimensional medical images typically represent sequences of volumetric data temporally collected through 3D imaging of a dynamically changing ROI. A 4D ROI usually depicts an anatomical area of interest, which can be a beating heart in a dynamic CT scan, or a functional activation pattern, such as the blood oxygenation level-dependent (BOLD) signal in fMRI data.

The continuously improving visual quality of such 3D and 4D data due to gains in spatial and temporal resolutions poses a big burden on computational resources needed to store, access and transmit massive amounts of data for clinical use and research studies [3]. For example, a typical fMRI scanning session involves temporal acquisition of the 3D volume of the brain every 1-2 seconds and always involves a group of subjects. With at least 8-10 subjects typically scanned, with 200+ megabytes of data per subject, an fMRI study typically results in several gigabytes of imaging data. Lee et al. reported in [4] that the University of Washington Medical Centre, a medium-sized hospital performs approximately 80,000 MRI studies per year. At 30 MB per study, the amount of digital images generated using MRI technology is 2.4 Tera bytes of data per year.

In current practice, picture archiving and communication systems (PACS), which contain a collection of specialized networks and computational infrastructure, are commonly used for storage, retrieval,

distribution, and visualization of medical images [5-7]. This requires that the underlying image data be efficiently stored, accessed and transmitted over networks of various bandwidth capacities. Moreover, a recent trend towards facilitating the general public online access to their own medical records has also become of significant interest to many major companies and healthcare institutions [8]. Hence, the design of cost effective and accurate compression techniques for storing, transmitting and accessing 3D and 4D medical images has become a significant challenge.

Image compression techniques can be broadly classified into two large groups, lossy compression techniques and lossless compression techniques [9]. Lossy compression techniques are able to achieve a good compression performance by permanently removing some information from the original data. On the other hand, lossless compression retains all the original information in the data by sacrificing compression performance. Lossless compression is usually the standard in medical imaging to reduce the storage and transmission burden of such data, while at the same time avoiding any loss of valuable clinical data which may result in serious clinical and legal implications.

In the case of structural medical images (3D or 4D), the most desirable properties of any lossless compression method include: 1) high compression ratios; 2) resolution scalability, which refers to the ability to decode the compressed image data at various resolutions; 3) quality scalability, which refers to the ability to decode the compressed image at various qualities or signal-to-noise ratios (SNR) up to lossless reconstruction; and 4) ROI coding, which refers to the ability to decode any section of the compressed image without having to decode the entire data set. In the case of functional 4D medical images, resolution and quality scalability, and ROI coding are not a main interest, as these sequences are usually used in batch to conduct statistical image analysis and are not used individually as single volumes.

Most medical image compression algorithms are comprised of three main components, a decorrelation algorithm, a main compression engine and a formatting scheme. These three components are illustrated in Fig. 1.1.

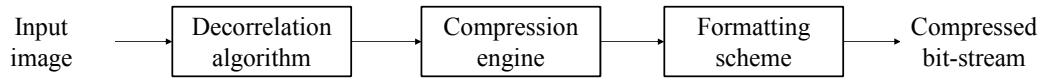


Fig. 1-1: Block diagram of a medical image compression scheme

Decorrelation algorithms exploit the data redundancies in medical images by employing a predictive model or a multi-resolution model. Prediction models minimize the difference between consecutive samples, slices or volumes and generate residual data by using either motion compensation and estimation or differential pulse code modulation. On the other hand, multi-resolution models decorrelate image data by using a transform, such as discrete wavelet transforms (DWT) or discrete cosine transforms (DCT). The decorrelated images and residual data are then losslessly compressed using an entropy encoder, such as Huffman, arithmetic and run-length coding [10-12]. Prior to entropy coding, the decorrelated images are

usually quantized to improve the compression performance. In lossless coding schemes, no quantization is applied to the decorrelated data. Finally, a formatting process is applied to the compressed image to ensure that the encoded data is suitable for transmission and storage purposes.

In the remainder of this chapter, we present our thesis statement and list the research objectives. Next, we give an overview of the state-of-the art on medical image compression, including an overview of the wavelet transform and motion compensation and estimation, and the entropy coding algorithms EBCOT and CABAC. Finally, we list the thesis contributions and present the organizations of the following chapters.

1.2 Thesis Statement and Objectives

1.2.1 Thesis statement

In this thesis, we address different aspects of lossless medical image compression for efficient storage, transmission and on demand scalable access for 3D and 4D data. We approach the problem from an energy reduction perspective, in which the main objective is to reduce the overall energy of the medical imaging data, and thus attain a higher compression performance. Our research has shown that, the wavelet transform is an efficient way to pack the energy of 3D medical images in a few coefficients and provide resolution scalability. Furthermore, by exploiting the inherent properties of 3D medical images in the wavelet domain, such as the structural symmetry of the ROI depicted by the image data, an efficient energy reduction can be achieved. Our research has also shown that motion compensation and estimation is capable to efficiently reduce the energy of 4D medical images by exploiting the data redundancies within each volume and between volumes. Moreover, we demonstrate that the use of entropy coders tailored to the characteristics of the medical imaging data provides a higher lossless compression performance than that achieved by the state-of-the-art, while providing scalability and ROI coding.

1.2.2 Thesis objectives

Based on the description in the previous section, the objectives of this research include the following:

1. Develop a prediction method for 3D medical images that exploits the symmetrical characteristics of the depicted ROI to efficiently reduce the energy of the data in the wavelet domain.
2. Develop an entropy coder for residual data of 3D medical images in the wavelet domain that achieves a high compression ratio and generates a bit-stream that is scalable by resolution and quality, and provides ROI coding capabilities.
3. Develop a prediction method for 4D medical images that exploits data redundancies within each volume and between volumes to efficiently reduce the energy of the data.

4. Develop an entropy coder for residual data of 4D medical images that achieves a high compression ratio.

To address these objectives, we develop various lossless compression methods for 3D and 4D medical images that reduce the energy of the data by using the wavelet transform, motion compensation and estimation, and prediction methods. Moreover, we propose modified versions of the state-of-the-art entropy coders EBCOT and CABAC tailored to the characteristics of the medical imaging data in order to attain high compression ratios, scalability and ROI coding capabilities.

1.3 Overview of the State-of-the-art on Compression of 3D and 4D Medical Images

1.3.1 State-of-the-art on 3D medical image compression

The state-of the-art on compression of 3D medical images includes several scalable compression methods which provide resolution and quality scalability [13-23]; and ROI coding of a 3D section of the image, i.e., volume of interest (VOI) coding [19,24,25]. These methods are based on the DWT, whose inherent properties produce a bit-stream that is resolution-scalable. Quality scalability is then achieved by employing bit-plane based entropy coding algorithms that exploit the dependencies between the location and value of the wavelet coefficients. Examples of such techniques are the ones proposed by Schelkens et al. [13], Wang et al. [14], Xiong et al. [16], Menegaz et al. [16, 17], Srikanth [18] and Krishnana et al. [19].

Schelkens et al. presented an overview of several state-of-the-art 3D wavelet coders suitable for medical images. The reviewed coders are based on quad-tree and block-based coding, layered zero-coding principles and context-based arithmetic coding. All of the different coding principles that were evaluated produce embedded bit-streams that can be decoded up lossless reconstruction and provide quality and resolution scalability.

Wang et al. proposed a 3D medical image compression technique based on a separable non-uniform 3D wavelet transform. The method employs one filter bank on each 2D slice and then a second filter bank along the slice (third) dimension. This approach improves the compression performance when the resolution of the third dimension is lower than that of the individual slices. Performance evaluations show an improvement on compression ratio of 70% compared to 2D wavelet compression.

Xiong et al. presented a study on lossy-to-lossless compression of volumetric medical images using 3D integer wavelet transforms. The study introduces a 3D integer wavelet packet transform structure and focuses on context modelling for efficient arithmetic coding of the transform coefficients. The authors also incorporate these ideas into previous wavelet coding algorithms such as the 3D Set Partitioning in Hierarchical Trees (SPIHT) [26] and the 3D Embedded Sub-band Coding with Optimized Truncation (ESCOT) algorithms [27].

Menegaz et al. proposed in [16] a 3D wavelet-based coding system with 3D encoding/2D decoding capabilities. The algorithm provides 2D decoding by encoding every 2D sub-band independently. Fast access to any slice of a particular volume is possible by decoding only the corresponding information. Performance results show an improvement on compression ratios of 3D images with high correlation along the third dimension compared to 2D compression algorithms. Menegaz et al. also proposed a lossy-to-lossless object-based coding scheme for volumetric MRI data [17]. The scheme exploits the diagnostic relevance of the different regions of a particular volume for rate allocation. It allows for random access to any object at any bit-rate. The novelty of the scheme is the absence of artifacts along the object borders, which is obtained by selecting, in each sub-band, the set of wavelet coefficients which are necessary for reconstructing the object as if the whole set of sub-band samples were available. Lossless compression is achieved by using a separable 3D-DWT performed by the lifting steps scheme.

Srikanth proposed an improvement to mesh-coding techniques for brain MRI, which consists of eliminating any clinically irrelevant background to code only the brain ROI. The method is based on context-based mesh generation using spatial edges and optical flow between two consecutive slices and context-based entropy encoding of the residuals after motion compensation. Performance evaluations of the technique show an improvement in bit-rate of about 2 bpv (bits per voxel), compared to those achieved by other state-of-the art 3D wavelet-based coders, including the 3D Embedded Zerotree Wavelet (3D-EZW) coder [28].

Krishnana et al. proposed a compression technique for medical images for transmission and remote visualization. The technique, which is based on the JPEG2000 standard and the JPEG2000 Interactive Protocol (JPIP), transmits data in a multi-resolution and progressive fashion [29,30]. The technique is aimed at 3D medical images and offers the possibility of transmitting VOIs progressively. The authors exploit the packetization features of JPEG2000 to provide a prioritized transmission of packets according to the user's request.

1.3.2 State-of-the-art on 4D medical image compression

The state-of the-art on compression of 4D medical images includes a limited number of compression methods, as this is relatively new area of research. These 4D compression methods use mainly the DWT, motion compensation and estimation or differential pulse code modulation to decorrelate the data and improve the compression performance. [31-35]. Among these techniques, the most advanced are the ones proposed by Lalgudi et al. [31], Kassim et al. [32] and Zeng et al. [33].

Lalgudi et al. studied the extension of JPEG2000 to four dimensions and proposed a method to encode fMRI using a DWT and JPEG2000. The proposed technique first applies a one dimensional (1D) DWT along the fourth dimension, followed by a 1D-DWT along the third dimension with the resulting transform slices then encoded using JPEG2000.

Kassim et al. proposed a lossy-to-lossless compression technique for 4D medical images using a combination of 3D integer wavelet transform (3D-IWT) and 3D motion compensation. 3D-SPIHT is then

used to code the transform coefficients [36]. Performance evaluations show an improvement in coding efficiency compared to JPEG2000 and 3D-SPIHT.

Zeng et al. proposed an extension to four dimensions of the EZW algorithm to compress arbitrarily sized medical images. The algorithm provides a superior lossy compression performance compared to 2D slice compression using the 2D version of the EZW algorithm.

1.4 Review of Decorrelation Algorithms

In this section, we present a brief review of two of the most important decorrelation algorithms employed for compression of 3D and 4D medical images: the discrete wavelet transform and motion compensation and estimation. These algorithms are the foundation of the advances in medical image compression presented in this thesis.

1.4.1 The discrete wavelet transform

The basic idea of the discrete wavelet transform (DWT) is to represent a signal as a superposition of a wavelet basis that is discretely sampled [37,38]. The coefficients of the basis can then be used to reconstruct the original signal.

The DWT can be extended to multidimensional signals. In the case of 2D signals or images, the 2D-DWT gives a spatial and frequency representation of the image. Each level d of the 2D-DWT decomposes its input into four spatial frequency sub-bands denoted as LL_d, LH_d, HL_d, and HH_d. The approximation low-pass sub-band, LL, is a coarser version of the original signal, while the other sub-bands represent the high frequency details in the horizontal, vertical and diagonal directions, respectively. The decomposition is usually iterated on the approximation low-pass sub-band, which for most natural images contains most of the energy [38]. Figure 1.2 illustrates the implementation of a one-level 2D-DWT decomposition.

The wavelet transform has many features that make it suitable for scalable compression, such as representation of an image at different resolutions and packing of most of the energy in a few wavelet coefficients. In the case of 3D signals or images, each level of decomposition, d, of the transform decomposes the 3D image input into eight 3D frequency sub-bands denoted as LLL_d, LLH_d, LHL_d, LHH_d, HLL_d, HLH_d, HHL_d, and HHH_d. The approximation low-pass sub-band, LLL, is a coarser version of the original 3D image, while the other sub-bands represent the details of the image. The decomposition is also iterated on the approximation low-pass sub-band. Figure 1.3 illustrates the implementation of aone-level 3D-DWT decomposition.

For lossless compression, an integer wavelet transform (IWT) is required [39-41]. An integer wavelet transform maps integers to integers and allows for perfect invertibility with finite precision arithmetic, which is required for perfect reconstruction of a signal. Although integer wavelet transforms are not easy to

construct, the construction becomes very simple using the lifting steps scheme by first factoring the traditional discrete wavelet transform into lifting steps and then applying a rounding operation at each step [42]. Among current wavelet-based lossless compression methods that employ the IWT, the JPEG2000 standard is one of the most advanced ones [30]. JPEG2000 incorporates functionalities such as lossless and lossy compression, spatial and quality scalability, ROI coding, random access to the bit-stream and error-resilient coding.

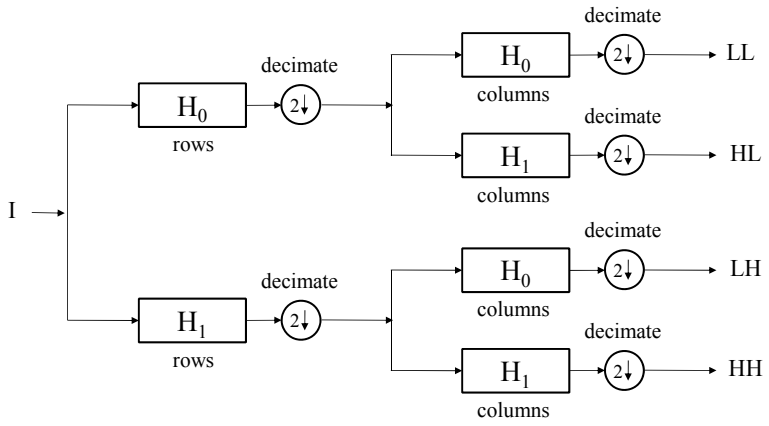


Fig. 1-2: Diagram illustrates the steps in of one-level wavelet decomposition along the *x* and *y* axes of a 2D image. The original image, I, is processed in one dimension with low- (H_0) and high-pass (H_1) filters. The output is decimated by two (every other output value from the low-pass and high-pass filters is deleted), and the process is repeated in the second dimension. The end result is a blur image (the approximation low-pass sub-band LL) and three directionally sensitive detail images (sub-bands HL, LH and HH).

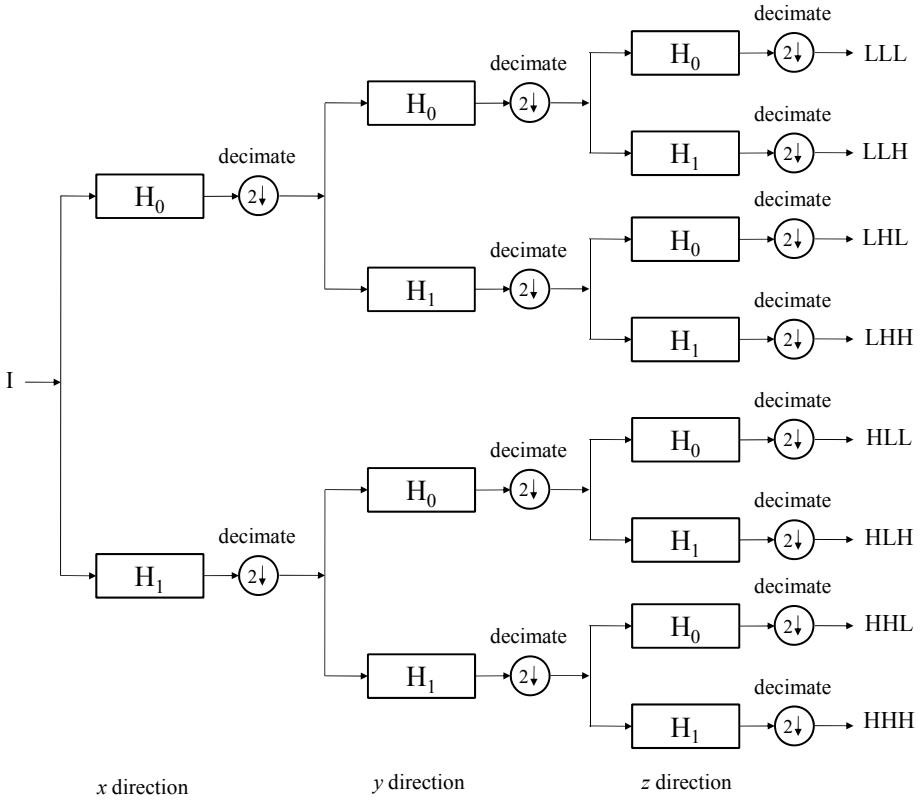


Fig. 1-3: Diagram illustrates the steps in a one-level wavelet decomposition along the *x*, *y* and *z* axes of a 3D image. The original image, I, is processed in one dimension with low- (H_0) and high-pass (H_1) filters. The output is decimated by two and the process is repeated in the second and third dimensions. The end result is a blur image (the approximation low-pass sub-band LLL) and seven directionally sensitive detail images (sub-bands LLH, LHL, LHH, HLL, HLH, HHL, and HHH).

1.4.2 Motion compensation and estimation

Motion compensation and estimation is an efficient way to reduce data redundancies in the temporal dimension between frames of video sequences [43-45]. The simplest and most widely used motion compensation technique is known as block matching, which estimates the amount of motion on a block-by-block basis which minimizes the difference between two frames [46]. A video sequence is first divided into groups of pictures (GOP) and each GOP is coded independently. The first picture (i.e., frame) of a GOP is usually encoded as an "Intra" frame using no prediction or only information contained in the picture itself. The remaining pictures are typically encoded as "Inter" frames using motion compensation on a block-by-block basis by first dividing them into non-overlapping blocks of pixels, usually of 16×16 pixels. Each block is predicted from a block of equal size in a reference picture. The difference between the position of a block in the current picture and the position of the matched block in the reference picture is denoted by a motion vector. This process is illustrated in Fig. 1.4.

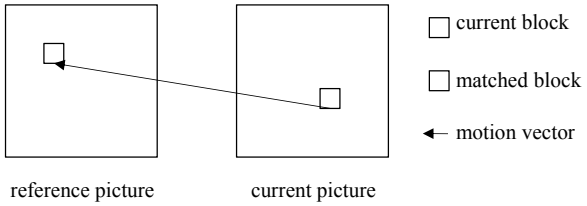


Fig. 1-4: Block matching process. Diagram shows a block in the current picture and the matched block in the reference picture.

Matched blocks of the current picture comprise the current predicted picture, which is subtracted from the current picture to calculate the residual. In multi-fame motion compensation (MF-MC), multiple pictures may be employed as reference, as illustrated in Fig 1.5.

Among current compression methods based on MF-MC, the H.264/AVC standard is one of the most advanced ones [47]. H.264/AVC incorporates several advanced coding features such as block-based spatial prediction for Intra-frames, MF-MC with bi-directional prediction and variable block size matching, and two unique algorithms for entropy coding of residual and motion vector data: the context-adaptive variable-length coding (CAVLC) and the context-based adaptive binary arithmetic coding (CABAC) algorithms [47,48].

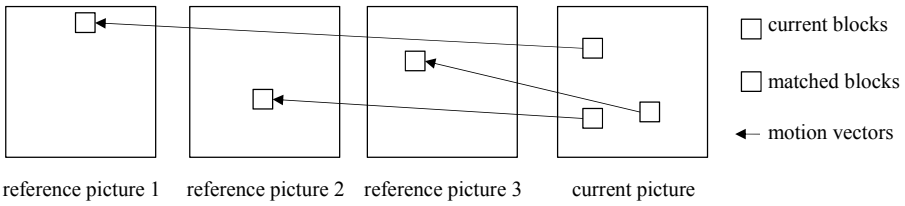


Fig. 1-5: Multi-fame motion compensation process. Diagram shows how multiple pictures may be employed as a reference to predict the current picture.

1.5 Review of Entropy Coding Algorithms

In this section, we present a brief review of two state-of the-art entropy coding algorithms: the embedded block coding with optimized truncation (EBCOT) [49] algorithm and CABAC [48]. We first present an overview of the entropy coding principles for lossless compression. We then review the most important coding principles of EBCOT and CABAC, which we employ in the compression methods presented in this thesis.

1.5.1 Overview of entropy coding

Entropy coding for data compression can be split in two processes: 1) modeling and 2) coding. The first process, modeling, estimates the probabilities of the symbols to be coded. The second process, coding, produces a sequence of bits from the symbol probabilities estimated during the modeling process. Since an accurate probability estimation is required in order to achieve a high compression rate, modeling is one of the

most important tasks in entropy coding.

The entropy of a memoryless information source, measure in bits per symbol, is defined as:

$$H(X) = -\sum_{x} P(x) \log_2 P(x) \tag{1.1}$$

where X is a random variable, and x is a possible realization with probability $P(x)$ (i.e, the probability that x takes one value from a set of 2^n values, n = 8 for 8 bpp images). Equation (1) is called the zero*th*-order entropy since no consideration is given to the fact that variable X may have statistical dependencies with its neighbours. If the probability of occurrence of samples in the source depends on L - 1 previous samples, then conditional entropy may be defined. For conditional entropy, we assume that L – 1 components from L-dimensional vector $\mathbf{X}$, x_1, x_2, . . ., x_{L-1}, have already been received by the decoder. The conditional entropy of component x_L is then defined as [50]:

$$H(X_L | X_1, X_2, ..., X_{L-1}) = -\sum_{\mathbf{x}} P(\mathbf{X}) \log_2 P(x_L | x_1, x_2, ..., x_{L-1}) \tag{1.2}$$

The higher the number of components used to calculate the conditional entropy in Eq. 1.2 (i.e., the higher the order of the entropy of component x_L), the lower the resulting entropy. Shannon's Source Coding Theorem states that there is a relationship between the symbol probabilities and the corresponding compressed bit-stream [51]. According to this theorem, it is possible to code without distortion, a source of entropy H using $H + \varepsilon$ bits per sample, where ε is an arbitrary small positive quantity. The maximum achievable compression rate C, is then defined by

$$C = \frac{\text{Bit rate of original raw data}}{\text{Bit rate of encoded data}} = \frac{n}{n_a} = \frac{n}{H + \varepsilon} \approx \frac{n}{H} \tag{1.3}$$

Equation 1.3 indicates that the higher the order of the entropy used, the higher the maximum achievable compression rate. One of most efficient entropy coding techniques is arithmetic coding, which achieves high compression rates, especially if the statistical dependencies between samples are taken into account in the coding process. Arithmetic coding is a type of variable-length entropy encoding that converts a source of symbols into a bit-stream that represents high-probable symbols using fewer bits and less-probable symbols using more bits, with the goal of using fewer bits in total [11]. As opposed to other entropy coding techniques that separate the input message into its component symbols and replace each symbol with a code word, arithmetic coding encodes the entire message into a single number, a fraction n where ($0.0 \leq n < 1.0$). Binary arithmetic coding [52], a special case of arithmetic coding where the symbols can only take two values (i.e., “1” or “0”), is the foundation of the state-of-the-art algorithms EBCOT and CABAC.

1.5.2 Embedded block coding with optimized truncation (EBCOT)

The Embedded Block Coding with Optimized Truncation (EBCOT) is an entropy coding algorithm for wavelet-transformed images. EBCOT partitions each sub-band into small blocks of samples, called code-blocks, and generates a separate scalable bit-stream for each code-block, Cb_i. The bit-stream of each code-block Cb_i may be independently truncated to any of a collection of different lengths, R_i^n. EBCOT is capable to independently compress relative small code-blocks (e.g., 32×32 or 64×64 samples each) with an embedded bit-stream consisting of a large number of truncation points, R_i^n, such that most of these truncation points lie on the convex hull of the corresponding rate-distortion curve [49].

The output bit-stream is then generated by concatenating the suitably truncated representations of each code-block, Cb_i, including sufficient auxiliary information to identify the truncation points, n_i, and the corresponding lengths, R_i^n. This output bit-stream is resolution scalable, since the information representing the individual code-blocks and hence the sub-bands and resolution levels are clearly delineated. Also, the output bit-stream possesses a random access feature, since given any ROI and a wavelet transform with finite support kernels, it is possible to identify the region within each sub-band and hence the code-blocks which are required to correctly reconstruct the ROI.

Although the output bit-stream is composed by a set of SNR scalable block bit-streams, it is not quality-scalable, because only a single truncation point and length are identified for each code-block. In order to create a quality-scalable output bit-stream, EBCOT collects incremental contributions from the various code-blocks into a number of quality layers. In this way, truncating the output bit-stream to any whole number of layers yields a rate-distortion optimal representation of the image; while truncating the output bit-stream to an intermediate bit-rate yields a bit-stream which is approximately optimal, provided the number of quality layers is relatively large. Each quality layer must include auxiliary information to identify the size of each code-block's contribution to the layer. When the number of layers is large, only a sub-set of the code-blocks contributes to any given layer, introducing additional auxiliary information for those code-blocks that provide no contribution to a specific layer. The layered bit-stream concept is illustrated in Fig. 1.6.

Unlike well known wavelet-based scalable image compression algorithms, such as the embedded zerotree wavelet coding (EZW) [28] and the set partitioning in hierarchical trees (SPIHT) [26] algorithms, which offer only quality scalability, EBCOT is capable to provide various bit-stream organizations, ranging from single-layer bit-streams which possess only the resolution scalable and random access attributes, through bit-streams with a large number of layers, which offer quality scalability, in combination with the resolution scalable and random access attributes.

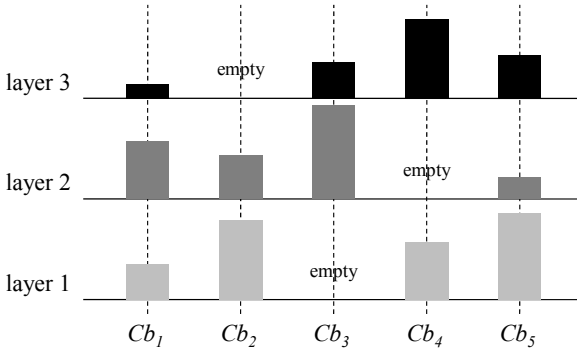


Fig. 1-6: Progressive appearance of embedded code-block bit-streams in quality layers. Diagram shows five blocks and three layers. Note how some code-blocks provide no contribution to some of the layers.

Block coding in EBCOT

The block coding algorithm in EBCOT is based on context adaptive binary arithmetic coding and bit-plane coding. Each block Cb_i, is partitioned into a 2D sequence of sub-blocks, Cb_i[j], whose size is typically 16×16 samples. For each bit-plane p, EBCOT first encodes information to identify those sub-blocks that contain one or more significant samples; all other sub-blocks are by-passed in the remaining coding phases for that bit-plane. A sample s is said to be significant in the current bit-plane p if and only if $|s| \geq 2^p$. EBCOT employs four coding primitive operations to code new information for a single sample s in the current bit-plane p. The four coding primitive operations are: 1) zero coding (ZC); 2) run-length coding (RLC); 3) sign coding (SC); and 4) magnitude refinement (MR). A combination of the ZC and RLC primitives encodes whether or not sample s becomes significant in the current bit-plane p. The significance of sample s is coded using ten different context models that exploit the correlation between the significance of sample s and that of its adjacent neighbours. If sample s becomes significant in the current bit-plane p, the SC primitive encodes the sign information of sample s using five different context models. The MR primitive uses three different context models to encode the value of sample s only if it is already significant in the current bit-plane p.

Rate distortion optimization in EBCOT

As mentioned earlier, the bit-stream of each code-block Cb_i may be independently truncated to any of a collection of different lengths, R_i^n. The truncated bit-streams are then organized into a number of quality layers to create a quality-scalable bit-stream. The contribution from Cb_i to distortion in the reconstructed image is denoted D_i^n, for each truncation point, n. EBCOT assumes that the distortion metric is additive [49]:

$$D = \sum_i D_i^{n_i} \tag{1.4}$$

where D denotes the overall distortion and n_i denotes the truncation point selected for code-block Cb_i. The distortion metric employed by EBCOT approximates Mean Square Error (MSE) and is defined as [49]:

$$\hat{D}_i^n = \omega_{g_i}^2 \sum_{\mathrm{k} \in Cb_i} (\hat{s}_i^n[\mathrm{k}] - s_i[\mathrm{k}])^2 \tag{1.5}$$

where $s_i[\mathrm{k}]$ denotes the 2D sequence of samples in code-block Cb_i, $\hat{s}_i^n[\mathrm{k}]$ denotes the quantized representation of these samples associated with truncation point n, and ω_{g_i} denotes the L2-norm of the wavelet basis functions for the sub-band g_i, to which code-block Cb_i belongs.

The optimal selection of truncations points is found by minimizing the distortion subject to a constraint $R^{\max}$, on the available bit-rate [49]:

$$R^{\max} \geq R = \sum_i R_i^{n_i} \tag{1.6}$$

1.5.3 Context-based adaptive binary arithmetic coder (CABAC)

One of the most advanced entropy coding algorithms for residual data obtained from a motion compensation and estimation process is the context-based adaptive binary arithmetic coder (CABAC) [48]. The encoding process of CABAC consists of three main steps: 1) code assignment, 2) context modeling, and 3) binary arithmetic coding [48].

In the first *code assignment* step, a unique binary code is assigned to the magnitude of a residual sample (level, hereafter). Each level is assigned a unique binary code that consists of two parts: the prefix and the suffix, as illustrated in Fig. 1.7. For levels smaller than 15, the binary codes consist of only a prefix part, while for levels greater than or equal to 15, the binary codes consist of a prefix and a suffix part.

$$|\, decimal\ \ value \,| > 0 \rightarrow \underbrace{[prefix]\ [suffix]}_{Binary\ \ code}$$

Fig. 1-7: Code assignment in CABAC. The magnitude of a decimal value greater than zero is assigned a unique binary code comprised of a prefix and a suffix part.

The prefix part is formed by a truncated unary binarization scheme, where each level with a magnitude x, is assigned a unary code word consisting of x "1" bits plus a terminating "0" bit. For $x < 14$, the code is given by the unary code, whereas for $x = 14$ the terminating "0" bit is neglected such that the code of $x = 14$ is given by a codeword consisting of x "1" bits only. The suffix part, on the other hand, is formed by an Exponential Golomb code of zero order (EG0) for the value of (level−15). These types of codes, which were

first proposed by Teuhola [53] in the context of run-length coding schemes, have been proven to be optimal prefix-free codes for geometrically distributed sources [54].

In the second *context modeling,* a probability model is assigned to the binary codes, which then, in the *binary arithmetic coding* step, drives the actual coding engine to generate a sequence of bits as a coded representation of the symbols according to the model distribution. In CABAC, contexts are specified by a modeling function $F: \boldsymbol{T} \rightarrow \boldsymbol{C}$ operating on a pre-defined set of past symbols, or context template $\boldsymbol{T}$, assuming a related set of contexts $\boldsymbol{C} = \{0,1,\ldots,\boldsymbol{C}\text{-}1\}$. For each symbol b to be coded, a conditional probability $p(b|F(z))$ is estimated by switching between different probability models according to the already coded neighboring symbols z belongs $\boldsymbol{T}$. After encoding b using the estimated conditional probability $p(b|F(z))$, the probability model is updated with the value of the encoded symbol b. Therefore, CABAC estimates $p(b|F(z))$ on the fly by tracking the actual source statistics. In order to limit the number of contexts used and avoid inaccurate estimates of $p(b|F(z))$, CABAC imposes two important restrictions. First, limited context templates consisting of a few neighbors of the current symbol to encode are employed. Second, context modeling is restricted to selected bits of the binarized symbols.

In the third *binary arithmetic coding* step, the bits of the binary codes are coded into a more compact representation by using a binary arithmetic coder according to a set of probability models [48]. In CABAC, binary arithmetic coding is based on the principle of recursive interval subdivision, where a lower bound L and a range R represent the corresponding coding interval. Assuming $p_{LPS} \in (0,0.5]$ is an estimate of the probability of the least probable symbol (LPS), the coding interval is subdivided into two subintervals, one interval of width $R_{LPS} = R \cdot p_{LPS}$, which is associated with the LPS, and the dual interval of width $R_{MPS} = R - R_{LPS}$, which is assigned to the most probable symbol (MPS) having a probability estimate of $1 - p_{LPS}$. Depending on the observed binary decision, either identified as the LPS or the MPS, the corresponding subinterval is then chosen as the new current interval. A binary value pointing into that interval represents the sequence of binary decisions processed so far, whereas the range of that interval corresponds to the product of the probabilities of those binary symbols. The minimum precision of bits specifying the lower bound of the final interval is then employed to unambiguously identify the coded sequence of binary decisions.

1.6 Thesis Contributions and Organization

1.6.1 Summary of contributions

The main contributions of this thesis are summarized as follows:

- We show that the IWT is an efficient way to attain resolution and quality scalability for 3D medical image lossless compression.
- We propose a scalable lossless compression method for 3D medical images that exploits the symmetrical characteristics of the data to achieve a higher lossless compression ratio. Our method

employs 2D wavelet-based compression of slices within a 3D medical image. Specifically, it encodes slices by first applying a 2D-IWT, followed by block-based intra-band prediction of the resulting sub-bands. The block-based intra-band prediction method exploits the structural symmetry of the ROI depicted by the image data to reduce the energy of the sub-bands. Residual data generated by the intra-band prediction method are then compressed using a modified version of the EBCOT algorithm designed according to the characteristics of the residual data.

- We propose a 3D scalable medical image compression method with optimized VOI coding. The proposed method reorders the output bit-stream after encoding, so that those bits with greater contribution to the distortion of a VOI and are included earlier while simultaneously maximizing the overall reconstruction quality of the 3D image. In other words, our method is designed to optimize the reconstruction quality of a VOI at any bit-rate, while incorporating partial background information and allowing for gradual increase in peripheral quality around the VOI. The method employs the 3D-IWT and a modified EBCOT algorithm with 3D contexts. The bit-stream reordering procedure is based on a weighting model that incorporates the position of the VOI and the mean energy of the wavelet coefficients to create an optimized scalable layered bit-stream.
- We show that multi-frame motion compensation and estimation is an efficient way to reduce the data redundancies of slices of 4D medical images within each volume and between volumes.
- We propose a lossless compression method specifically designed for fMRI data. The proposed method employs a new multi-fame motion compensation process, which efficiently exploits the spatial and temporal correlations of fMRI data, and a new CABAC to losslessly compress the residual and motion vector data generated by the motion compensation process. The proposed multi-frame motion compensation uses a 4D search, bi-directional prediction and variable-size block matching for motion estimation. The proposed CABAC takes into account the probability distribution of the residual and motion vector data in order to assign proper probability models to these data and improve the compression performance.

1.6.2 Thesis organization

This is a manuscript-based thesis which follows the specifications required by the University of British Columbia for this format. In addition to this introductory chapter, this thesis includes three chapters which were originally prepared for journal publication and have been slightly modified in order to present a logical progression. The remaining chapters of this thesis are organized as follows. In Chapter 2 we present a novel symmetry-based technique for scalable lossless compression of 3D medical image data. We also present a modified version of the EBCOT algorithm, tailored according to the characteristics of the data, to encode the residual data generated after prediction to provide resolution and quality scalability. We demonstrate that, when evaluated over a large set of 3D medical images, the proposed method achieves an average

improvement of 14% in lossless compression ratios when compared to the state-of-the-art compression methods 3D-JPEG2000, JPEG2000 and H.264/AVC intra-coding. In Chapter 3, we present a novel 3D scalable compression method for medical images with optimized VOI coding. We show how VOI coding may be attained by an optimization technique that reorders the output bit-stream after encoding, so that those bits with greater contribution to the distortion of a VOI are included earlier while allowing for gradual increase in peripheral quality around the VOI. We also demonstrate that at various bit-rates, the proposed method achieves a higher reconstruction quality, in terms of the peak signal-to-noise ratio (PSNR), than those achieved when employing only random access to the bit-stream with no optimization and by 3D-JPEG2000 with ROI coding. In Chapter 4, we show that multi-frame motion compensation and estimation is an efficient method to reduce data redundancies in the spatial an temporal directions of 4D medical images. In Chapter 5, we extend our research work presented in Chapter 4 and present a method for lossless compression of fMRI data based on a new multi-frame motion compensation process that employs 4D search, variable-size block matching and bi-directional prediction; and a new CABAC designed for lossless compression of the residual and motion vector data. We demonstrate that, on a large sample of real fMRI data of varying resolutions, the proposed method achieves superior lossless compression ratios than those achieved by two state-of-the-art methods; 4D-JPEG2000 and H.264/AVC, with an average improvement of 13%.

Finally, in Chapter 6, we present our conclusions, summarize the contributions of this thesis and propose several suggestions for future research in this topic.

1.6.3 Source of test data

We tested the performance of our proposed method using a large set of real 3D and 4D medical images. The details of the source of the test data used in each chapter are as follows:

Chapter 2

- CT sequences from the 'Visible Male' and 'Visible Woman' data sets maintained by the National Library of Medicine (NLM). The entire 'Visible Male' CT data consist of axial CT scans of the entire body taken at 1mm intervals at a pixel resolution of 512×512 with each pixel made up of 12 bits of gray tone. There are 1,871 cross-sections for CT images. The complete male data set is approximately 15 gigabytes. The 'Visible Female' CT data set has the same characteristics as the 'Visible Male'. The data are available at http://www.nlm.nih.gov/research/visible/
- MRI sequences from the Internet Brain Segmentation Repository (IBSR), supported by the National Institute of Health, US. The IBSR is a World Wide Web resource providing access to magnetic resonance brain image data and segmentation results contributed and utilized by researchers from all over the world. The MRI sequences were provided by the Center for Morphometric Analysis at Massachusetts General Hospital and are available at http://www.cma.mgh.harvard.edu/ibsr/

- MRI sequences provided by the Biomedical Signal and Image Computing Laboratory, Department of Electrical and Computer Engineering, The University of British Columbia.
- MRI sequences provided by Dr. Alex L. MacKay, Department of Medical Physics, The University of British Columbia.
- MRI sequences provided by Dr. Roger Tam, Department of Radiology, The University of British Columbia

Chapter 3

- CT sequences from the 'Visible Male' and 'Visible Woman' data sets maintained by the National Library of Medicine (NLM).
- MRI sequences provided by the Biomedical Signal and Image Computing Laboratory, Department of Electrical and Computer Engineering, The University of British Columbia.
- MRI sequences provided by Dr. Alex L. MacKay, Department of Medical Physics, The University of British Columbia.

Chapter 4

- fMRI sequences provided by Dr. Martin McKewon, Department of Neurology, The University of British Columbia.
- PET sequences provided by Dr. Vesna Sossi, Department of Medical Physics, The University of British Columbia.
- 4D MRI sequences from The Mayo Clinic, US. Downloaded from http://nova.nlm.nih.gov/Mayo/ (last time visited: April, 2007)

Chapter 5

- fMRI sequences provided by Dr. Martin McKewon, Department of Neurology, The University of British Columbia.

1.6.4 Implementation details of the compression methods used in performance evaluations

We compared the performance of our proposed compression methods to that of several state-of-the-art compression methods. The implementation details of these state-of-the-art compression methods are as follows:

- JPEG2000. We used the Kakadu implementation of JPEG2000 available at http://www.kakadusoftware.com
- 3D-JPEG2000. We used the Kakadu implementation of 3D-JPEG2000 available at http://www.kakadusoftware.com.

- 4D-JPEG2000. We implemented 4D-JPEG2000 by first applying a one dimensional discrete wavelet transform across the fourth dimension of medical images. We then compressed the resulting transform slices using 3D-JPEG2000, as previously described.
- H.264/AVC. We used the H.264/AVC reference software available at http://iphome.hhi.de/suehring/tml
- H.264/AVC intra-coding. We used the H.264/AVC intra-coding reference software available at http://iphome.hhi.de/suehring/tml

Our proposed methods were implemented using several publicly available libraries and code. Specifically, we employed libraries and code from the OpenJPEG2000 implementation, available at http://www.openjpeg.org/, to implement the EBCOT algorithm; we also employed the H.264/AVC reference software, available at http://iphome.hhi.de/suehring/tml, to implement the motion compensation and estimation algorithms.

It is important to note that the use of publicly available libraries and code facilitates the implementation of our proposed algorithms into current compression standards such as JPEG2000, 3D-JPEG2000 and H.265/AVC.

1.7 References

[1] D. Feng, "Information technology applications in biomedical functional imaging," *IEEE Transactions on Information Technology in Biomedicine,* vol. 3, no. 3, pp. 221-230, September 1999.

[2] J. Prince and J. Links. Medical Imaging Signals and Systems. Prentice Hall, 2006.

[3] R. Logeswaran. "Compression of Medical Images for Teleradiology" in Teleradiology. S. Kumar and E. A. Krupinski. Berlin: Springer, 2008, pp. 21-31.

[4] H. Lee, Y. Kim, A. H. Rowberg, and E. A. Riskin, "Statistical Distributions of DCT Coefficients and their Application to an Interframe Compression Algorithm for 3-D Medical Images," *IEEE Transactions on Medical Imaging*, vol. 12, no. 3, pp. 478-485, 1993.

[5] N. Strickland, "PACS (picture archiving and communication systems): filmless radiology," *Archives of Disease in Childhood,* vol. 1, no. 83, pp. 82–86, July 2000

[6] S. K. Mun and M. Freedman, "The role of PACS in redesigning the radiologic practice," *Proceedings of the National Forum on Military Telemedicine On-Line Today, 1995. 'Research, Practice, and Opportunities'*, pp. 39-42, March 1995.

[7] C. Xinhua and H.K. Huang, "Current status and future advances of digital radiography and PACS," *IEEE Engineering in Medicine and Biology Magazine*, vol. 19, no. 5. pp. 80 – 88, September 2000.

[8] R.B. Jones, S.M. McGhee, D. McGhee, "Patient on-line access to medical records in general practice," *Health Bulletin*, vol. 2, no. 50, pp. 143-150, March 1992

[9] A. K. Jain, "Image Data Compression: A Review, " *Proceedings of the IEEE,* vol. 69, no. 3, pp. 349-402, March 1981

[10] D. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE,* vol. 40, no. 9, pp. 1098-1101, September 1952

[11] B. Fu, K. K. Parhi, "Generalized multiplication-free arithmetic coders," *IEEE Transactions on Communications,* vol. 45, no.5, pp. 497-501, May 1997

[12] H. Meyr, H. Rosdolsky and T. Huang, "Optimum run-length codes," *IEEE Transactions on Communications,* vol 22, no. 6, pp. 826-835, June 1974

[13] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto and J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Trans. on Medical Imaging,* vol. 22, no. 3, pp. 441-458, March 2003

[14] J. Wang and H.K. Huang, "Medical Image Compression by Using Three-Dimensional Wavelet Transformation," *IEEE Transactions on Medical Imaging*, vol. 15, no. 4, pp.547-554, August 1996

[15] Z. Xiong, X. Wu, S. Cheng and J. Hua, "Lossy-to-lossless compression of medical volumetric images using three-dimensional integer wavelet transforms," *IEEE Trans. on Medical Imaging,* vol. 22, no. 3, pp. 459-470, March 2003

[16] G. Menegaz and J.P. Thirian, "Three-dimensional encoding/two-dimensional decoding of medical data," *IEEE Trans. on Medical Imaging,* vol. 22, no. 3 , pp. 424-440, March 2003

[17] G. Menegaz and J. P. Thirian, "Lossy to Lossless Object-Based Coding of 3-D MRI Data," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1053-1061, September 2002

[18] R. Srikanth and A.G. Ramakrishnan, "Contextual Encoding in Uniform and Adaptive Mesh-Based Lossless Compression of MR Images," *IEEE Trans. on Medical Imaging*, vol. 24, no. 9, pp. 1199-1206, September 2005

[19] K. Krishnan. M. Marcellin, A. Bilgin and M. Nadar, "Efficient Transmission of Compressed Data for Remote Volume Visualization," *IEEE Trans. on Medical Imaging,* vol. 25, no. 9, pp. 1189-1199, September 2006

[20] X. Wu and T Qiu, "Wavelet coding of volumetric medical images for high throughput and operability," *IEEE Trans. on Medical Imaging,* vol. 24, no. 6, pp. 719-727, June 2005.

[21] K. Krishnan, M. W. Marcellin, A. Bilgin and M.S. Nadar, "Compression/decompression strategies for large volume medical imagery," *Proc. SPIE/Medical Imaging 2005: PACS and Imaging Informatics*, San Diego, CA, February 2004

[22] E. Siegel, K. Siddiqui, J. Johnson, O. Crave, Z. Wu, J. Dagher, A. Bilgin, M. Marcellin, M. Nadar and B. Reiner, "Compression of Multislice CT: 2D vs. 3D JPEG2000 and Effects of Slice Thickness," *Proc. SPIE/Medical Imaging 2005: PACS and Imaging Informatics,* vol. 5748, pp. 162-170, April 2005.

[23] W. Hwang, C. Chine and K. Li, "Scalable Medical Data Compression and Transmission Using Wavelet Transform for Telemedicine Applications," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, no. 1, pp. 54-63, March 2003

[24] Y. Liu and W. A. Pearlman, "Region of interest access with three-dimensional SBHP algorithm," *Proc. SPIE 6077*, pp. 17-19, 2006.

[25] C. Doukas and I. Maglogiannis, "Region of Interest Coding Techniques for Medical Image Compression," *IEEE Engineering in Med. and Biol. Magazine*, vol. 25, no. 5, pp. 29-35, September-October 2007

[26] A. Said and W. Pearlman, "A new fast and efficient image coded based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 6, pp. 243-250, June 1996

[27] J. Xu, Z. Xiong, S. Li, and Y. Zhang, "3-D embedded sub-band coding with optimal truncation (3D-ESCOT)," *J. Appl. Comput. Harmon. Anal.*, vol. 10, pp. 290–315, May 2001.

[28] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3462, December 1993

[29] D. S. Taubman and R. Prandolini, "Architecture, philosophy and performance of JPIP: Internet protocol standard for JPEG2000," *Proceeding of the SPIE International Symposium on Visual Communication*, vol. 5150, pp. 791-805, 2003

[30] Information Technology—JPEG 2000 Image Coding System—Part 2: Extensions, ISO/IEC 15 444-2, 2002

[31] H. G. Lalgudi, A. Bilgin, M. W. Marcellin, and M. S. Nadar, "Compression of fMRI and ultrasound images using 4D-SPIHT," *Proceedings of 2005 International Conference on Image Processing*, vol. 2, pp. 11-14, September 2005

[32] A. Kassim, P. Yan, and W. S. Lee, "Motion compensated lossy-to-lossless compression of 4-D medical images using integer wavelet transforms," *IEEE Trans. on Information Technology in Biomedicine,* vol. 9, no. 1, pp. 132-138, March 2005

[33] L. Zeng, C.P. Jansen, S. Marsch, M. Unser, and P.R. Hunziker, "Four-dimensional wavelet compression of arbitrarily sized echocardiographic data," *IEEE Trans. on Medical Imaging,* vol. 21, no. 9, pp. 1179-1187, September 2002

[34] L. Ying, and W.A. Pearlman, "Four-dimensional wavelet compression of 4-D medical images using scalable 4D-SBHP," *2007 Data Compression Conference*, pp. 233-242, March 2007

[35] L. Zeng, C. Jansen, M. Unser and P. Hunziker, "Extension of wavelet compression algorithms to 3D and 4D image data: exploitation of data coherence in higher dimensions allows very high compression ratios", *Proceedings of SPIE Int. Soc. Opt. Eng.,* vol. 4478, pp. 427-433, 2001

[36] B.J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate embedded video coding with 3D set partitioning in hierarchical trees (3D SPIHT)," *IEEE Transactions on. Circuits and Systems for Video Technology,* vol. 10*,* pp. 1365-1374, December 2000

[37] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math.,* vol.41, pp. 909-996, 1998

[38] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674-693, June 1989

[39] S. Dewitte and J. Cornelis, Lossless integer wavelet transform, *"IEEE Signal Processing Letters,"*, vol. 4, no. 6, pp. 158-160, June 1997

[40] A. R. Calderbank, I. Daubechies, W. Sweldens and B.L, Yeo, "Wavelet transforms that map integers to integers," *Appl. Comput. Harmon. Anal.,* vol. 5, no. 3, pp. 332-369, 1998

[41] A. R. Calderbank, I. Daubechies, W. Sweldens and B.L, Yeo, "Lossless image compression using integer to integer wavelet transforms," in *Proc. Int. Conf. Image Procession (ICIP),* 1997, pp. 569-599

[42] I. Daubechies and W. Sweldens, "Factoring wavelet transform into lifting steps," *J. Fourier Anal. Appl.,* vol. 41, no. 3, pp. 247-269, 1998

[43] T. Sikora, "MPEG digital video-coding standards," *IEEE Signal Preocessing Magazine,* vol. 14, no. 5, pp. 82-100, September 1997

[44] G. Sullivan, P. Topiwala and A. Luthra, "The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions", *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 5558, pp. 454-474, August 2004

[45] M. Budagavi and J.D. Gibson, "Multiframe video coding improved performance over wireless channels", *IEEE Trans. On Image Processing,* vol. 10, no. 2, pp. 252-265, February 2001

[46] J. R. Jain and A. K. Jain, "Displacement measurement and its applications in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, no. 12, pp. 1799-1808, 1981

[47] G. Sullivan, P. Topiwala and A. Luthra, "The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions", *Proc. SPIE Int. Soc. Opt. Eng.*, vol 5558, pp. 454-474, August 2004

[48] D. Marpe, H. Schwarz and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard", *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 1, no. 7, pp. 620-623, July 2003

[49] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing,* vol. 9, no. 7, pp. 1158-1170, July 2000

[50] S. S. Yu, and N. P. Galatsanos, "Binary Decompositions for High-Order Entropy Coding of Grayscale Images," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 6, no. 1, pp. 21-31, February 1996

[51] C. E. Shannon, "The mathematical theorem of communication," Parts I and 11, Bell Syst. Tech. J., vol. 27, pp. 379 and 623, 1948

[52] A. Moffat, R. Neal, and I. Witten, "Arithmetic coding revisited," *ACM Transactions on Information Systems,* vol. 16, no. 3, pp. 256-294, 1998.

[53] J. Teuhola, "A compression method for clustered bit-vectors," *Information Processing Letters*, vol. 7, pp. 308-311, October 1978

[54] R. Gallager and D. Van Voorhis, "Optimal source codes for geometrically distributed integer alphabets", *IEEE Transactions on Information Theory*, vol. 21, pp. 228-230, March 1975

Chapter 2

2. Symmetry-Based Scalable Lossless Compression of 3D Medical Image Data[1]

2.1 Introduction

Over the past decade, three dimensional (3D) medical imaging has become a main pillar of clinical research and practice, making this type of image data an integral part of any patient's records. These routinely acquired 3D medical images, e.g., magnetic resonance imaging (MRI) and computed tomography (CT), comprise two-dimensional (2D) images (or slices) of cross-sections of the anatomy of a region of interest (ROI) and are usually very large in file size.

The increasingly sophisticated data archiving and networking technologies render it crucial to improve lossless compression techniques that would enable efficient storage and quick access to 3D medical images for future study and follow-up. An important number of technical advances in lossless compression of 3D medical images have been reported in the literature. Most of these methods exploit correlations between slices within the data volume to improve the compression performance by either employing a 3D discrete wavelet transform (3D-DWT) [1-5] or motion compensation and estimation [6]. Although these methods attain the three most desirable properties of any compression method for 3D medical images, namely 1) high lossless compression ratios; 2) resolution scalability, and 3) quality scalability; they do not exploit some of the inherent characteristics of medical image data, such as the symmetry of the depicted region of interest (ROI); thus leaving room for further improvement.

In this Chapter, we propose a novel scalable lossless compression method for 3D medical images that attains the three desired properties listed before and exploits the symmetrical characteristics of the data to achieve a higher lossless compression ratio. Our method employs two-dimensional (2D) wavelet-based compression of slices within a 3D medical image. Specifically, it encodes slices by first applying a 2D integer wavelet transform (2D-IWT), followed by block-based intra-band prediction of the resulting sub-bands. The block-based intra-band prediction method exploits the structural symmetry of the ROI depicted by the image data to reduce the energy of the sub-bands. Residual data generated by the intra-band prediction method are then compressed using a modified version of the embedded block coder with optimized truncation (EBCOT) [7] designed according to the characteristics of the residual data.

---

[1] A version of this chapter has been published. V. Sanchez, R. Abugharbieh, and P. Nasiopoulos, "Symmetry–Based Scalable Lossless Compression of 3D Medical Image Data," *IEEE Transactions on Medical Imaging*, vol. 28, no. 7, pp. 1062–1072, 2009

Performance evaluations over a large set of 3D medical images show that our proposed method achieves higher lossless compression ratios when compared to other state-of-the-art, namely 3D-JPEG2000, JPEG2000 and H.264/AVC intra-coding methods.

The rest of the chapter is organized as follows. We describe our proposed compression method in Sections 2.2 and 2.3. Performance evaluations and comparisons to other state-of-the-art compression methods are presented in Section 2.4 and conclusions are given in Section 2.5

2.2 Proposed Scalable Lossless Compression Method

The block diagram of our proposed method for scalable lossless compression of a 3D medical image is illustrated in Fig. 2.1

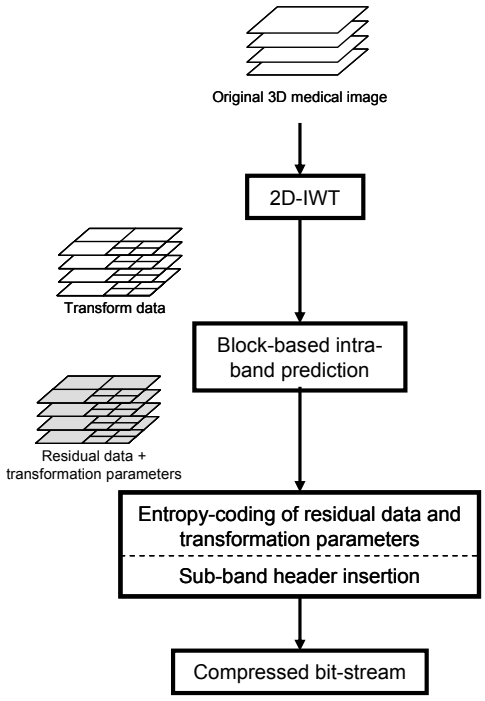


Fig. 2-1: Block diagram of the proposed scalable lossless compression method. 2D-IWT: two-dimensional integer wavelet transform.

Each slice is first decomposed using a 2D-IWT with n levels of decomposition. The resulting sub-bands are then coded independently by first employing a novel block-based intra-band prediction method followed by entropy coding of the residual coefficients into l quality layers using a modified version of the embedded block coding with optimized truncation (EBCOT) algorithm [7]. The transformation parameters generated by the block-based intra-band prediction method are compressed using a variable-length coder

(VLC) and are included in a sub-band header. The block-based intra-band prediction method is aimed at reducing the energy of each sub-band by predicting the value of the coefficients on a block-by-block basis by exploiting the symmetrical content of the sub-band.

The inherent properties of the IWT in conjunction with the coding of the residual data using a modified version of the EBCOT algorithm produce a bit-stream that is resolution- and quality-scalable, namely, the bit-stream contains distinct sub-sets, $\tau_{d,q}$, such that $\bigcup_{k=0}^{q} \tau_{d,k}$ together represent the samples of only those sub-bands in resolution level *d*, at some quality level *q*. The resolution and quality scalability of the resulting bit-stream provides interesting possibilities for access and transmission of the data. For instance, in telemedicine a client with limited bandwidth using a remote image retrieval system can obtain a thumbnail view or low-resolution full view of a slice at different qualities by first transmitting and decoding the corresponding lowest frequency sub-bands.

The following sections detail our proposed block-based intra-band prediction method and the modified EBCOT algorithm.

2.2.1 Block-based intra-band prediction method

In wavelet-based compression methods, the dependencies between wavelet coefficients are exploited before entropy coding in order to achieve a better compression performance. Several coding methods for wavelet coefficients exploit the dependencies between the location and value of the coefficients across sub-bands. Examples of such methods are the embedded zerotree wavelet coding (EZW) [14] and the set partitioning in hierarchical trees (SPIHT) algorithms [15]. These *inter-band* coding techniques group the insignificant coefficients in trees that span across the sub-bands and code them with "zero" symbols. The zero regions in the significance maps are then approximated as a set of tree structured regions. Inter-band coding performs well in smooth images where the approximation low-pass sub-band contains most of the energy and the high frequency sub-bands contain just a few non-zero valued coefficients whose spatial arrangement can be exploited using a tree structure. However, in images with a high number of edges, such as the slices of most 3D medical images, the edge information spreads out in the whole sub-band structure resulting in high-frequency sub-bands with a high-energy content and a distribution of non-zero valued coefficients whose spatial arrangement is difficult to code using a tree structure [16,17]. For this type of images, *intra-band* coding is a more suitable approach, where only the dependencies within a sub-band are exploited. Here, the zero regions of the significance maps are represented by a set of independent rectangular zero regions of fixed and variable sizes. Examples of such methods are the embedded block coding by optimized truncation (EBCOT) [7] and the square partitioning (SPQ) algorithms [18]. The coding gains resulting from using intra-band coding have been reported in [19, 20].

Based on the above observations, in this work we employ intra-band coding on the sub-bands generated by the 2D-IWT. We first reduce the energy of the sub-bands by predicting the value of coefficients

on a block-by-block basis, followed by block-by-block entropy coding of the resulting residual data. In order to tune the design of our intra-band coding method, we studied the characteristics of slices of 3D medical images. Such data usually depict cross-sections of the anatomy of an ROI and normally contain a large amount of edge information. Furthermore, in most of the cases the depicted ROI is symmetrical due to the inherent symmetry of human anatomy. These characteristics are illustrated in Fig. 2.2.

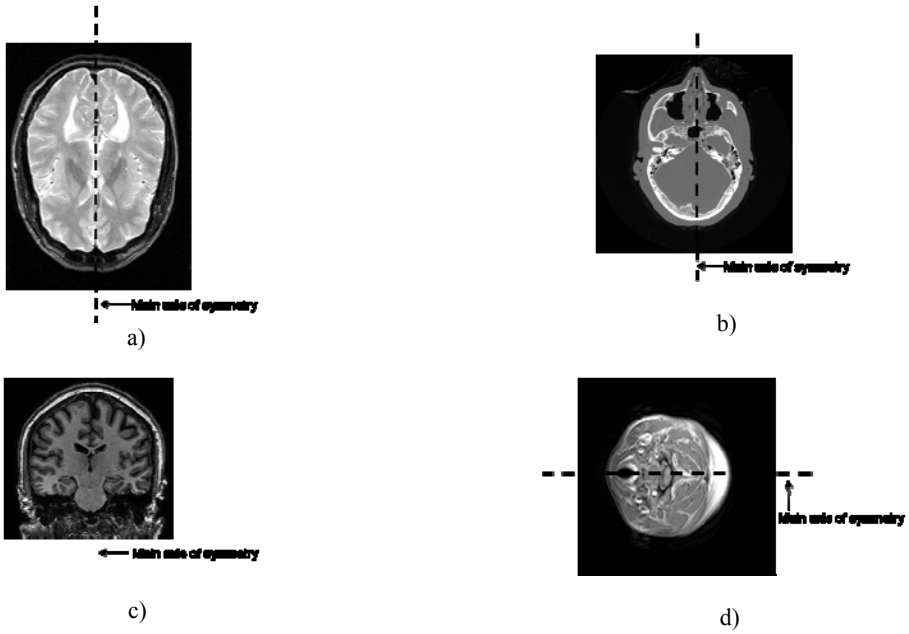
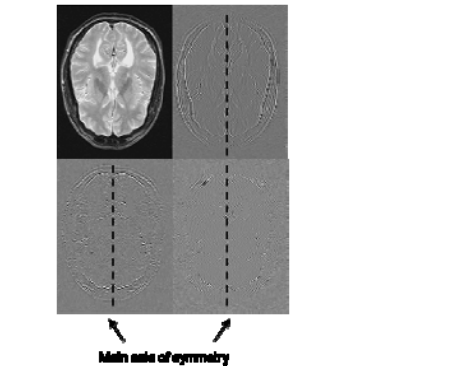


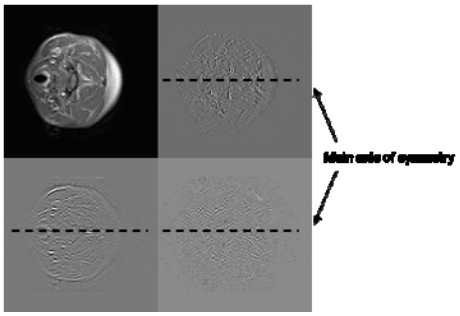
Fig. 2-2: Example slices of 3D medical images. a) Axial view of MRI brain scan; b) axial view of a CT head scan; c) coronal view of MRI brain scan; and d) axial view of MRI scan of a human spinal cord. Note the high number of edges and the symmetry of the region of interest (ROI) depicted in each slice.

Following the application of a 2D-IWT on slices, edge information tends to spread out in most of the sub-bands generating a distribution of non-zero valued coefficients (e.g., high energy sub-bands). The symmetry of the original structures depicted in the slices tends to be preserved as symmetries in the location and value of wavelet coefficients within each sub-band. This is illustrated in Fig. 2.3.

Figure 2.4 illustrates the horizontal high-pass sub-band (LH_1) of the MRI slice illustrated in Fig. 2.3(a). One can easily identify a main vertical axis of symmetry centered in the sub-band. The sub-band can thus be partitioned into two areas along this axis of symmetry. Let us denote the area to the left of the axis as LH_1-L and the area to the right of the axis as LH_1-R (see Fig. 2.4(b)). If LH_1-L is to be flipped along the axis of symmetry, it would be expected to provide a good approximation to LH_1-R and can therefore be used to predict LH_1-R (see Fig. 2.4(c)). Sub-band LH_1 may then be reduced to LH_1-L and the prediction error (or residual) between LH_1-R and $G(LH_1\text{-L})$, where $G(a)$ denotes a spatial transformation on area a, in this case a horizontal flip (see Fig. 2.4(d)).

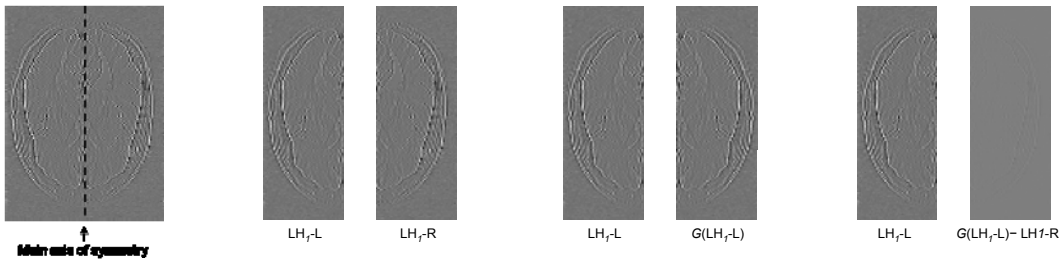


a)



b)

Fig. 2-3: Two-dimensional integer wavelet transform (2D-IWT) sub-bands for one level decomposition of the slice of a) the axial view of the brain MRI scan of Fig. 2-2(a); and b) the axial view of the spinal cord MRI scan of Fig. 2-2(d). Note the symmetry of the sub-bands.



a) b) c) d)

Fig. 2-4: Horizontal high-pass sub-band (LH_1) of a slice of an MRI volume of the axial view of a human head. a) Original LH_1 sub-band. b) LH_1 sub-band after partition into two areas, LH_1-L and LH_1-R, along the vertical axis of symmetry. c) LH_1 sub-band after flipping area LH_1-L along the axis of symmetry. d) LH_1 sub-band after calculating the prediction error (or residual) between LH_1-R and $G(LH_1\text{-L})$. $G(a)$ denotes a spatial transformation on area a, in this case a horizontal flip.

Figure 2.5 illustrates the vertical high-pass sub-band (HL_1) of the MRI slice shown in Fig. 2.3(b). In this case, one can easily identify a main horizontal axis of symmetry centered in the sub-band. This axis divides the sub-band into an upper and lower area, denoted as HL_1-U and HL_1-L, respectively. After flipping

HL_1-U along the axis of symmetry, sub-band HL_1 is reduced to HL_1-U and the residual between HL_1-L and $G(HL_1\text{-U})$, where $G(a)$ denotes a spatial transformation on area a, in this case a vertical flip.

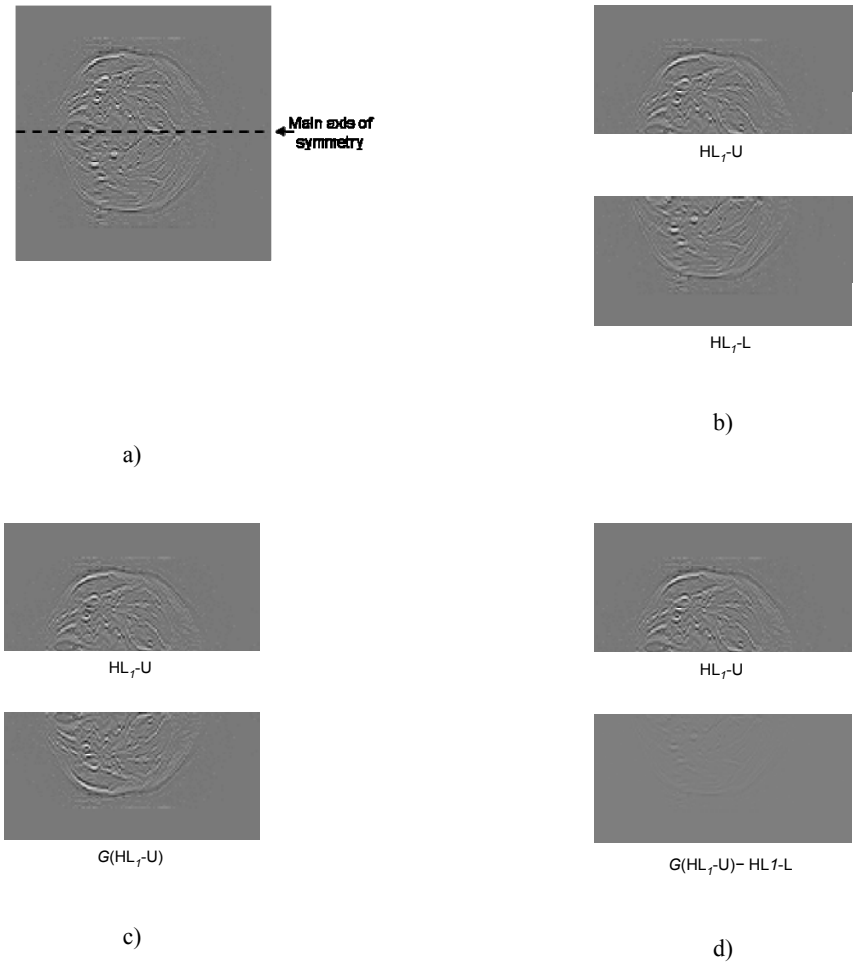


Fig. 2-5: Vertical high-pass sub-band (HL_1) of a slice of an MRI volume of the axial view of a human spinal cord. a) Original HL_1 sub-band. b) HL_1 sub-band after partition into two areas, HL_1 -U and HL_1-L, along the axis of symmetry. c) HL_1 sub-band after flipping area HL_1-U along the axis of symmetry. d) HL_1 sub-band after calculating the prediction error (or residual) between HL_1-L and $G(HL_1\text{-U})$. $G(a)$ denotes a spatial transformation on area a, in this case a vertical flip.

Consider now the case of the low-pass sub-band (LL_1) of a slice of an MRI volume of the sagittal view of a human spinal cord as illustrated in Fig. 2.6(a). Here, it is not possible to identify a main axis of symmetry for the sub-band. Nevertheless, it is possible to identify smaller regions that are symmetrical. Fig. 2.6(b) illustrates a small region of the LL_1 sub-band where a vertical axis of symmetry can be identified. This region can thus be partitioned into two areas along this axis of symmetry, left and right. It is then possible to predict the right area using the spatially transformed version of the left area, as explained before.

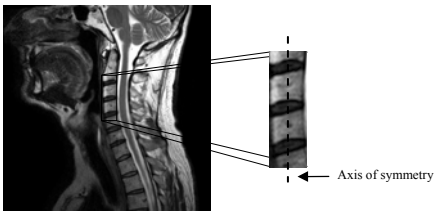


Fig. 2-6: Low-pass sub-band (LL_1) of a slice of an MRI volume of the sagittal view of a human spinal cord. Figure shows a small symmetrical area and the corresponding axis of symmetry.

Our intra-band coding method exploits this symmetry to predict the value of coefficients. Due to the fact that it may not be possible to identify a main axis of symmetry, or there may exist more than one axis of symmetry in a sub-band, and that these axes may be located at different positions (e.g., horizontally, vertically, diagonally), we employ a block-based approach to predict coefficients on a block-by-block basis. Working with small blocks gives us the flexibility to identify the block that, after a spatial transformation $G(a)$, best approximates a current block without having to identify the corresponding axis of symmetry. In this work we employ eight spatial transformations that modify the spatial relationship between coefficients in a block, mapping coefficient locations in an input block to new locations in an output block. These spatial transformations and their corresponding geometric operations are summarized in Table 2.1.

We first divide each sub-band into blocks of 16×16 coefficients. We further, to determine if smaller blocks result in an improvement in coding performance. We selected these sizes for We process blocks in a raster-scanning order and predict each block using any previously coded block after undergoing a spatial transformation $G(a)$. We further divide the 16×16 blocks into two blocks of 8×8 and four blocks of 4×4 coefficients to determine if smaller blocks result in improvement in coding performance. We select the block size that attains the best coding performance. Our experiments on a large set of medical images have shown that block sizes of 16×16, 8×8 and 4×4 provide the best trade-off between coding efficiency and computational complexity (coding time).

Let b_c denote the current block to be predicted at position c following a raster scanning order. Let Q denote the set of candidate blocks for prediction of block b_c as defined in (2.1):

$$Q = \{b_1, b_2, b_3, ..., b_{c-2}, b_{c-1}\} \tag{2.1}$$

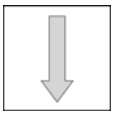
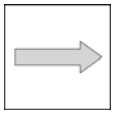
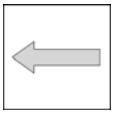
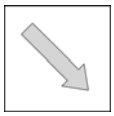
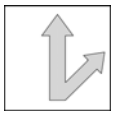
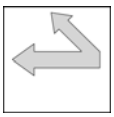
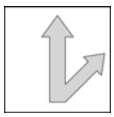
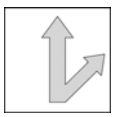
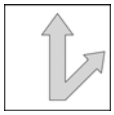
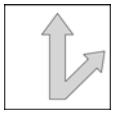
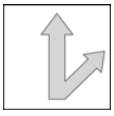
We apply K=8 different spatial transformations (see Table 2.1) to each block in Q to produce K different sets of spatially transformed blocks. The k^{th} set of spatially transformed blocks is defined as:

$$\widetilde{Q}^k = \{G^k(Q_1), G^k(Q_2), ..., G^k(Q_{c-2}), G^k(Q_{c-1})\} \tag{2.2}$$
$$k \in [1, K]$$

where $G^k(a)$ denotes the k^{th} spatial transformation on block a and Q_n denotes the n^{th} element of Q. We then

compute the sum-of-absolute differences (*SAD*) between each transformed block in $\tilde{Q}^k$ ($k \in [1, K]$) and block b_c to produce K sets of *SAD* values. The k^{th} set of *SAD* values is defined as:

Table 2-1: Spatial transformations

Index k	Geometric operation	Sample input block	Corresponding output block
1	Vertical flip		
2	Horizontal flip		
3	Diagonal flip		
4	Left rotation(90°)		
5	Right rotation(90°)		
6	Left rotation(90°) + vertical flip		
7	Right rotation(90°) + vertical flip		
8	No operation		

$$J^k = \left\{ SAD(\tilde{Q}_1^k, b_c), SAD(\tilde{Q}_2^k, b_c), ..., SAD(\tilde{Q}_{c-1}^k, b_c) \right\} \quad k \in [1, K] \tag{2.3}$$

where *SAD*(*a,b*) denotes the sum-of-absolute differences between block *a* and block *b,* and $\widetilde{Q}_n^k$ denotes the n^{th} element of $\widetilde{Q}^k$. The minimum residual block, *m*, is then defined as the element of J^k ($k \in [1, K]$) with the minimum *SAD* value:

$$m = \min\left(\min(J^1), \min(J^2), ..., \min(J^{K-1}), \min(J^K)\right) \tag{2.4}$$

We then compute the energy of block b_c and the energy of block *m*. The energy of a block *a*, e_a ,is defined as:

$$e_a = \sum_{x=1}^{X} \sum_{y=1}^{Y} w(x,y)^2 \tag{2.5}$$

where *w*(*x, y*) is the value of the coefficient at position (*x, y*) in block *a*. If the energy of block *m* is less than the energy of block b_c, we employ the reference block *r*, associated to *m* to predict b_c; otherwise no prediction is employed. The residual data is obtained by subtracting the predicted sub-band from the original sub-band. To reconstruct a sub-band from the corresponding residual data, it is necessary to know the information about the reference block and the spatial transformation used to predict each block. The prediction process thus generates a transformation parameter, t_c, for block b_c as defined by Eq. 2.6:

$$t^c = \{p, k\} \tag{2.6}$$

where *p* is the position of the reference block *r* in the sub-band following a raster-scan order and *k* is the index of the corresponding spatial transformation (see Table 2.1).

For the compression of the transformation parameters, we employ a variable length coder (VLC) with a single infinite-extent codeword table generated using an Exp-Golomb code of order k=0 (EG0), which has simple and regular decoding properties [21]. The compressed transformation parameters are included in the final bit-stream as a sub-band header.

For the compression of the residual data, we employ a modified version of the EBCOT algorithm to account for the fact that the characteristics of the residual data may differ from the characteristics of the original coefficients. The following section details our modified EBCOT algorithm.

2.3 Entropy Coding of Residual Data

In this section, we describe the entropy coding of residual data using a modified version of the EBCOT algorithm, which generates a bit-stream that is both resolution and quality scalable.

As summarized in Chapter 1, section 1.5.2, EBCOT is an image compression algorithm for wavelet-transformed images. EBCOT partitions each sub-band in small blocks of samples, called code-blocks, and generates a separate scalable bit-stream for each code-block, Cb_i. The algorithm is based on context adaptive binary arithmetic coding and bit-plane coding, and employs four coding passes to code new information for a single sample s in the current bit-plane p. The coding passes are: 1) zero coding (ZC); 2) run-length coding (RLC); 3) sign coding (SC); and 4) magnitude refinement (MR). A combination of the ZC and RLC passes encodes whether or not sample s becomes significant in the current bit-plane p. A sample s is said to be significant in the current bit-plane p if and only if $|s| \geq 2^p$. The significance of sample s is coded using ten different context models that exploit the correlation between the significance of sample s and that of its adjacent neighbors. If sample s becomes significant in the current bit-plane p, the SC pass encodes the sign information of sample s using five different context models. The MR pass uses three different context models to encode the value of sample s only if it is already significant in the current bit-plane p.

The bit-stream of each code-block Cb_i may be independently truncated to any of a collection of different lengths, R_i^n. The truncated bit-streams are then organized into a number of quality layers to create a quality-scalable bit-stream. The contribution from Cb_i to distortion in the reconstructed image is denoted D_i^n, for each truncation point, n. EBCOT assumes that the distortion metric is additive [7]:

$$D = \sum_i D_i^{n_i} \tag{2.7}$$

where D denotes the overall distortion and n_i denotes the truncation point selected for code-block Cb_i. The distortion metric employed by EBCOT approximates Mean Square Error (MSE) and is defined as [7]:

$$\hat{D}_i^n = \omega_{g_i}^2 \sum_{\mathbf{k} \in Cb_i} (\hat{s}_i^n[\mathbf{k}] - s_i[\mathbf{k}])^2 \tag{2.8}$$

where $s_i[\mathbf{k}]$ denotes the 2D sequence of samples in code-block Cb_i, $\hat{s}_i^n[\mathbf{k}]$ denotes the quantized representation of these samples associated with truncation point n, and ω_{g_i} denotes the L2-norm of the wavelet basis functions for the sub-band g_i, to which code-block Cb_i belongs.

The optimal selection of truncations points is found by minimizing the distortion subject to a constraint $R^{\max}$, on the available bit-rate [7]:

$$R^{\max} \geq R = \sum_i R_i^{n_i} \tag{2.9}$$

In this work, we propose a new context assignment for arithmetic coding and a new distortion metric to account for the fact that the characteristics of the residual data may differ from those of the original coefficients.

2.3.1 Proposed context assignment for arithmetic coding

After intra-band prediction, we code the significance of a residual sample *r* with respect to the current bit-plane *p* using the ZC and RLC passes, as there may be correlation between the significance of *r* and that of its adjacent neighbors. Similarly, we use the MC pass to code the value of *r* when the sample is already significant in the current bit-plane. However, the sign information of the residual data may drastically differ from that of the original coefficients as a consequence of the intra-band prediction process. We thus defined a new context assignment for the SC pass based on the statistics of the sign information of residual data.

Taubman illustrated in [7] that the sign bits of adjacent coefficients present high statistical dependencies which can be exploited to improve coding efficiency. Namely, the author claimed that horizontally adjacent coefficients from LH (horizontal high-pass) sub-bands tend to have the same sign, whereas vertically adjacent coefficients tend to have opposite signs. This claim was substantiated by the fact that coefficients of LH sub-bands have predominantly low-pass horizontal power spectra and high-pass vertical power spectra, due to the aliasing introduced by the high-pass filtering and decimation operations [7]. After intra-band prediction, the horizontal and vertical power spectra of the LH sub-band tends to be more flat, as the prediction process reduces the edge information and thus, the overall energy of the sub-band. Figure 2.7 illustrates the horizontal and vertical power spectra of the LH sub-bands of three slices of three different 3D medical images before and after intra-band prediction. Note that the horizontal power spectrum before intra-band prediction is indeed predominantly low-pass, while the vertical power spectrum is indeed predominantly high-pass. Also note the flattening effect that intra-band prediction has on the spectra.

The sign bits of adjacent residual samples still present statistical dependencies. However, empirical evidence now suggests that the sign statistics are approximately Markov; i.e., the sign of sample *r* depends upon the sign and significance information of its immediate eight neighbors. We thus define new context models for the SC pass. Let $X(r)$ denote the sign bit of sample *r*, *I*(*r*) denote the number of insignificant neighbors of *r*, *F*(*r*) denote the number of positive neighbors of *r,* and *N*(*r*) denote the number of negative neighbors of *r*. The context assignment for the SC pass for residual data of LH sub-bands is summarized in Table 2.2. The binary valued symbol which is coded with respect to the corresponding context is $X(r) \cdot \hat{X}(r)$. Since EBCOT transposes the HL, the same context assignment may be applied to the LH and HL sub-bands; for simplicity, this context assignment is extended without modification to the less important LL and HH sub-bands.

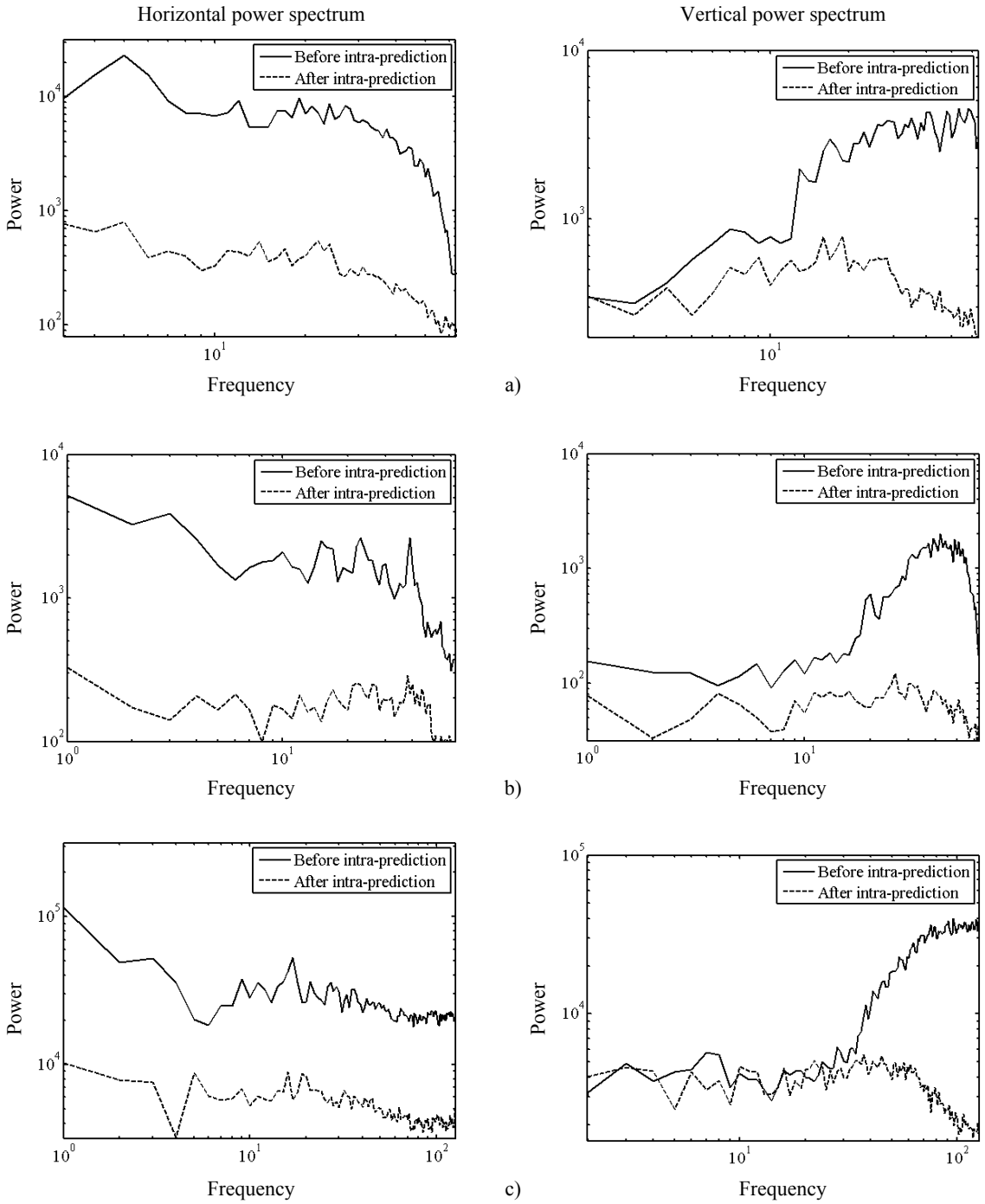


Fig. 2-7: Horizontal and vertical power spectra of the LH (horizontal high-pass) sub-band of a slice of a) an MRI volume of the axial view of a human head; b) an MRI volume of the axial view of a human spinal cord and c) an MRI volume of the sagittal view of a human knee. Note that the horizontal power spectrum before intra-band prediction is predominantly low-pass, while the vertical power spectrum is predominantly high-pass. Also note the flattening effect that intra-band prediction has on the spectra.

Table 2-2: Assignment of the seven (Sign Coding) SC contexts for residual data based on the signs and significance of the immediate eight neighbors

$I(r)$	$F(r)$	$N(r)$	$\hat{X}(r)$	Context label
8	0	0	1	0
0	8	0	1	1
0	0	8	-1	2
0	4	4	1	3
0	[4,8]	**X**	1	4
0	**X**	[4,8]	-1	5
X	[1,3]	0	1	6
X	0	[1,3]	-1	7

X: don't care

2.3.2 Proposed distortion metric

Consider the case of prediction of block b_c in sub-band g_i at position c (following a raster scanning order) using block r_{p1} at position $p1$ as reference. In order to reconstruct b_c, it is necessary to reconstruct block r_{p1}, whose reconstruction may in turn depend on the reconstruction of reference block r_{p2} at position $p2$. These dependences may extend to the first block of sub-band g_i, which is not coded using prediction.

Due to this inherent coding order, the distortion of a code-block Cb_i at truncation point n increases as samples are reconstructed using samples of reference blocks whose bit-streams are also truncated at point n. We model this increase in distortion as an added error in the samples:

$$\hat{D}_i^n = \omega_{g_i}^2 \sum_{\mathrm{k} \in Cb_i} \left[\left(\hat{s}_i^n[\mathrm{k}] + \hat{e}_i^n[\mathrm{k}] \right) - s_i[\mathrm{k}] \right]^2 \tag{2.10}$$

where $\hat{e}_i^n[\mathrm{k}]$ denotes the 2D sequence of error samples to be added to the 2D sequence of samples in $\hat{s}_i^n[\mathrm{k}]$ associated with truncation point n and is defined as:

$$\hat{e}_i^n[\mathrm{k}] = \sum_{p=1}^{P} \hat{r}_p^n[\mathrm{k}] - \sum_{p=1}^{P} r_p[\mathrm{k}] \tag{2.11}$$

where $r_p[\mathrm{k}]$ denotes the 2D sequence of samples of reference block at position p, P denotes the position of the last reference block needed to reconstruct the samples of Cb_i, and $\hat{r}_p^n[\mathrm{k}]$ denotes the quantized representation of these samples associated with truncation point n.

2.4 Results and Discussion

Three sets of experimental results were obtained. The first set evaluated the performance of the block-based intra-band prediction method in reducing the energy of the sub-bands. The second set evaluated the over-all performance of the proposed method for lossless compression. The third set evaluated the performance of the proposed method for scalable compression.

2.4.1 Evaluation of block-based intra-band prediction

We tested the performance of the block-based intra-band prediction method in reducing the energy of the sub-bands of 90 different slices of various 3D medical images after one level of decomposition using the Le Gall 5/3 wavelet filter implemented using the lifting step scheme [12]. Test slices 1 to 30 depict cross-sections of MRI volumes of an axial view of a human head, where a main vertical axis of symmetry can be identified. Test slices 31 to 60 depict cross-sections of MRI volumes of an axial view of a human spinal cord, where a main horizontal axis of symmetry can be identified. Test slices 61 to 90 depict cross-sections of MRI volumes of a sagittal view of a human spinal cord and thus, no main axis of symmetry can be identified. We employed blocks of 16×16, 8×8 and 4×4 coefficients and allowed the intra-band prediction method to select the block size that yielded the smallest residual data. For the case of axial view slices, we also calculated the energy of the residual data obtained by employing intra-band prediction and only two areas of equal size. We generated these two areas of equal size by partitioning the sub-bands along the main axis of symmetry. We then predicted one area using the second area after applying the spatial transformation on the second area that yielded the smallest residual. In Table 2.3 we report the mean of the ratios between the energy of the residual data and the energy of the original sub-bands. Results summarized in Table 2.3 show that our block-based intra-band prediction method achieves reductions in energy of up to 66%, with minimum reductions of 47%. It is worth noting that that the intra-band prediction method only uses prediction if the energy of a residual block is less than the energy of the original block, therefore, there is never an energy increase in the sub-bands. Compared to the ratios reported in column 3 (intra-band prediction using two areas of equal size), our intra-band prediction method reduces the energy of sub-bands by up to 40%, confirming the advantages of using block-based prediction even when a main axis of symmetry can be identified in the slices.

We also report in column 5 of Table 2.3 the index of the spatial transformation used in the majority of the blocks of each type of sub-band. As expected, for the test slices with a symmetrical ROI along a main vertical axis of symmetry (slices 1-30), the spatial transformation with index k=2 (horizontal-flip, see Table 2.1) is the one used in the majority of the blocks of all sub-bands. For the test slices with a symmetrical ROI along a main horizontal axis of symmetry (slices 31-60), the spatial transformation with index k=1 (vertical-flip, see Table 2.1) is the most commonly used in all sub-bands. For the test slices depicting the sagittal view of a human spinal cord (slices 61-90), the spatial transformations with indices k={1,2,3} (vertical-, horizontal- and diagonal-flip, see Table 2.1) are the most commonly used in all sub-bands.

Table 2-3: Mean of the ratios between the energy of the residual 2D-IWT sub-bands of 60 MRI slices after prediction and the original sub-bands

Slice: *pixels per slice*	Sub-band	Mean of ratio $e(R_p)/e(O)$	Mean of ratio $e(R_b)/e(O)$	Index k of most common $G(a)$ (Table 2.1)
Slices 1 to 30: axial view of a human head: *192×256 pixels*	LL	0.59	0.34	2
	LH	0.76	0.46	2
	HL	0.74	0.48	2
	HH	0.71	0.40	2
Slices 31 to 60: axial view of a human spinal cord: *256×256 pixels*	LL	0.63	0.37	1
	LH	0.76	0.48	1
	HL	0.73	0.50	1
	HH	0.75	0.57	1
Slices 61 to 90: sagittal view of a human spinal cord: *512×512 pixels*	LL	N/A	0.41	{1,2,3}
	LH	N/A	0.53	{1,2,3}
	HL	N/A	0.51	{1,2,3}
	HH	N/A	0.49	{1,2,3}

$e(O)$: energy of original sub-band. $e(R_p)$: energy of residual data obtained by employing intra-band prediction and two areas of equal size. $e(R_b)$: energy of residual data obtained by employing block-based intra-band prediction. $G(a)$: spatial transformation on block a. N/A: not applicable.

2.4.2 Lossless compression performance

Our test data set consisted of fifty six MRI and four CT volumes. The characteristics of these 3D test images are summarized in columns 1 and 2 of Table 2.4. Volumes 1 to 12 comprise MRI slices (axial view) of a human head; volumes 13 to 16 comprise MRI slices (coronal view) of a human head; volume 17 comprises MRI slices (sagittal view) of a human spinal cord; and volume 18 comprises MRI slices (sagittal view) of a human knee. Volumes 19 to 38 comprise MRI slices (axial view) of a human head maintained by the Internet Brain Segmentation Repository (IBSR, http://www.cma.mgh.harvard.edu/ibsr/). Volumes 39 to 56 comprise MRI slices (coronal view) of a human head maintained by the IBSR. The test CT volumes comprise consecutive slices of the 'Visible Male' (volumes 59 and 60) and 'Visible Woman' (volumes 61 and 62) data sets maintained by the National Library of Medicine (NLM) [22].

Table 2.4 tabulates the lossless compression ratios and bit-rates (in bits per pixel) of four compression algorithms: JPEG2000 [23], 3DJPEG2000 [24], H.264/AVC intra-coding [25] and our proposed compression method. We specifically compared our compression method to these particular compression algorithms as they possess the qualities of scalability similar to our method. Moreover, the coding algorithms employed in JPEG2000, 3DJPEG2000, H.264/AVC intra-coding are very similar to those employed in the technical advances in lossless compression of 3D medical images reported in the literature [1-6]. JPEG2000 is an international standard for compression of still images which in 2001 was selected for inclusion in the Digital Imaging and Communications in Medicine (DICOM) standard for medical image compression. JPEG2000 employs the 2D discrete wavelet transform and provides functionalities such as lossless and lossy compression, quality and resolution scalability, ROI coding and robustness to bit errors. For lossless compression, JPEG2000 employs the 2D-IWT implemented using the lifting step scheme [23].

3D-JPEG2000 is the extension of JPEG2000 for compression of 3D images. 3D-JPEG2000 employs a discrete wavelet transform across the slices with the resulting transform slices being encoded using JPEG2000 [24].

Table 2-4: Lossless compression ratios and Bit-rates of 3D medical images using various compression methods

Modality- *slices: pixels per slice: bits per pixel*	**Compression method**			
	H.264/AVC intra-coding	JPEG2000	3D-JPEG2000	**Proposed method**
	compression ratio (bit-rate: bits per pixel)			
1. MRI-24:192×256:16	2.65:1 (6.03 bpp)	2.62:1 (6.10 bpp)	2.71:1 (5.90 bpp)	**3.25:1 (4.92 bpp)**
2. MRI-30:192×256:16	2.58:1 (6.20 bpp)	2.51:1 (6.33 bpp)	2.68:1 (5.97 bpp)	**3.21:1 (4.98 bpp)**
3. MRI-30:256×256:16	2.41:1 (6.63 bpp)	2.57:1 (6.22 bpp)	2.69:1 (5.94 bpp)	**3.28:1 (4.87 bpp)**
4. MRI-35:256×256:16	2.41:1 (6.63 bpp)	2.57:1 (6.22 bpp)	2.68:1 (5.97 bpp)	**3.29:1 (4.86 bpp)**
5. MRI-60:256×256:16	3.31:1 (4.83 bpp)	3.48:1 (4.59 bpp)	3.52:1 (4.54 bpp)	**3.86:1 (4.14 bpp)**
6. MRI-55:256×256:16	3.29:1 (4.86 bpp)	3.45:1 (4.63 bpp)	3.48:1 (4.59 bpp)	**4.01:1 (3.99 bpp)**
7. MRI-65:224×192:16	2.31:1 (6.92 bpp)	2.61:1 (6.13 bpp)	2.64:1 (6.06 bpp)	**3.03:1 (5.28 bpp)**
8. MRI-58:224×192:16	2.33:1 (6.86 bpp)	2.58:1 (6.20 bpp)	2.61:1 (6.13 bpp)	**2.92:1 (5.47 bpp)**
9. MRI-24:256×256:16	8.58:1 (1.86 bpp)	8.45:1 (1.89 bpp)	8.49:1 (1.88 bpp)	**9.49:1 (1.69 bpp)**
10. MRI-28:256×256:16	8.61:1 (1.85 bpp)	8.46:1 (1.89 bpp)	8.50:1 (1.88 bpp)	**9.52:1 (1.68 bpp)**
11. MRI-26:256×256:16	8.48:1 (1.88 bpp)	8.42:1 (1.90 bpp)	8.47:1 (1.89 bpp)	**9.44:1 (1.69 bpp)**
12. MRI-24:256×256:16	8.27:1 (1.93 bpp)	8.21:1 (1.94 bpp)	8.29:1 (1.93 bpp)	**9.20:1 (1.73 bpp)**
13. MRI-182:176×176:16	3.23:1 (4.95 bpp)	3.41:1 (4.69 bpp)	3.44:1 (4.65 bpp)	**3.75:1 (4.26 bpp)**
14. MRI-190:176×176:16	3.37:1 (4.74 bpp)	3.51:1 (4.55 bpp)	3.65:1 (4.38 bpp)	**3.87:1 (4.13 bpp)**
15. MRI-180:176×176:16	3.27:1 (4.89 bpp)	3.37:1 (4.74 bpp)	3.58:1 (4.46 bpp)	**3.70:1 (4.32 bpp)**
16. MRI-196:176×176:16	3.15:1 (5.07 bpp)	3.18:1 (5.03 bpp)	3.37:1 (4.74 bpp)	**3.65:1 (4.38 bpp)**
17. MRI-11:512×512:8	2.85:1 (2.80 bpp)	2.84:1 (2.81 bpp)	2.88:1 (2.77 bpp)	**3.10:1(2.58 bpp)**
18. MRI-50:512×512:8	2.93:1 (2.73 bpp)	2.91:1 (2.74 bpp)	2.96:1 (2.70 bpp)	**3.18:1 (2.51 bpp)**
19. MRI-61:256X256:16	4.12:1 (3.88 bpp)	4.17:1 (3.84 bpp)	4.21:1 (3.80 bpp)	**4.67:1 (3.43 bpp)**
20. MRI-63:256X256:16	3.13:1 (5.11 bpp)	3.17:1 (5.05 bpp)	3.21:1 (4.98 bpp)	**3.44:1 (4.65 bpp)**
21. MRI-63:256X256:16	3.16:1 (5.07 bpp)	3.20:1 (5.01 bpp)	3.27:1 (4.89 bpp)	**3.43:1 (4.66 bpp)**
22. MRI-61:256X256:16	3.80:1 (4.21 bpp)	3.81:1 (4.20 bpp)	3.81:1 (4.19 bpp)	**4.19:1 (3.82 bpp)**
23. MRI-60:256X256:16	4.07:1 (3.93 bpp)	4.12:1 (3.89 bpp)	4.18:1 (3.83 bpp)	**4.58:1 (3.49 bpp)**
24. MRI-63:256X256:16	3.85:1 (4.16 bpp)	3.89:1 (4.11 bpp)	3.93:1 (4.07 bpp)	**4.32:1 (3.70 bpp)**
25. MRI-60:256X256:16	2.53:1 (6.31 bpp)	2.56:1 (6.24 bpp)	2.59:1 (6.18 bpp)	**2.74:1 (5.83 bpp)**
26. MRI-63:256X256:16	3.01:1 (5.32 bpp)	3.05:1 (5.25 bpp)	3.10:1 (5.17 bpp)	**3.27:1 (4.90 bpp)**
27. MRI-63:256X256:16	3.04:1 (5.26 bpp)	3.08:1 (5.20 bpp)	3.12:1 (5.13 bpp)	**3.28:1 (4.88 bpp)**
28. MRI-63:256X256:16	3.01:1 (5.32 bpp)	3.05:1 (5.25 bpp)	3.08:1 (5.20 bpp)	**3.23:1 (4.96 bpp)**
29. MRI-63:256X256:16	3.05:1 (5.25 bpp)	3.08:1 (5.19 bpp)	3.11:1 (5.14 bpp)	**3.27:1 (4.89 bpp)**
30. MRI-60:256X256:16	3.84:1 (4.17 bpp)	3.75:1 (4.26 bpp)	3.78:1 (4.23 bpp)	**4.13:1 (3.87 bpp)**
31. MRI-60:256X256:16	4.10:1 (3.91 bpp)	4.15:1 (3.86 bpp)	4.38:1 (3.65 bpp)	**4.91:1 (3.26 bpp)**
32. MRI-63:256X256:16	3.88:1 (4.13 bpp)	3.89:1 (4.12 bpp)	3.96:1 (4.04 bpp)	**4.27:1 (3.75 bpp)**
33. MRI-63:256X256:16	3.34:1 (4.80 bpp)	3.38:1 (4.74 bpp)	3.42:1 (4.68 bpp)	**3.64:1 (4.40 bpp)**

Modality-*slices: pixels per slice: bits per pixel*	**Compression method**			
	H.264/AVC intra-coding	JPEG2000	3D-JPEG2000	**Proposed method**
	compression ratio (bit-rate: bits per pixel)			
34. MRI-63:256X256:16	3.00:1 (5.33 bpp)	3.04:1 (5.27 bpp)	3.09:1 (5.19 bpp)	**3.26:1 (4.91 bpp)**
35. MRI-63:256X256:16	3.01:1 (5.31 bpp)	3.05:1 (5.25 bpp)	3.08:1 (5.20 bpp)	**3.26:1 (4.91 bpp)**
36. MRI-63:256X256:16	3.14:1 (5.10 bpp)	3.18:1 (5.04 bpp)	3.20:1 (5.00 bpp)	**3.39:1 (4.72 bpp)**
37. MRI-63:256X256:16	3.12:1 (5.13 bpp)	3.06:1 (5.22 bpp)	3.19:1 (5.02 bpp)	**3.45:1 (4.64 bpp)**
38. MRI-63:256X256:16	3.05:1 (5.25 bpp)	3.08:1 (5.19 bpp)	3.11:1 (5.14 bpp)	**3.33:1 (4.81 bpp)**
39.MRI-128:256X256:16	7.58:1 (2.11 bpp)	7.67:1 (2.09 bpp)	7.84:1 (2.04 bpp)	**9.00:1 (1.78 bpp)**
40.MRI-128:256X256:16	6.71:1 (2.38 bpp)	6.79:1 (2.35 bpp)	6.95:1 (2.30 bpp)	**7.98:1 (2.01 bpp)**
41.MRI-128:256X256:16	8.07:1 (1.98 bpp)	8.17:1 (1.96 bpp)	8.53:1 (1.88 bpp)	**9.40:1 (1.70 bpp)**
42.MRI-128:256X256:16	7.81:1 (2.05 bpp)	7.91:1 (2.02 bpp)	8.18:1 (1.96 bpp)	**9.17:1 (1.74 bpp)**
43.MRI-128:256X256:16	7.89:1 (2.03 bpp)	7.99:1 (2.00 bpp)	8.16:1 (1.96 bpp)	**9.38:1 (1.71 bpp)**
44.MRI-128:256X256:16	8.28:1 (1.93 bpp)	8.39:1 (1.91 bpp)	9.13:1 (1.75 bpp)	**10.01:1 (1.60 bpp)**
45.MRI-128:256X256:16	9.03:1 (1.77 bpp)	9.14:1 (1.75 bpp)	9.87:1 (1.62 bpp)	**11.00:1 (1.45 bpp)**
46.MRI-128:256X256:16	8.32:1 (1.92 bpp)	8.42:1 (1.90 bpp)	9.31:1 (1.72 bpp)	**9.91:1 (1.61 bpp)**
47.MRI-128:256X256:16	8.57:1 (1.87 bpp)	8.68:1 (1.84 bpp)	9.14:1 (1.75 bpp)	**10.29:1 (1.55 bpp)**
48.MRI-128:256X256:16	8.41:1 (1.90 bpp)	8.52:1 (1.88 bpp)	9.27:1 (1.73 bpp)	**10.17:1 (1.57 bpp)**
49.MRI-128:256X256:16	6.15:1 (2.60 bpp)	6.22:1 (2.57 bpp)	6.69:1 (2.39 bpp)	**6.95:1 (2.30 bpp)**
50.MRI-128:256X256:16	8.39:1 (1.91 bpp)	8.50:1 (1.88 bpp)	8.82:1 (1.81 bpp)	**10.24:1 (1.56 bpp)**
51.MRI-128:256X256:16	8.18:1 (1.96 bpp)	8.28:1 (1.93 bpp)	8.82:1 (1.81 bpp)	**9.70:1 (1.65 bpp)**
52.MRI-128:256X256:16	7.83:1 (2.04 bpp)	7.93:1 (2.02 bpp)	8.73:1 (1.83 bpp)	**9.35:1 (1.71 bpp)**
53.MRI-128:256X256:16	7.58:1 (2.11 bpp)	7.68:1 (2.08 bpp)	8.73:1 (1.83 bpp)	**9.45:1 (1.69 bpp)**
54.MRI-128:256X256:16	7.45:1 (2.15 bpp)	7.54:1 (2.12 bpp)	8.62:1 (1.86 bpp)	**9.24:1 (1.73 bpp)**
55.MRI-128:256X256:16	6.72:1 (2.38 bpp)	6.80:1 (2.35 bpp)	7.73:1 (2.07 bpp)	**8.37:1 (1.91 bpp)**
56.MRI-128:256X256:16	6.84:1 (2.34 bpp)	6.92:1 (2.31 bpp)	7.80:1 (2.05 bpp)	**8.61:1 (1.86 bpp)**
57. CT-40:512×512:16	3.89:1 (4.11 bpp)	4.44:1 (3.60 bpp)	4.51:1 (3.54 bpp)	**5.34:1 (2.99 bpp)**
58. CT-40:512×512:16	3.81:1 (4.19 bpp)	4.39:1 (3.64 bpp)	4.49:1 (3.56 bpp)	**5.28:1 (3.03 bpp)**
59. CT-40:512×512:16	3.52:1 (4.92 bpp)	4.16:1 (3.84 bpp)	4.28:1 (3.73 bpp)	**4.50:1 (3.55 bpp)**
60. CT-40:512×512:16	3.68:1 (4.34 bpp)	4.12:1 (3.88 bpp)	4.19:1 (3.81 bpp)	**5.09:1 (3.14 bpp)**

MRI: magnetic resonance imaging. CT: computed tomography

H.264/AVC is an international standard for compression of video sequences. It is based on multi-frame motion compensation and estimation using variable block sizes. H.264/AVC includes a block-based intra-coding mode where still images are coded using only information contained in the image itself. H.264/AVC intra-coding uses selectable position-dependent linear combinations of neighboring sample values to form a prediction block [25].

In our proposed compression method we employed the Le Gall 5/3 wavelet filter implemented using the lifting step scheme to decompose each slice of the 3D test images with four levels of decomposition. We employed blocks of 16×16, 8×8 and 4×4 coefficients for intra-band prediction (with the block size and spatial transformation being selected by the encoder). We employed code-blocks of 32×32 samples for entropy

coding of the residual data. Our experiments on a large set of medical images have shown that code-blocks of 32×32 samples provide the best coding performance when employing blocks of 16×16, 8×8 and 4×4 coefficients for intra-band prediction.

For the case of JPEG2000 and 3D-JPEG2000, we employed lossless compression with four levels of decomposition and code-blocks of 32×32 coefficients for entropy coding. We employed code-blocks of 32×32 coefficients in this case to provide a fair comparison with our proposed method (note that we employed code-blocks of 32×32 samples in the proposed method). We used the Kakadu implementation of JPEG2000 and 3D-JPEG2000 [26]. For the case of H.264/AVC intra-coding, we employed blocks of 16×16, 8×8 and 4×4 pixels (with the block size and direction of prediction being selected by the encoder) and no quantization of the residual data. We used version 10.1 of the H.264/AVC reference software [27].

Results reported in Table 2.4 show that for all 3D test images, our method achieves the highest lossless compression ratios with an average improvement of 14% over JPEG2000 and H.264/AVC intra-coding and of 11% over 3D-JPEG2000. Even though JPEG2000 and 3D-JPEG2000 employ an entropy coding algorithm that encodes sub-bands without inter-band dependencies, the high number of edges in the test images results in high-frequency sub-bands with high energy content, which consequently affects the coding performance of the entropy coder. 3D-JPEG2000 performs better than JPEG2000 as it exploits the correlation between slices. H.264/AVC Intra-coding only employs the neighboring sample values for prediction of a block of pixels. This performs well in smooth images with few edges and transitions, which is not the case for most medical images.

Our proposed method decorrelates the data of each slice by applying a 2D-IWT. Based on the symmetrical content of the sub-bands, the energy of the sub-bands is then reduced by using block-based intra-band prediction. The result is a better lossless compression performance achieved by our modified EBCOT algorithm.

Table 2.5 shows the coding and decoding times (in seconds) of five 3D medical images when compressed using H.264/AVC intra-coding, JPEG2000, 3D-JPEG2000 and our proposed method on a 3.6 GHz Pentium IV processor. The proposed method reports the highest coding times as it performs an exhaustive search (based on SAD) amongst all previously coded blocks in the current sub-band to select the best predictor. This search is performed for each block at every resolution level. However, the decoding times are very similar to those of JPEG2000. Decoding times play a much more important role in medical image compression since after compression; medical images are usually stored and maintained by a database server, so a number of clients can access the datasets remotely. Coding times may be reduced by employing parallel processing architectures to encode various sub-bands simultaneously.

Table 2-5: Coding and decoding times (in seconds) of 3D medical images using various compression methods

Modality *slices: pixels per slice: bits per pixel*	**Compression method**							
	H.264/AVC intra-coding		JPEG2000		3D-JPEG2000		**Proposed method**	
	EcT	DcT	EcT	DcT	EcT	DcT	EcT	DcT
1.MRI: *24:192×256:16*	43.51	3.46	4.01	3.18	4.10	3.18	**48.23**	**3.23**
2.MRI: *35:256×256:16*	49.77	3.59	4.43	3.55	4.48	3.57	**56.12**	**3.74**
3.MRI: *11:512×512:8*	88.85	5.27	6.61	5.23	6.63	5.26	**95.12**	**5.24**
4.CT: *40:512×512:16*	321.36	18.75	24.30	18.82	24.31	18.83	**335.12**	**18.84**
5.CT: *40:512×512:16*	322.10	18.78	24.06	18.99	24.09	19.00	**349.67**	**19.06**

EcT: encoding time. DcT: decoding time
MRI: magnetic resonance imaging. CT: computed tomography.

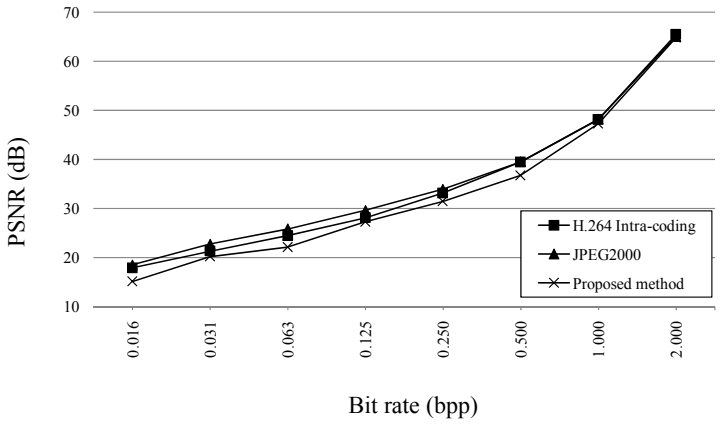
It is important to mention that the proposed compression method may be employed to compress any 3D medical image, independently of the imaging modality. However, it provides the best coding performance with structural data, such as MRI and CT data. When employed with 3D data where little structural information is depicted (i.e., little symmetrical content is depicted), the proposed method never increases the energy content of the data, thus avoiding any increase in the uncompressed bit-rate.

2.4.3 Scalable compression performance

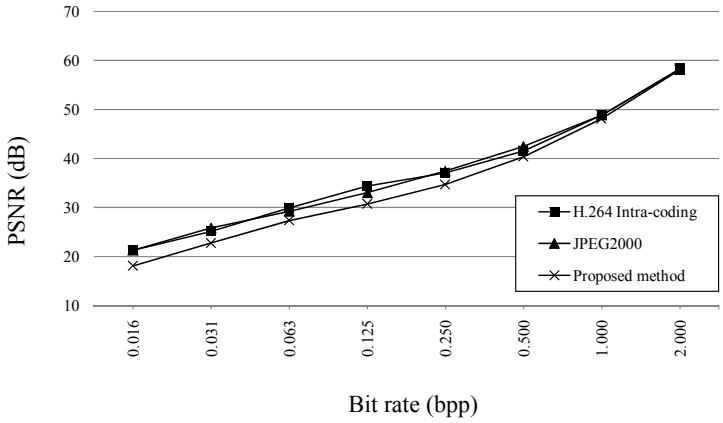
We evaluated the scalable compression performance of the proposed method on various slices of 3D medical image data with various orientations (acquisition views). Figure 2.8 plots the peak signal-to-noise ratio (PSNR) of the test slices after decoding at a variety of bit-rates. We compared our method against JPEG2000 and H.264/AVC intra-coding. For the case of our proposed method, we employed 15 quality layers and code-blocks of 32×32 samples to encode the residual data. JPEG2000 employs an entropy coding based on the EBCOT algorithm and thus allows the creation of a layered bit-stream [23]. In this case, we employed 15 quality layers and code-blocks of 32×32 coefficients to encode the test slices. For the case of H.264/AVC intra-coding, we encoded the test slices at the different bit-rates shown in Fig. 2.8.

Figure 2.8 shows that JPEG2000 and H.264/AVC intra-coding achieve PSNR values higher that those achieved by the proposed method, especially at very low bit-rates. This is expected, since in the proposed method the correct reconstruction of a sub-band depends on the correct reconstruction of all blocks after intra-prediction. If a reference block is not reconstructed at full quality, the reconstruction of subsequent blocks is thus affected. However, as the reconstruction quality of reference blocks improves (higher bit-rates), the reconstruction quality of subsequent blocks improves as well, resulting in a higher PSNR. Note that at bit-rates higher then 0.5bpp, the proposed method achieves PSNR values comparable to those achieved by JPEG2000 and H.264/AVC intra-coding.

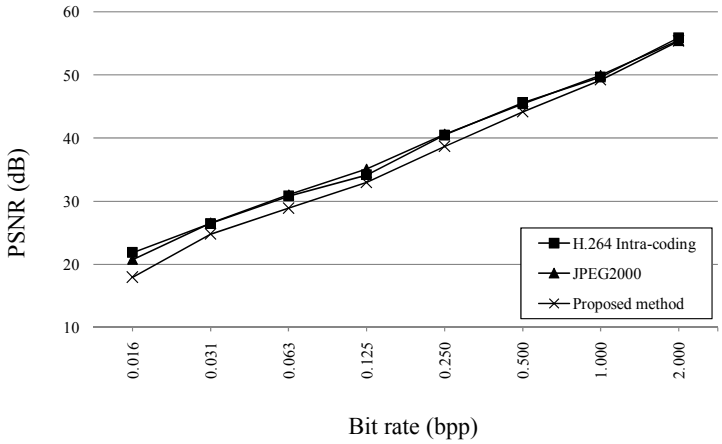
It is important to note that JPEG2000 provides random access to any region of an image. Our proposed method, although based on block-by-block entropy coding of sub-bands, does not provide random access to



a)



b)



c)

Fig. 2-8: PSNR values (in dB) of slices of medical image data decoded at various bit-rates after compression using different methods. A slice of an a) MRI volume of the axial view of a human head (256×192 pixels, 16 bits per pixel); b) an MRI volume of the sagittal view of a human spinal cord (512×512 pixels, 8 bits per pixel); and c) a CT volume of the axial view of a male body (512×512 pixels, 16 bits per pixel)

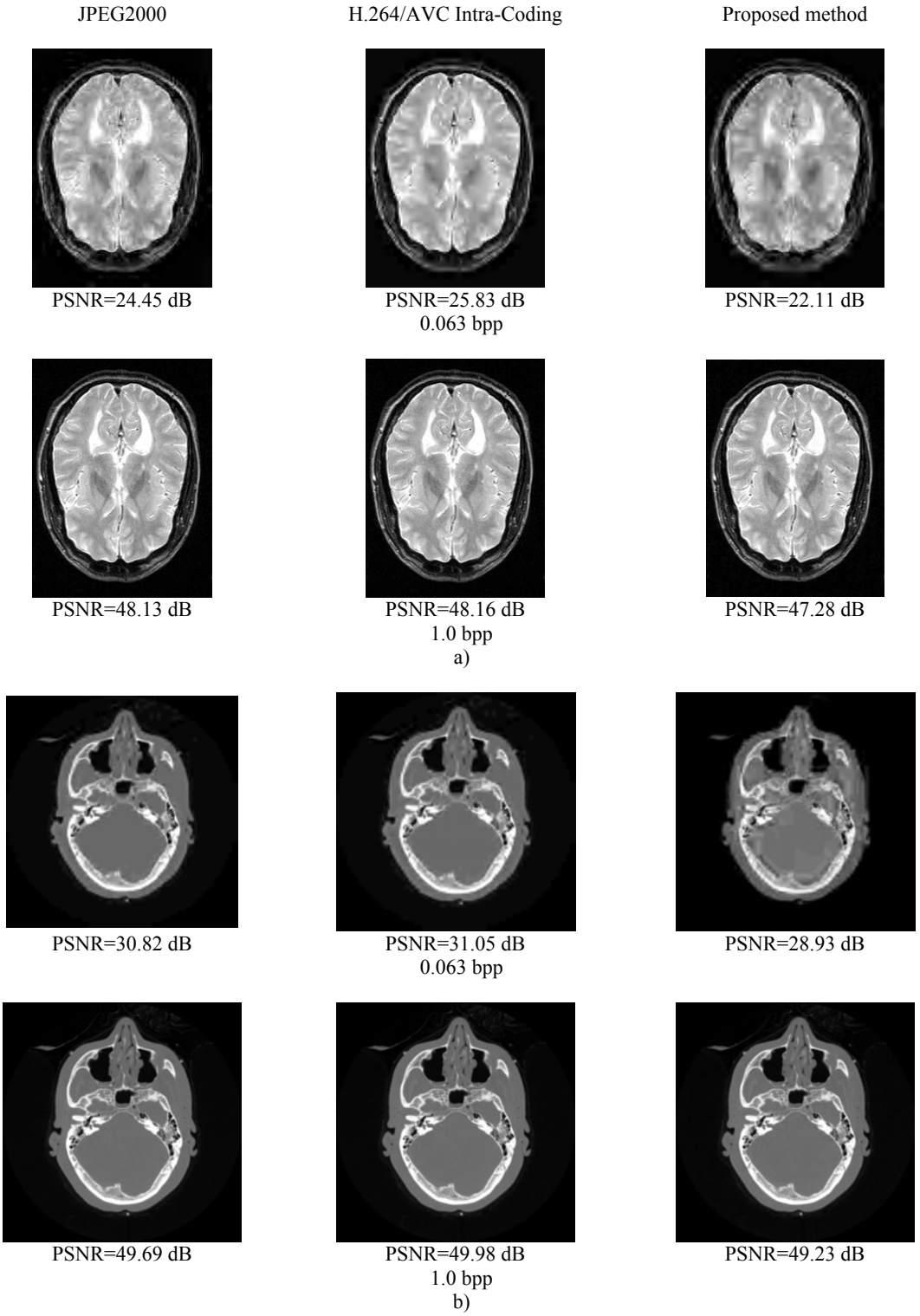


Fig. 2-9: Decoded slices of medical image data at different bit-rates after compression using JPEG2000, H.264/AVC Intra-coding and the proposed method. a) A slice of an MRI volume of the axial view of a human head (256×192 pixels, 16 bits per pixel). b) A slice of a CT volume of the axial view of a male body (512×512 pixels, 16 bits per pixel).

any region of an image, because the decoding of a block necessitates the decoding of the corresponding reference blocks.

Figure 2.9 illustrates two test slices reconstructed at different bit-rates using JPEG2000, H.264/AVC intra-coding and the proposed method. Note that at 1.0 bpp, the proposed method achieves PSNR values only 1.8% lower than those achieved by JPEG2000.

2.5 Conclusions

We presented a novel wavelet-based scalable lossless compression method for 3D medical image data. Data decorrelation is performed by a 2D integer wavelet transform applied on slices within the medical image volume. The resulting sub-bands are then compressed independently by first employing a block-based intra-band prediction method to reduce their energy, followed by a modified version of the EBCOT algorithm to achieve resolution and quality scalability. The novelty of our intra-band prediction method is in that it exploits anatomical symmetries within the structural data captured to predict the value of the wavelet coefficients on a block-by-block basis.

The performance of the proposed method was compared to that of other state-of-the-art methods that allow for scalability, including JPEG2000, H.264/AVC intra-coding and 3D-JPEG2000. Results show that the proposed method achieves an average improvement in lossless compression ratio of 14% over JPEG2000 and H.264/AVC intra-coding and of 11% over 3D-JPEG2000.

2.6 References

[1] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto and J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Trans. on Medical Imaging,* vol. 22, no. 3, pp. 441-458, March 2003.

[2] Z. Xiong, X. Wu, S. Cheng and J. Hua, "Lossy-to-lossless compression of medical volumetric images using three-dimensional integer wavelet transforms," *IEEE Trans. on Medical Imaging*, vol. 22, no. 3, pp. 459-470, March 2003.

[3] X. Wu and T Qiu, "Wavelet coding of volumetric medical images for high throughput and operability," *IEEE Trans. on Medical Imaging*, vol. 24, no. 6, pp. 719-727, June 2005.

[4] K. Krishnan. M. Marcellin, A. Bilgin and M. Nadar, "Efficient Transmission of Compressed Data for Remote Volume Visualization," *IEEE Trans. on Medical Imaging*, vol. 25, no. 9, pp. 1189-1199, September 2006.

[5] G. Menegaz and J.P. Thirian, "Three-dimensional encoding/two-dimensional decoding of medical data," *IEEE Trans. on Medical Imaging*, vol. 22, no. 3 , pp. 424-440, March 2003.

[6] R. Srikanth and A.G. Ramakrishnan, "Contextual Encoding in Uniform and Adaptive Mesh-Based Lossless Compression of MR Images," *IEEE Trans. on Medical Imaging*, vol. 24, no. 9, pp. 1199-1206, September 2005.

[7] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158-1170, July 2000.

[8] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol.41, pp. 909-996, 1998.

[9] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674-693, June 1989.

[10] S. Dewitte and J. Cornelis, "Lossless integer wavelet transform," *IEEE Signal Processing Letters*, vol. 4, no. 6, pp. 158-160, June 1997.

[11] A. R. Calderbank, I. Daubechies, W. Sweldens and B.L, Yeo, "Wavelet transforms that map integers to integers," *Appl. Comput. Harmon. Anal.*, vol. 5, no. 3, pp. 332-369, 1998.

[12] I. Daubechies and W. Sweldens, "Factoring wavelet transform into lifting steps," *J. Fourier Anal. Appl.*, vol. 41, no. 3, pp. 247-269, 1998.

[13] A. R. Calderbank, I. Daubechies, W. Sweldens and B.L, Yeo, "Lossless image compression using integer to integer wavelet transforms," *in Proc. Int. Conf. Image Procession (ICIP)*, 1997, pp. 569-599.

[14] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3462, Dec. 1993.

[15] A. Said and W. Peralman, "A new fast and efficient image coded based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, pp. 243-250, June 1996.

[16] E. J. Candes et al., "Ridgelets and their derivatives: representation of images with edges," *in Curves and Surfaces*, Schumaker et al., Eds. Nashville, TN: Vanderbilt University Press, 1999.

[17] E. J. Candes et al., "Curvelets – A surprisingly effective nonadaptive representation for objects with edges," *in Curves and Surfaces*, L. L. Schumaker et al., Eds. Nashville, TN: Vanderbilt University Press, 1999.

[18] A. Munteanu, J. Cornelis, G. Van der Auwera, and P. Cristea, "Wavelet image compression – The quadtree coding approach," *IEEE Trans. Inform. Technol. Biomed.*, vol. 3, pp. 176-185, September 1995.

[19] ——, "Wavelet-based lossless compression scheme with progressive transmission capability," *Int. Journal Imag. Syst. Technology*, vol. 10, pp. 76-85, January 1999..

[20] A. Munteanu, J. Cornelis, and P. Cristea, "Wavelet-based lossless compression of coronary angiographic images," *IEEE Trans. on Medical Imaging*, vol. 18, pp. 272-281, March 1999.

[21] R. Gallager and D. Van Voorhis, "Optimal source codes for geometrically distributed integer alphabets", *IEEE Transactions on Information Theory*, vol. 21, pp. 228-230, March 1975.

[22] The National Library of Medicine (NLM): http://www.nlm.nih.gov

[23] JPEG2000 Verification Model 5.0 (Technical Description), ISO/IEC JTC1/SC29/WG1 N1420, 1999.

[24] Information Technology—JPEG 2000 Image Coding System—Part 2: Extensions, ISO/IEC 15 444-2, 2002.

[25] G. Sullivan, P. Topiwala and A. Luthra, "The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions", *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 5558, pp. 454-474, August 2004.

[26] Available online: http://www.kakadusoftware.com

[27] Available online: http://iphome.hhi.de/suehring/tml

Chapter 3

3. 3D Scalable Medical Image Compression with Optimized Volume of Interest Coding[1]

3.1 Introduction

The wide pervasiveness of medical imaging applications in healthcare settings and the increased interest in telemedicine technologies, it has become essential to reduce both storage and transmission bandwidth requirements needed for archival and communication of related data, preferably by employing lossless compression methods. Furthermore, providing random access as well as resolution and quality scalability to the compressed data has become of great utility. Random access refers to the ability to decode any section of the compressed image without having to decode the entire data set. Resolution and quality scalability, on the other hand, refers to the ability to decode the compressed image at different resolution and quality levels, respectively. The latter is especially important in interactive telemedicine applications, where clients (e.g., radiologists or clinicians) with limited bandwidth connections using a remote image retrieval system may connect to a central server to access a specific region of a compressed 3D data set, i.e., a volume of interest (VOI). The 3D image is then transmitted progressively within the VOI from an initial lossy to a final lossless representation.

Several compression methods for 3D medical images have been proposed in the literature, some of which provide resolution and quality scalability up to lossless reconstruction [1]-[6]. These methods are based on the discrete wavelet transform (DWT), whose inherent properties produce a bit-stream that is resolution-scalable. Quality scalability is then achieved by employing bit-plane based entropy coding algorithms that exploit the dependencies between the location and value of the wavelet coefficients, such as the embedded zerotree wavelet coding (EZW), the set partitioning in hierarchical trees (SPIHT), and the embedded block coding with optimized truncation (EBCOT) algorithms [7]-[9]. These compression methods, however, do not provide VOI decoding capabilities, i.e., the ability to reconstruct a VOI at higher quality than the rest of the 3D image.

Recently, a number of medical image compression methods that support VOI coding have been proposed [10]-[13]. In [10], the authors presented a compression method based on JPEG2000 that supports prioritized VOI coding based on the anatomical tissues depicted in a 3D medical image. The method employs

[1] A version of this chapter has been accepted for publication. V. Sanchez, P. Nasiopoulos, and R. Abugharbieh, "3D Scalable Medical Image Compression with Optimized Volume of Interest Coding," *IEEE Transactions on Medical Imaging*, May 2010.

a one-dimensional DWT (1D-DWT) along the slice direction with JPEG2000 encoding of the resulting transform slices. A priority is assigned to each group of coefficients describing the same spatial region at the same decomposition level according to its intensity level in the spatial domain. The method also allows for the definition of the relative importance of each sub-band in the coding process. In [11], the authors introduced a 3D medical image compression technique that supports VOI coding based on 3D sub-band block hierarchical partitioning (3D-SBHP), a highly scalable wavelet transform based entropy coding algorithm. A number of parameters that affect the effectiveness of VOI coding are studied, including the size of the VOI, the number of decomposition levels, and the target bit-rate. The authors also discussed an approach to optimize VOI decoding by assigning a decoding priority to the different wavelet coefficient bit-planes. In [12], the authors summarized the features of various methods for VOI coding, including the maximum shift (MAXSHIFT) and general scaling-based (GSB) methods supported by the JPEG2000 standard [14]. These particular methods scale up the coefficients associated with a VOI above the background coefficients, by a scaling value. The MAXSHIFT method employs a maximum scaling value so that VOI coefficients are completely decoded before any background coefficients. The GSB method, on the other hand, employs a lower scaling value so that VOI and background coefficients are decoded simultaneously. In [13], the authors presented a VOI coding method for volumetric images based on the GSB method and the shape-adaptive wavelet transform. The method extends the capabilities of the GSB method to 3D images with arbitrarily-shaped VOIs and allows for coding partial background information in conjunction with the VOI.

The main objective of this chapter is to present a 3D medical image compression method with a) scalability properties, by quality and resolution up to lossless reconstruction; and b) optimized VOI coding at any bit-rate. We are particularly interested in interactive telemedicine applications, where different remote clients with limited bandwidth connections may request the transmission of different VOIs of the same compressed 3D image stored on a central server. In this particular scenario, it is highly desirable to progressively transmit the different VOIs without the need to recode the entire 3D image for each client's request. Furthermore, in order to improve the client's experience in visualizing the data remotely, it is also desirable to transmit the VOI at the highest quality possible at any bit-rate, in conjunction with a low quality version of the background, which is important in a contextual sense to help the client observe the position of the VOI within the original 3D image [15]-[17]. In this work, the VOI is a cuboid defined in the spatial domain with possibly different values for the length, width and height.

The method presented in this paper employs a 3D integer wavelet transform (3D-IWT) and a modified EBCOT with 3D contexts to compress the 3D medical imaging data into a layered bit-stream that is scalable by quality and resolution, up to lossless reconstruction. VOI coding capabilities are attained after compression by employing a bit-stream reordering procedure, which is based on a weighting model that incorporates the position of the VOI and the mean energy of the wavelet coefficients. In order to attain optimized VOI coding at any bit-rate, the proposed method also employs after compression, an optimization technique that maximizes the reconstruction quality of the VOI, while allowing for the decoding of background information

with peripherally increasing quality around the VOI. The proposed method is different from the method in [10], where the VOI coding procedure is tissue-based, the relative importance of a specific sub-band is empirically assigned, and the entropy coding of wavelet coefficients is performed using 2D contexts. Our proposed method is also different from the VOI coding method proposed in [11], where the background information is only decoded after the VOI is fully decoded, which prevents observing the position of the VOI within the original 3D image. The proposed method also differs from the method in [13], where the scaling value of the VOI coefficients is empirically assigned and the shape information of the VOI must be encoded and transmitted, which may result in an increase in computational complexity as well as bit rate (due to shape encoding).

The novelties of the proposed method are threefold. First, our method employs the 3D-IWT in conjunction with a modified EBCOT with 3D contexts to exploit redundancies between slices and improve the coding performance, while at the same time creating a layered bit-stream that is scalable by resolution and quality up to lossless reconstruction. Second, the bit-stream reordering procedure is performed after encoding, thus allowing for the decoding of any VOI without the need to recode the entire 3D image. Third, the background information that is decoded in conjunction with the VOI allows for placement of the VOI into the context of the 3D image and enhances the visualization of the data at any bit-rate.

We test the performance of the proposed method on various real 3D medical images and compare it to 3D-JPEG2000 with VOI coding, using the MAXSHIFT and the GSB methods. Performance evaluation results show that, at various bit-rates, the proposed method achieves a higher reconstruction quality, in terms of the peak signal-to-noise ratio (PSNR), than those achieved by the MAXSHIFT and GSB methods.

The remainder of the chapter is organized as follows. In Section 3.2, we describe the proposed compression method. In Section 3.3, we present and discuss the experimental results. We give the concluding remarks in Section 3.4.

3.2 Proposed Compression Method

The proposed compression method is depicted in Fig. 3.1. We first apply a 3D-IWT with dyadic decomposition to an input 3D medical image. This transform maps integers to integers and allows for perfect invertibility with finite precision arithmetic, which is required for perfect reconstruction of a signal [18]. In this work, we employ the bi-orthogonal Le Gall 5/3 wavelet filter, implemented using the lifting step scheme [19]. Each level of decomposition, r, of the transform decomposes the 3D image input into eight 3D frequency sub-bands denoted as LLLr, LLHr, LHLr LHHr, HLLr, HLHr, HHLr, and HHHr. The approximation low-pass sub-band, LLL, is a coarser version of the original 3D image, whereas the other sub-bands represent the details of the image. The decomposition is iterated on the approximation low-pass sub-band.

We then group the wavelet coefficients into 3D groups and compute the mean energy of each group.

We encode each group of coefficients independently using a modified EBCOT with 3D contexts to create a separate scalable layered bit-stream for each group. The coordinates of the VOI in the spatial domain, in conjunction with the information about the mean energy of the grouped coefficients, are then used in a weight assignment model to compute a weight for each group of coded wavelet coefficients. These weights are used to reorder the output bit-stream and create an optimized scalable layered bit-stream with VOI decoding capabilities and gradual increase in peripheral quality around the VOI. At the decoder side, the wavelet coefficients are obtained by applying the EBCOT decoder. Finally, an inverse 3D-IWT is applied to obtain the reconstructed 3D image. The decoder can also truncate the received bit-stream to obtain a 3D image at any bit-rate.

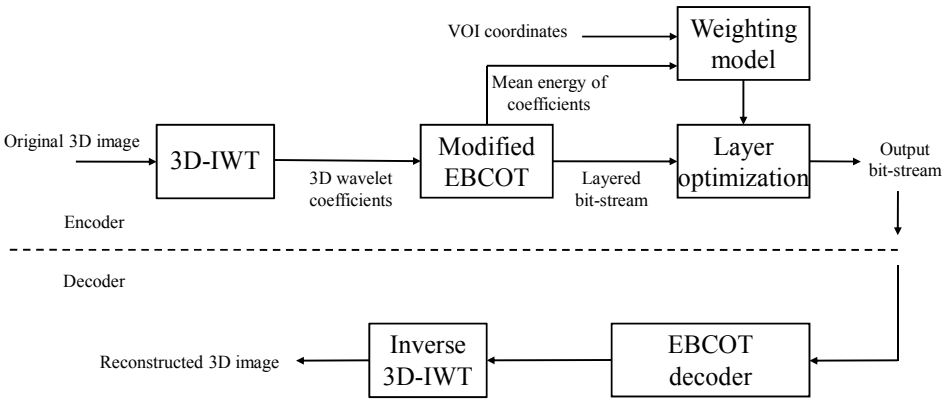


Fig. 3-1: Block diagram of the proposed scalable lossless compression method. 3D-IWT: three-dimensional integer wavelet transform. EBCOT: embedded block coding with optimized truncation.

It is important to mention that the proposed method attains VOI decoding capabilities after the 3D medical imaging data is coded. This is particularly advantageous in interactive telemedicine applications, where different clients may request different VOIs of the same compressed 3D image stored on a central server. The server may then transmit different versions of the same compressed bit-stream by simply performing the bit-stream reordering procedure for each requested VOI, thus saving time in recoding the entire 3D image for each client's request. Moreover, if a client requests a different VOI while transmission of a compressed bit-stream is taking place, the server only needs to update the coefficient weights according to the newly requested VOI and reorder the un-transmitted portion of bit-stream, which also saves time in recoding and retransmitting the entire 3D image. Note that the bit-stream reordering procedure can take place before transmission since the decoder is capable of decoding any bit-stream regardless of the order it is transmitted (due to the fact that code-cubes are encoded independently). Alternatively, the bit-stream reordering procedure may also be performed at the client side once the image has been fully transmitted. In this particular scenario, the main advantage of the proposed method lies on saving time in recoding the entire 3D image for different VOIs.

There are three key techniques in the proposed compression method. The first is the modified EBCOT. The second is the weight assignment model. The last is the creation of an optimized scalable layered bit-stream. We will discuss them in the next subsections.

3.2.1 Modified EBCOT

It is possible to employ EBCOT to code the wavelet coefficients on a slice-by-slice basis. However, in our compression method, the input samples to the entropy coding algorithm are 3D-IWT wavelet coefficients rather than 2D-IWT wavelet coefficients. Therefore, coding 3D-IWT wavelet coefficients on a slice-by-slice basis makes EBCOT less efficient since the correlation between coefficients is not exploited in three dimensions. Consequently, a modified EBCOT algorithm is needed to overcome this problem, which we solve by partitioning each 3D sub-band into small 3D groups of samples (i.e., wavelet coefficients), which we call code-cubes, and coding each code-cube independently by using a modified EBCOT with 3D contexts.

In this work, code-cubes are comprised of $a \times a \times a$ samples and describe a specific region of the 3D image at a specific decomposition level. We employ a pyramid approach to define the size of code-cubes across the different decomposition levels. In this approach, a code-cube of size $a \times a \times a$ samples and position $\{x,y,z\}$ at decomposition level r is related to a code-cube of size $a/2 \times a/2 \times a/2$ samples and position $\{x,y,z\}$ at decomposition level $r + 1$, where $r = 1$ is the first decomposition level. Fig. 3.2 shows the 3D-IWT sub-bands of a 3D image after two levels of decomposition in all three dimensions with a single code-cube in sub-bands HHH_2 and HHH_1. It can be seen that by employing a pyramid approach to define the size of code-cubes, it is possible to access any region of the 3D image at any resolution, which is essential for VOI coding. In this work, we limit the code-cube dimension, a, to be a power of 2, with $a \geq 2^3$.

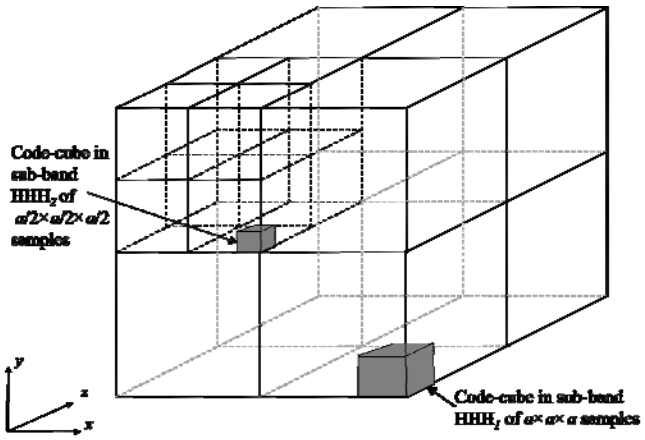


Fig. 3-2: 3D-IWT sub-bands of a 3D image after two levels of decomposition in all three dimensions with a single code-cube in sub-bands HHH_1 and HHH_2.

We code each code-cube independently using a modified EBCOT with 3D contexts that exploit inter-

slice correlations. Coding wavelet coefficients by extending 2D context modeling to 3D has been extensively used to improve coding efficiency [1, 2, 20, 21]. Here, we propose a 3D context model, based on the four coding passes previously discussed, that incorporates information from the immediate horizontal, vertical, diagonal and temporal neighbors of sample *c* located in slices *z*, *z*−1 and *z*+1, as illustrated in Fig. 3.3.

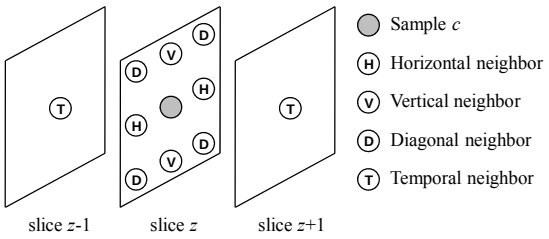


Fig. 3-3: The immediate horizontal, vertical, diagonal and temporal neighbors of sample *c* located in slices *z*, slices $z-1$ and $z+1$.

During the ZC pass, we code whether or not sample *c* becomes significant in the current bit-plane *p*. As explained by Taubman in [9], the significance of sample *c* is highly dependent upon the value of its immediate horizontal, vertical and diagonal neighbors. Here, in order to exploit inter-slice correlations, we also employ the information about the significance of the immediate temporal neighbors to code the significance of sample *c*. Let *h* denote the number of significant horizontal neighbors, with $0 \le h \le 2$. Let *v* denote the number of significant vertical neighbors, with $0 \le v \le 2$. Similarly, let *d* denote the number of significant diagonal neighbors, with $0 \le d \le 4$; and let *t* denote the number of significant temporal neighbors, with $0 \le t \le 2$. The proposed 3D context assignment for the ZC pass is summarized in Table 3.1. Note that this 3D context assignment emphasizes on the neighbors which are expected to present the strongest correlation in a particular sub-band. For example, we expect the strongest correlation amongst horizontally adjacent samples in sub-bands LLL, LLH, LHL and LHH; therefore, the proposed 3D context assignment emphasizes on horizontal neighbors for these sub-bands.

For the SC pass, we expect that the sign information of sample *c* exhibit some correlation with that of its temporal neighbors, in addition to the correlation exhibited with its vertical and horizontal neighbors, as explained in [9]. Therefore, in this pass, we employ the sign and significance information of the temporal, vertical and horizontal neighbors to code the sign information of sample *c*. Let $X(c)$ denote the sign bit of sample *c*, so that $X(c)=1$ if $c \ge 0$; otherwise $X(c)=0$. Let h_s denote the sign information of the horizontal neighbors, with $h_s = 0$ if both horizontal neighbors are insignificant or both are significant with different sign, $h_s = 1$ if at least one horizontal neighbor is positive, and $h_s = -1$ if at least one horizontal neighbor is negative. Let us define v_s and t_s in a similar fashion for the sign information of the vertical and temporal neighbors, respectively. The proposed 3D context assignment for the SC pass is summarized in Table 3.2. Note that this 3D context assignment exploits the fact that the distribution of $X(c)$ given any particular neighborhood should be identical to the distribution of $-X(c)$, given the dual neighborhood with the signs of all neighbors

reversed. The binary valued symbol that is coded with respect to the corresponding context is $X(c) \oplus \hat{X}(c)$, where $\hat{X}(c)$ is an auxiliary variable that indicates the sign prediction under a given context.

Table 3-1: Proposed 3D context assignment for the Zero Coding (ZC) pass of sample c_z

Sub-bands LLL, LLH, LHL, LHH					Sub-bands HLL, HLH					Sub-bands HHH, HHL			
h	*v*	*d*	*t*	Context	*h*	*v*	*d*	*t*	Context	*d*	*h* + *v*	*t*	Context
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	≥1	0	1	0	0	≥1	0	1	0	1	0	1
0	≥1	X	0	2	≥1	0	X	0	2	0	≥2	0	2
0	X	X	1	3	X	0	X	1	3	0	X	1	3
0	X	X	2	4	X	0	X	2	4	0	X	2	4
1	0	0	0	5	0	1	0	0	5	1	0	0	5
1	0	≥1	0	6	0	1	≥1	0	6	1	≥1	0	6
1	≥1	X	0	7	≥1	1	X	0	7	1	X	≥1	7
1	X	X	≥1	8	X	1	X	≥1	8	2	0	0	8
2	X	X	0	9	X	2	X	0	9	2	≥1	0	9
2	X	X	≥1	10	X	2	X	≥1	10	2	X	≥1	10

X: don't care

For the MR pass, we also expect that the magnitude of sample *c* exhibit some correlation with the magnitude of its immediate temporal neighbors. We thus employ the significance information of the immediate temporal neighbors, in addition to the significance information of the immediate horizontal and vertical neighbors, to code the magnitude of sample *c*. Let *S* denote the total number of significant temporal, horizontal and vertical neighbors of sample *c*, with $0 \le S \le 6$. Let σ be a variable that transitions from 0 to 1 after sample *c* is found to be significant for the for first time; i.e., after the MR pass is first applied to sample *c*. The proposed 3D context assignment for the MR pass is summarized in Table 3.3.

Note that for each coding pass, the coding engine maintains a look-up table in order to identify the probability model to be used by the adaptive arithmetic coder under each context.

Table 3-2: Proposed 3D context assignment for the Magnitude Refinement (MR) pass of sample c_z

σ	*S*	Context
0	0	0
0	{1,2}	1
0	{3,4}	2
0	{5,6}	3
1	X	4

X: don't care

Table 3-3: Proposed 3D context assignment for the Sign Coding (SC) pass of sample c_z

h_s	v_s	t_s	$\hat{X}(c)$	Context
1	1	1	1	0
1	1	{-1,0}	1	1
1	0	1	1	2
1	0	{-1,0}	1	3
1	-1	1	1	4
1	-1	{-1,0}	0	5
0	1	1	1	6
0	1	{-1,0}	1	7
0	0	X	1	8
0	-1	{-1,0}	0	7
0	-1	1	1	6
-1	1	{-1,0}	0	5
-1	1	1	0	4
-1	0	{-1,0}	0	3
-1	0	1	0	2
-1	-1	{-1,0}	0	1
-1	-1	1	0	0

X: don't care

3.2.2 Weight assignment model

The purpose of the weight assignment model is to enable the encoder to reorder the output bit-stream, so that the code-cubes that constitute the VOI are included earlier while allowing for gradual increase in peripheral quality around the VOI, under the constraint that the VOI is the main focal point. Techniques that allow gradual increase in peripheral quality around a focal point have been extensively used to improve image and video coding algorithms [22]-[25]. In the proposed compression method, we apply this technique to decode contextual background information with peripherally increasing quality around the VOI, which in turn enhances the visualization of the data at any bit-rate. We achieve this by considering two main factors: 1) the proximity of a code-cube to the VOI, and 2) the mean energy of a code-cube. The desired weight assignment for code-cube Cc_i is a function of the form:

$$w_{Cc_i}(\mathrm{P}_{Cc_i}, \mathrm{B}_{Cc_i}, \rho_{Cc_i}) \quad \in [0,1] \tag{3.1}$$

where P_{Cc_i} is a value in the range (0,1] that depends on the proximity between the center of code-cube Cc_i and

the center of the VOI, B_{Cc_i} is a value in the range [0,1] that depends on the mean energy of code-cube Cc_i, and ρ_{Cc_i} is a value in the range [0,1] that depends on the proportion of wavelet coefficients of code-cube Cc_i that contributes to the VOI. We define function $w_{Cc_i}(P_{Cc_i}, B_{Cc_i}, \rho_{Cc_i})$ by studying an important feature of 3D medical images in the spatial and wavelet domains.

In the spatial domain, 3D medical images usually depict the anatomy of one or more organs (or structures) over an empty background. Furthermore, the areas comprised by the depicted structures typically contain most of the energy of the 3D image.

In the wavelet domain, the original structures depicted in the 3D medical image are preserved as edge information within each sub-band. Following the grouping of wavelet coefficients into code-cubes, those code-cubes comprising the edge information thus tend to contain most of the sub-band energy.

Based on the above observations, we employ the information about the coordinates of the VOI and the mean energy of a code-cube to determine if a code-cube constitutes the VOI, the non-empty background, i.e., a structure depicted in the 3D medical image that is not part of the VOI, or the empty background. The main objective is to assign the largest weight ($w_{Cc_i} = 1$) to those code-cubes within the VOI, a smaller weight to those code-cubes within the non-empty background, and the smallest weight to those code-cubes within the empty background.

We determine which code-cubes constitute the VOI by using the VOI coordinate information and the location of the code-cubes in the spatial domain. The latter is calculated by tracing back the wavelet coefficients to a set of voxels using the footprint of the wavelet kernel used to transform the data. We employ $\rho_{Cc_i} \in [0,1]$ as a measure of the proportion of wavelet coefficients of code-cube Cc_i that contribute to the VOI, with $\rho_{Cc_i} = 0$ for those code-cubes outside the VOI, $\rho_{Cc_i} = 1$ for those code-cubes that fully contribute to the VOI, and $0 < \rho_{Cc_i} < 1$ for those code-cubes with some contribution to the VOI.

In order to determine which code-cubes constitute the empty background, we use the information about their mean energy, which for code-cube Cc_i is calculated as follows:

$$\hat{\varepsilon}_{Cc_i} = \frac{1}{K}\sum_{k=1}^{K} {c_k}^2 \tag{3.2}$$

where c_k is the kth sample of Cc_i, and K is the total number of samples in Cc_i.

We expect the value of $\hat{\varepsilon}_{Cc_i}$ to be zero for those code-cubes within the empty background. However, this may not always be true due to the discrete size of code-cubes and the smearing effects of the wavelet filter, which may result in some code-cubes comprising both structure and empty background information. The simplest possible method to determine if a code-cube is part of the empty background is to use a thresholding approach, where code-cube Cc_i is considered to constitute the empty background if the mean

energy $\hat{\varepsilon}_{Cc_i}$ is below a defined value. Generally, the use of continuous functions, where no hard decision is required to determine if a code-cube is part of the empty background, leads to better results. We, thus, use the following simple continuous, monotonically decreasing function to determine if code-cube Cc_i in sub-band s is part of the empty background:

$$\mathrm{B}_{Cc_i} = 1 - \frac{\hat{\varepsilon}_{Cc_i}}{\max\limits_{Cc_i \in s}\{\hat{\varepsilon}_{Cc_i}\}} \quad \in [0,1] \tag{3.3}$$

where $\max\limits_{Cc_i \in s}\{\hat{\varepsilon}_{Cc_i}\}$ is the maximum mean energy $\hat{\varepsilon}_{Cc_i}$ in sub-band s. A value of B_{Cc_i} close to one means a high probability that code-cube Cc_i is part of the empty background, corresponding to a low mean energy content, whereas a value of B_{Cc_i} close to zero means a low probability that code-cube Cc_i is part of the empty background, corresponding to a high mean energy content. All values B_{Cc_i} are calculated during the encoding process and are stored as header information.

We now define function $w_{Cc_i}(\mathrm{P}_{Cc_i}, \mathrm{B}_{Cc_i}, \rho_{Cc_i})$ to assign weight w_{Cc_i} to code-cube Cc_i. We also employ a continuous, monotonically decreasing function with a range [0,1] as follows:

$$w_{Cc_i}(\mathrm{P}_{Cc_i}, \mathrm{B}_{Cc_i}, \rho_{Cc_i}) = \rho_{Cc_i} + \left(1 - \rho_{Cc_i}\right) e^{-\left(\frac{\mathrm{B}_{Cc_i}}{\mathrm{P}_{Cc_i}}\right)^2} \tag{3.4}$$

where B_{Cc_i} is as defined in (3.3), ρ_{Cc_i} is the proportion of wavelet coefficients of code-cube Cc_i that contributes to the VOI, and P_{Cc_i} is the probability that code-cube Cc_i is located peripherally close to the VOI and is calculated by:

$$\mathrm{P}_{Cc_i} = 1 - \frac{d_{Cc_i}}{D_{max}} \quad \in (0,1] \tag{3.5}$$

where d_{Cc_i} is the radial distance between the center of the VOI and the center of the region represented by code-cube Cc_i in the spatial domain, and D_{max} is the maximum radial distance in the spatial domain between two samples of the 3D image:

$$D_{max} = \sqrt{x^2 + y^2 + z^2} \tag{3.6}$$

where $\{x,y,z\}$ denotes the size of the 3D image in the spatial domain. Note that P_{Cc_i} may only take values in the range (0, 1], since $0 < d_{Cc_i} < D_{max}$ for code-cubes outside the VOI with a dimension $a \geq 2^3$. A value of P_{Cc_i} close to one means a high probability that code-cube Cc_i is located peripherally close to the VOI, whereas a value of P_{Cc_i} close to zero means a low probability that code-cube Cc_i is located peripherally close to the VOI. We employ the function in (4) as it is one of the simplest functions to provide the desired gradual decrease in weight w_{Cc_i} that quickly falls off as the probability that code-cube Cc_i is part of the empty background increases (i.e., as the value of B_{Cc_i} increases) and the probability that is located peripherally close to the VOI decreases (i.e., as the value of P_{Cc_i} decreases), but still leads to weights equal to one for those code-cubes within the VOI (i.e., with a value of ρ_{Cc_i} equal to one). Function $w_{Cc_i}(\mathrm{P}_{Cc_i}, \mathrm{B}_{Cc_i}, \rho_{Cc_i})$ has a simple probabilistic interpretation. The value of weight w_{Cc_i} corresponds to the probability of code-cube Cc_i being within the VOI and thus containing structure information. For code-cubes that fully contribute to the VOI, this probability is equal to one since the underlying assumption is that all code-cubes within the VOI contain structure information. For code-cubes outside the VOI, this probability follows a Gaussian distribution with a peak value of one centered at $\mathrm{B}_{Cc_i} = 0$ and a decaying rate controlled by P_{Cc_i}. For code-cubes that partially contribute to the VOI, this probability depends on their proportion of wavelet coefficients that contribute to the VOI, and a Gaussian distribution controlled by B_{Cc_i} and P_{Cc_i}. Figure 3.4 shows the plot of (4) for code-cubes outside the VOI for various values of P_{Cc_i}. It can be seen that the value of w_{Cc_i} slowly decays peripherally around the center of the VOI for small values of B_{Cc_i} and large values of P_{Cc_i}, whereas it quickly approaches zero for large values of B_{Cc_i} and small values of P_{Cc_i}.

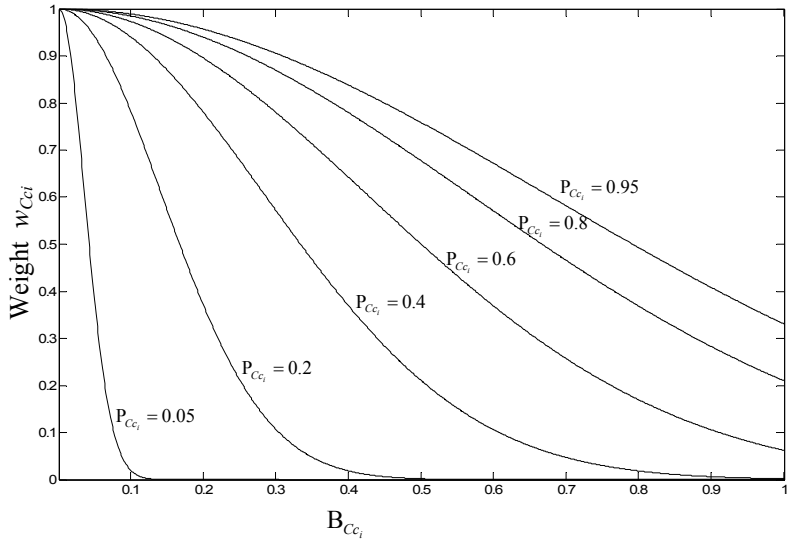


Fig. 3-4: Weight assignment for code-cube Cc_i according to B_{Cc_i}, its probability of being part of the empty background, for various values of P_{Cc_i}, its probability of being located peripherally close to the VOI.

Note that after the image is coded, the calculation of the code-cube weights for any VOI requires only the re-computation of two values for each code-cube, 1) its probability of being peripherally close to the VOI (i.e., value P_{Cc_i}), and 2) its contribution to the VOI (i.e., value ρ_{Cc_i}). There is no need to re-compute the code-cube probabilities of being within the empty background (i.e., values B_{Cc_i}), since these probabilities are independent of the VOI and are calculated only once during the coding process (values B_{Cc_i} are stored as header information).

3.2.3 Creation of an optimized scalable layered bit-stream

The bit-stream of each code-cube Cc_i may be independently truncated to any of a collection of different lengths, due to the entropy coding process, which is performed using a number of coding passes. We organize these truncated bit-streams into a number of quality layers to create a scalable layered bit-stream. This is done by collecting the incremental contributions from the various code-cubes into the quality layers such that the code-cube contributions result in a rate-distortion optimal representation of the 3D image, for each quality layer L. The code-cube incremental contributions into each quality layer are stored as header information during the coding process.

After the creation of the scalable layered bit-stream, the main objective is to reorder this bit-stream, so that the code-cubes that constitute the VOI are included earlier in conjunction with contextual background information. We achieve this by finding the optimal collection of truncated bit-streams that minimizes the overall distortion of the reconstructed 3D image at quality layer L, while attaining VOI decoding capabilities. We incorporate weight w_{Cc_i}, as calculated in Section 3.2.1, in the optimization process to achieve a gradual

increase in peripheral quality around the VOI.

In this work, we employ the Mean Square Error (MSE) to quantify the distortion of code-cube Cc_i at quality layer L:

$$\mathrm{M}_{Cc_i}^{L} = \frac{1}{K}\sum_{k=1}^{K}(c_k - \hat{c}_k)^2 \tag{3.7}$$

where c_k is the kth sample of Cc_i, $\hat{c}_k$ is the quantized representation of the kth sample of Cc_i associated with the truncated bit-stream at quality layer L, and K is the total number of samples in Cc_i. The MSE is easily calculated by using the information about the code-cube contributions into quality layer L stored as header information during the coding process.

The MSE of code-cube Cc_i at quality layer L in sub-band s on a per-voxel basis over the entire 3D image may then be calculated as:

$$\overline{\mathrm{M}}_{Cc_i}^{L} = \frac{g_s}{N_s}\frac{q_s}{Q}\mathrm{M}_{\mathrm{Cc_i}}^{L} = 2^{2r}\frac{g_s}{N_s}\mathrm{M}_{\mathrm{Cc_i}}^{L} \tag{3.8}$$

where Q is the total number of image voxels, r is the decomposition level to which Cc_i belongs (r = 1 corresponds to the first decomposition level), $q_s = Q/2^{2r}$ is the number of coefficients in s, N_s is the number of code-cubes in s (the code-cubes are of equal size), $\mathrm{M}_{Cc_i}^{L}$ is as defined in (3.7), and g_s is a factor used to compensate for the non-energy preserving characteristics of the bi-orthogonal Le Gall 5/3 wavelet filter. Factor g_s is a function of the specific wavelet filters used for reconstruction and is calculated from the filter coefficients [26].

In order to attain a gradual increase in peripheral quality around the VOI, we define a weighted MSE for code-cube Cc_i over the entire reconstructed 3D image as follows:

$$\hat{\overline{\mathrm{M}}}_{Cc_i}^{L} = \frac{1}{(1+w_{Cc_i})}\overline{\mathrm{M}}_{Cc_i}^{L} \tag{3.9}$$

where w_{Cc_i} is the weight of Cc_i as defined in (3.4) and $\overline{\mathrm{M}}_{Cc_i}^{L}$ is as defined in (3.8). Note that for code-cubes

within the VOI, $w_{Cc_i} = 1$ and $\hat{\overline{\mathrm{M}}}^{L}_{\mathrm{Cc}_1} = \frac{1}{2}\overline{\mathrm{M}}^{L}_{\mathrm{Cc}_1}$. However, for code-cubes outside the VOI, $w_{Cc_i} < 1$ and $\hat{\overline{\mathrm{M}}}^{L}_{\mathrm{Cc}_1} > \frac{1}{2}\overline{\mathrm{M}}^{L}_{\mathrm{Cc}_1}$. The latter translates into a greater distortion, at quality layer *L*, for those code-cubes with a low mean energy content and located peripherally far from the VOI.

Thus, for a 3D image coded using a total of *I* code-cubes, the overall distortion at quality layer *L* is:

$$D^{L} = \sum_{i=1}^{I} \hat{\overline{\mathrm{M}}}^{L}_{Cc_i} \tag{3.10}$$

The key to attaining VOI decoding capabilities at quality layer *L*, is to include only the truncated bit-streams of those code-cubes within the VOI. Under this condition, the output bit-stream at quality layer *L* is the summation of the truncated bit-streams of the code-cubes within the VOI:

$$Y^{L} = \sum_{Cc_i \in \mathrm{VOI}} y^{L}_{Cc_i} \tag{3.11}$$

where Y^L is the output bit-stream at quality layer *L* and $y^{L}_{Cc_i}$ is the truncated bit-stream of Cc_i at quality layer *L*. The bit-rate of Y^L is then the summation of the bit-rates of each $y^{L}_{Cc_i}$:

$$R_{Y^L} = \sum_{Cc_i \in \mathrm{VOI}} R_{y^{L}_{Cc_i}} \tag{3.12}$$

where R_{Y^L} denotes the overall bit-rate of Y^L and $R_{y^{L}_{Cc_i}}$ denotes the bit-rate of $y^{L}_{Cc_i}$.

The overall distortion of the reconstructed 3D image at quality layer *L*, assuming the output bit-stream in (3.11), is thus:

$$D^{L} = \sum_{Cc_i \in \mathrm{VOI}} \hat{\overline{\mathrm{M}}}^{L}_{Cc_i} + \sum_{Cc_i \notin \mathrm{VOI}} \hat{\overline{\mathrm{m}}}^{L}_{Cc_i} \tag{3.13}$$

where $\hat{\overline{\mathrm{m}}}^{L}_{Cc_i}$ denotes the weighted MSE added to the overall distortion D^L if $y^{L}_{Cc_i}$ is not included in layer *L*, and

$\hat{\overline{\mathrm{M}}}^{L}_{\mathrm{Cc_i}}$ is as defined in (3.9). Using equations (3.7)-(3.9), $\hat{\overline{\mathrm{m}}}^{L}_{Cc_i}$ is calculated by equating $\hat{c}$, the quantized representation of the *k*th sample of code-cube Cc_i, to zero.

In order to increase the overall quality of the reconstructed 3D image at quality layer *L*, while retaining the VOI decoding capabilities and allowing for the decoding of contextual background information, we encode some bit-streams $y^{L}_{Cc_i} \notin \mathrm{VOI}$ along with bit-streams $y^{L}_{Cc_i} \in \mathrm{VOI}$. For a maximum bit-rate at quality layer *L*, some bit-streams $y^{L}_{Cc_i} \in \mathrm{VOI}$ in (3.11) may have to be discarded in order to accommodate bit-streams $y^{L}_{Cc_i} \notin \mathrm{VOI}$. Due to the resolution scalability features of the output bit-stream, bit-streams $y^{L}_{Cc_i} \in \mathrm{VOI}$ should be discarded in a sequential order starting with those comprising the first decomposition level (i.e., the highest-frequency sub-bands) and ending with those comprising the last decomposition level (i.e., the lowest-frequency sub-bands). Hence, the distortion in (3.13) can be expressed as follows:

$$\begin{aligned} D^{L} &= \sum_{i=i}^{I} \hat{\overline{\mathrm{M}}}^{L}_{Cc_i} \delta(y^{L}_{Cc_i}) + \sum_{i=i}^{I} \hat{\overline{\mathrm{m}}}^{L}_{Cc_i} \left[1 - \delta(y^{L}_{Cc_i})\right] \\ &= \sum_{i=i}^{I} \delta(y^{L}_{Cc_i}) \left[\hat{\overline{\mathrm{M}}}^{L}_{Cc_i} - \hat{\overline{\mathrm{m}}}^{L}_{Cc_i}\right] + \sum_{i=i}^{I} \hat{\overline{\mathrm{m}}}^{L}_{Cc_i} \end{aligned} \tag{3.14}$$

where $\delta(y^{L}_{Cc_i})$ is 1 if $y^{L}_{Cc_i}$ is included in layer *L* (otherwise it is zero).

In order to attain the optimal overall reconstruction quality of the 3D image at quality layer *L,* we minimize D^{L} in (3.14) under two bit-rate constraints:

$$\begin{aligned} &\sum_{i=1}^{I} R_{y^{L}_{Cc_i}} \delta(y^{L}_{Cc_i}) \le R_{Y^{L}} \\ &\sum_{Cci \notin \mathrm{VOI}} R_{y^{L}_{Cc_i}} \delta(y^{L}_{Cc_i}) < \sum_{Cci \in VOI} R_{y^{L}_{Cc_i}} \delta(y^{L}_{Cc_i}) \end{aligned} \tag{3.15}$$

where $R_{y^{L}_{Cc_i}}$ is the bit-rate of $y^{L}_{Cc_i}$, $R_{Y^{L}}$ is the maximum available bit-rate at quality layer *L*, and $\delta(y^{L}_{Cc_i})$ is 1 if $y^{L}_{Cc_i}$ is included in layer *L* (otherwise it is zero). Note that the constraints in (3.15) force the bit-rate spent on bit-streams $y^{L}_{Cc_i} \notin \mathrm{VOI}$ to be less than the bit-rate spent on bit-streams $y^{L}_{Cc_i} \in \mathrm{VOI}$. This guarantees that the VOI is decoded at higher quality than the rest of the 3D image.

We solve the optimization problem defined in (3.14)-(3.15) by finding the points that lie on the lower convex hull of the rate-distortion plane corresponding to the possible sets of bit-stream assignments.

3.3 Experimental Results and Discussion

We obtained two sets of experimental results. The first set evaluated the performance of the proposed method for VOI decoding at various bit-rates, including lossless reconstruction. The second set evaluated the effect of code-cube sizes on coding performance and size of the decoded VOI. We conclude this section with a discussion on the complexity of the proposed method.

3.3.1 Evaluation of VOI decoding at various bit-rates

Our test data set consisted of three 8-bit MRI and three 12-bit CT sequences of various resolutions. We defined a single VOI comprising clinically relevant information in each of the test sequences. The characteristics of the 3D test sequences, the corresponding VOI coordinates and code-cube sizes used for entropy coding are summarized in Table 3.4. Sequence 1 comprises MRI slices (sagittal view) of a human spinal cord; Sequence 2 comprises MRI slices (axial view) of a human head; and Sequence 3 comprises MRI slices (sagittal view) of a human knee. The test CT sequences comprise consecutive slices (axial view) of the 'Visible Male' (Sequences 4 and 5) and 'Visible Woman' (Sequence 6) data sets maintained by the National Library of Medicine (NLM) [27]. In this work, the VOI is defined in the spatial domain by two sets of values, $p(x,y,z)$ and $m[X,Y,Z]$; where $p(x,y,z)$ denotes the lower-left corner coordinates closest to the coordinate origin and $m[X,Y,Z]$ denotes the dimensions of the VOI.

Table 3-4: 3D test medical images and corresponding VOI coordinates and code-cube sizes

Modality	**Dimensions** Size {x,y,z}(mm) (slices:pixels per slice:bpv)	**VOI coordinates**		**Code-cube size*** $a\times a\times a$ (samples)	**Scaling value**†
		$p(x,y,z)$ (mm)	$m[X,Y,Z]$ (mm)		
1. MRI	{240,240,33} (11:512×512:8)	p(117,24,9)	m[85,140,12]	*32×32×32*	3
2. MRI	{256,256,100} (100:256×256:8)	p(128,0,0)	m[128,256,100]	*16×16×16*	4
3. MRI	{272,272,100} (50:512×512:8)	p(44,31,20)	m[170,159,54]	*32×32×32*	3
4. CT	{270,270,120} (120:512×512:12)	p(122,21,50)	m[111,212,50]	*32×32×32*	6
5. CT	{270,270,100} (100:512×512:12)	p(48,48,0)	m[185,137,30]	*32×32×32*	5
6. CT	{480,480,200} (200:512×512:12)	p(47,83,30)	m[353,286,40]	*32×32×32*	6

* Defined for the first level of decomposition.
† Used in the GSB method to scale up coefficients associated with the VOI above background coefficients.
MRI: magnetic resonance imaging. CT: computed tomography. VOI: volume of interest.

We compared the performance of the proposed compression method to that of 3D-JPEG2000 with VOI coding, using the MAXSHIFT and GSB methods. 3D-JPEG2000 is the extension of JPEG2000 for

compression of 3D images [28, 29]. 3D-JPEG2000 employs a 3D discrete wavelet transform across the slices with the resulting 3D sub-bands being entropy coded by first grouping coefficients into smaller 3D sections called 3D code-blocks. As mentioned earlier, MAXSHIFT scales up the coefficients associated with a VOI well above the background coefficients. At the decoder side, the non-zero VOI and background coefficients are identified by their magnitude, and thus, VOI coefficients are completely decoded before any background coefficients. The GSB method, on the other hand, scales up the coefficients associated with a VOI by a certain scaling value. Depending on the scaling value, some of the bits of the VOI coefficients may be encoded in conjunction with the bits of the background coefficients. The GSB method requires, however, the generation of a VOI mask at the encoder and decoder sides, as well as the coding and transmission of the VOI shape information, which may increase the computational complexity and overall bit-rate of the compressed bit-stream [30].

It is important to note that due to the scaling-up process performed by MAXSHIFT, the entropy decoder in 3D-JPEG2000 must be capable of decoding a large number of bit-planes. Current decoder implementations conforming to the JPEG2000 standard may not be capable to decode such large number of bit-planes, which renders the MAXSHIFT method not suitable for lossless reconstruction of 12-bit medical imaging data with VOI decoding capabilities. In this work, we used the *OpenJPEG* implementation of 3D-JPEG2000 [31].

In our proposed compression method, we employed the Le Gall 5/3 wavelet filter implemented using the lifting step scheme to decompose the test images with four levels of decomposition in all three dimensions. We created a layered output bit-stream whose reconstruction quality progressively improves up to lossless reconstruction.

For the case of 3D-JPEG2000, we employed four levels of decomposition in all three dimensions. We losslessly coded the resulting 3D sub-bands using 3D code-blocks and precincts to group coefficients describing the same 3D spatial region at the same decomposition level. The dimensions of the 3D code-blocks and precincts were selected to match the dimensions of the code-cubes employed in our proposed method, as tabulated in column 4, Table 3.4. For each test sequence, we created a layered output bit-stream whose reconstruction quality progressively improves up to lossless reconstruction. We employed the MAXSHIFT method (8-bit sequences) and the GSB method (8- and 12-bit sequences) to define a VOI according to the VOI coordinates tabulated in column 3, Table 3.4.

In order to measure the reconstruction quality of the VOI and background at different bit-rates, we employed the peak signal-to-noise ratio (PSNR), which for a 3D image of bit-depth m is defined by:

$$\mathrm{PSNR} = 20\log_{10}\frac{(2^m - 1)}{\sqrt{\mathrm{MSE}}}$$

$$\text{MSE} = \frac{1}{K}\sum_{k=1}^{K}(c_k - \hat{c}_k)^2 \tag{3.16}$$

where MSE denotes the mean square error, $(2^m - 1)$ is the maximum voxel value in the 3D image, K is the total number of voxels in the area to be evaluated (e.g., the VOI), c_k and $\hat{c}_k$ are the original and reconstructed values of the kth voxel, respectively.

In the case of the GSB method, we empirically selected the scaling value that produces the VOI quality (in terms of the PSNR) most similar to the VOI quality attained by our proposed method, while still allowing for decoding of partial background information at different bit-rates. The selected scaling values for the test sequences are tabulated in column 5, Table 3.4.

Figure 3.5 plots the PSNR values of the VOI and background of the 3D test sequences after decoding at a variety of bit-rates (in bits per voxel, bpv). It can be seen that, for all test sequences, the proposed method achieves higher PSNR values for the VOI and background than those achieved by the GSB method. Even though the GSB method allows for decoding of partial background information in conjunction with the VOI, this partial information is determined by manually selecting a scaling value for the background coefficients, which may affect the coding performance. The proposed method requires no manual selection of a scaling value, as it employs a weighting model and an optimization technique to determine the optimal amount of background information that minimizes the overall distortion of the image. This is done by reordering the output bit-stream after compression. As a result, in the peripheral regions of the VOI, low-frequency code-cubes with large weights have a greater opportunity to be included earlier in the output bit-stream. In the VOI, both low- and high-frequency code-cubes are included earlier in the output bit-stream because of their large weights.

It can also be seen in Fig. 3.5 that, for 8-bit sequences, MAXSHIFT achieves higher PSNR values for the VOI than those achieved by the proposed method, especially at low bit-rates (e.g., bit-rates lower than 0.40 bpv, Sequences 1 and 3). This is expected, since MAXSHIFT first decodes the VOI coefficients before decoding any background coefficients. Note that the apparent high PSNR values achieved by MAXSHIFT for the background at low bit-rates are due to the smearing effects of the wavelet filter, which may result in some background areas surrounding the VOI being decoded in conjunction with the VOI. Also note that as the bit-rate increases, the quality of the VOI decoded by the three evaluated methods tends to be very similar, since more bits are decoded and the reconstruction quality approaches the lossless case.

It is important to mention that the plots in Fig. 3.5(a)-(c) present different behaviors for the case of the MAXSHIFT method. This is mainly due to the size of the VOI, which may affect the number of bits in each bit-plane needed to fully reconstruct the VOI before the background. For small VOIs, e.g., those decoded in Sequences 1 and 3, MAXSHIFT achieves higher PSNR values for the VOI than the proposed method at bit-rates lower than 0.4 bpv (see Fig. 3.5(a) and 5(c)). In this case, the VOI does not include a large proportion of

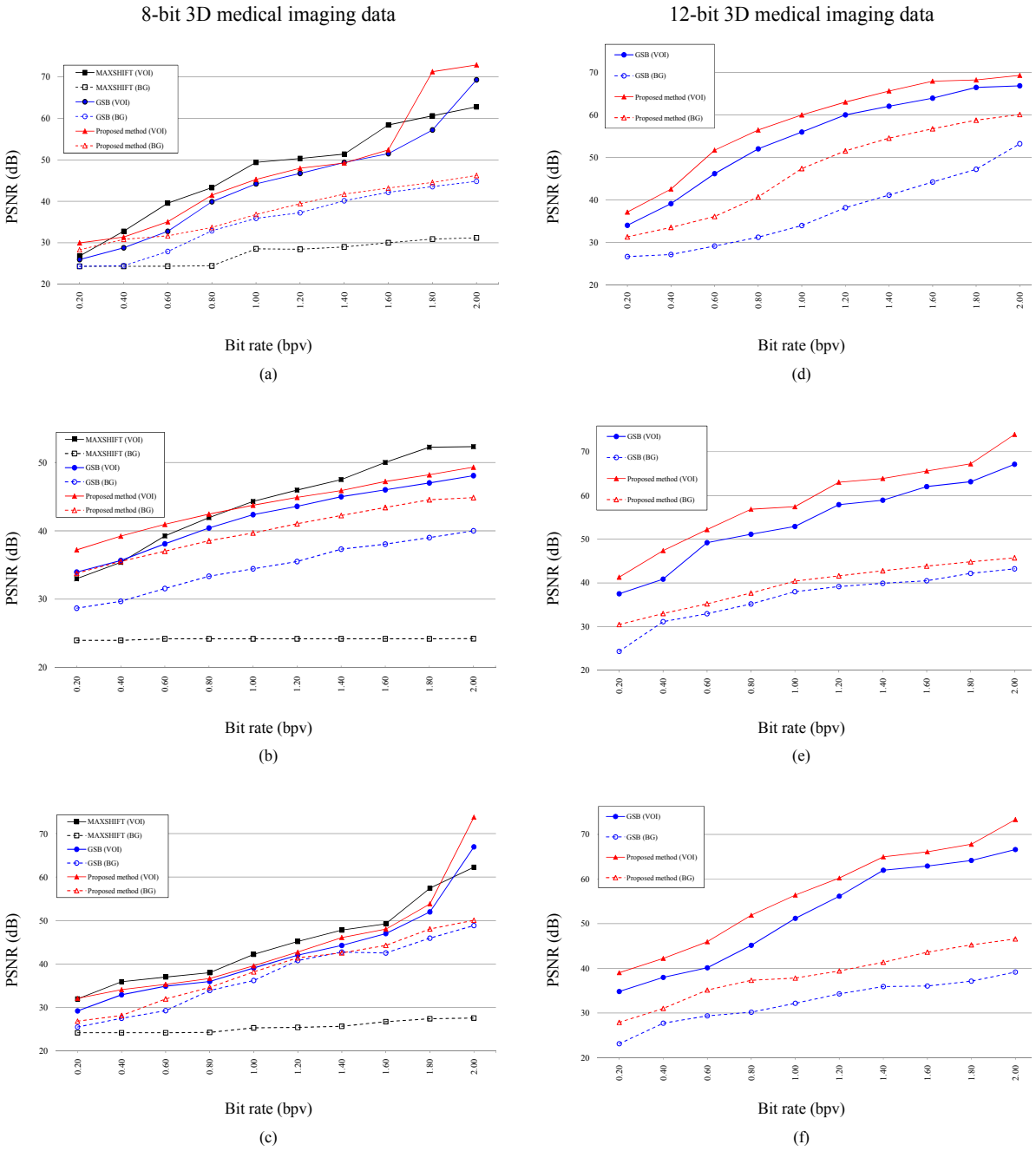


Fig. 3-5: PSNR values (in dB) for the VOI and background of 8-bit and 12-bit 3D medical imaging data decoded at various bit-rates after compression using different methods (see Table IV). (a) Sequence 1, MRI slices (sagittal view) of a human spinal cord. (b) Sequence 2, MRI slices (axial view) of a human head. (c) Sequence 3, MRI slices (sagittal view) of a human knee. (d) Sequence 4 and (e) Sequence 5, consecutive CT slices (axial view) of the 'Visible Male' data set maintained by the National Library of Medicine (NLM) [27]. (f) Sequence 6, consecutive CT slices (axial view) of the 'Visible Woman' data set maintained by the NLM.

significant coefficients and the number of most significant bit-planes in the VOI and background are similar. MAXSHIFT is therefore capable of decoding the VOI at high qualities at low bit-rates. For larger VOIs, e.g., that decoded in Sequence 2 (where the VOI comprises half of the entire volume), the proportion of significant coefficients in the VOI is larger and, therefore, a larger number of bits is needed to fully recover the VOI before the background. This explains the lower PSNR values achieved by MAXSHIFT for the VOI at bit-rates lower than 0.80 bpv when compared to the proposed method (see Fig 3.5(b)).

Lossless compression ratios and bit-rates for the three evaluated methods are tabulated in Table 3.5. The proposed method achieves compression ratios comparable to those achieved by MAXSHIFT and the GSB method, with the additional advantage of allowing for decoding any VOI from the same compressed bit-stream.

Table 3-5: Lossless compression ratios and bit-rates of 3D medical images using various compression methods

Modality Size {x,y,z}(mm) (slices:pixels per slice:bpv)	**Compression method ***		
	MAXSHIFT method	GSB method	**Proposed method**
	compression ratio (bit-rate, bits per voxel)		
1. MRI {240,240,33} (11:512×512:8)	2.39:1 (3.34 bpv)	2.46:1 (3.25 bpv)	**2.44:1** **(3.27 bpv)**
2. MRI {256,256,100} (100:256×256:8)	1.98:1 (4.04 bpv)	2.02:1 (3.96 bpv)	**2.00:1** **(4.00 bpv)**
3. MRI {272,272,100} (50:512×512:8)	4.25:1 (1.88 bpv)	4.31:1 (1.85 bpv)	**4.34:1** **(1.83 bpv)**
4. CT {270,270,120} (120:512×512:12)	N/A	3.25:1 (4.92 bpv)	**3.22:1** **(4.96 bpv)**
5. CT {270,270,100} (100:512×512:12)	N/A	2.31:1 (6.92.bpv)	**2.36:1** **(6.77 bpv)**
6. CT {480,480,200} (200:512×512:12)	N/A	2.47:1 (6.47 bpv)	**2.49:1** **(6.41 bpv)**

***** A single VOI was defined in each sequence as specified in Table IV.
MRI: magnetic resonance imaging. CT: computed tomography. VOI: volume of interest.
N/A: not applicable.

Figure 3.6 illustrates sample reconstructed slices at 0.6 bpv belonging to the VOI of Sequences 1 and 3 (see Table IV). It can be seen that the GSB and the proposed methods are capable to decode the VOI while still including partial background information, which allows placing the VOI into the context of the 3D image, in this case the sagittal view of a human spinal cord (Sequence 1), and a human knee (Sequence 3). The proposed method, however, decodes the background information peripherally around the VOI according to the mean energy of the code-cubes, which results in a higher reconstruction quality than that attained by the GSB method. Note that the VOIs decoded by the MAXSHIFT and the GSB methods appear to be of different

size when compared to the VOI decoded by the proposed method. Let us remember that the MAXSHIFT and the GSB methods work on a coefficient-basis and thus, both methods are able to decode more precisely only those coefficients within the VOI at higher quality than the background coefficients. In other words, the granularity of the MAXSHIFT and GSB methods in representing a VOI in the sub-band domain is one wavelet coefficient, as opposed to the proposed method, where the granularity is one code-cube. Therefore, in the proposed method, the code-cube sizes have a direct impact on the size of the decoded VOI. This will be further discussed in the next subsection.

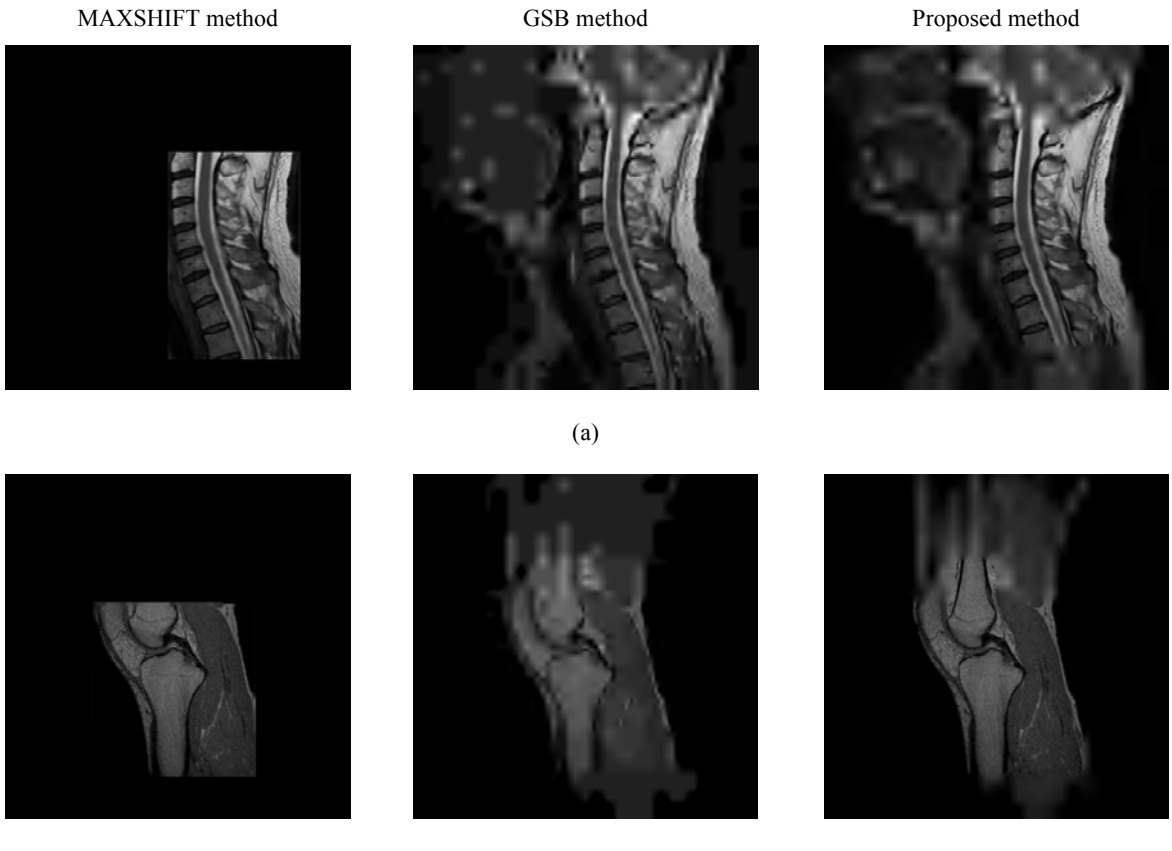


Fig. 3-6: (a) Slice no. 5 belonging to the VOI of Sequence 1 and (b) slice no. 22 belonging to the VOI of Sequence 3 (see Table IV) reconstructed at 0.6 bpv after compression using the MAXSHIFT method, the GSB method, and the proposed method. Observed PSNR values were (a) 39.56 dB (VOI) and 24.33 dB (background) for MAXSHIFT; 32.97 dB (VOI) and 27.90 dB (background) for the GSB method; and 35.07 dB (VOI) and 31.60 dB (background) for the proposed method; and (b) 39.99 dB (VOI) and 24.16 dB (background) for MAXSHIFT; 34.89 dB (VOI) and 29.25 dB (background) for the GSB method; and 35.31 dB (VOI) and 31.90 (background) for the proposed method.

3.3.2 Evaluation of the effect of code-cube sizes

In this section, we evaluated the trade-off between the size of the decoded VOI and coding performance, for various code-cube sizes. In order to measure how similar the size and location of the decoded VOI are to the size and location of the desired VOI, we employ a VOI shape decoding quality defined by a set of two values, $e_p(x,y,z)$ and v_r; where $e_p(x,y,z)$ denotes the absolute value of the difference between the lower left hand corner coordinates of the desired VOI and those of the decoded VOI, and v_r is a real number defined by:

$$v_r = \frac{\sum_{Cc_i \in \mathrm{VOI}} V_{Cc_i}}{V_{\mathrm{VOI}}} \tag{3.17}$$

where V_{Cci} is the volume size of the region represented by code-cube Cc_i in the spatial domain (the summation of all $Cc_i \in \mathrm{VOI}$ comprise the decoded VOI), and V_{VOI} is the volume size of the desired VOI in the spatial domain. A value $v_r = 1$ means that the decoded VOI is equal in volume size to the desired VOI, a value $v_r > 1$ means that the decoded VOI is larger in volume size than the desired VOI, whereas a value $v_r < 1$ means that the decoded VOI is smaller in volume size than the desired VOI.

Let us remember that each code-cube is associated with a limited spatial region due to the finite footprint of the wavelet kernel. It is thus expected that small code-cube sizes will result in higher VOI shape decoding qualities (i.e., $e_p(x,y,z)$ values close to (0,0,0) with v_r values close to 1). However, small code-cube sizes may also result in reduced coding performance due to the increased number of independent bit-streams needed to represent all the code-cubes at each quality layer L.

Figure 3.7 plots the VOI shape decoding quality and PSNR values for the VOI of Sequences 1 and 4 (see Table 3.4) after decoding at a variety of bit-rates using different code-cube sizes. Figure 3.8 shows sample reconstructed slices at 0.6 bpv of Sequence 1 after encoding using different code-cube sizes.

As expected, results in Fig. 3.7 show that as the code-cube size is reduced the coding performance decreases, but the VOI shape decoding quality increases. This can be seen in Fig. 8(b), where the VOI seems to be larger than in Fig. 3.8(d)-(f), because the code-cubes are not small enough for the VOI to be accurately decoded. Also note the blocky artifacts when employing code-cubes of 8×8×8 samples, which are the result of the low coding performance due to the increased number of independent bit-streams needed to represent all code-cubes. In this case, code-cubes of 32×32×32 and 64×64×64 samples (defined for the first decomposition level) present the best trade-off between VOI shape decoding quality and coding performance. If the VOI coordinates are known *a priori* the entropy coding process, large code-cubes may be employed for large VOIs or if the whole volume needs to be decoded; whereas small code-cubes may be employed for small VOIs or if a high VOI shape decoding quality is required.

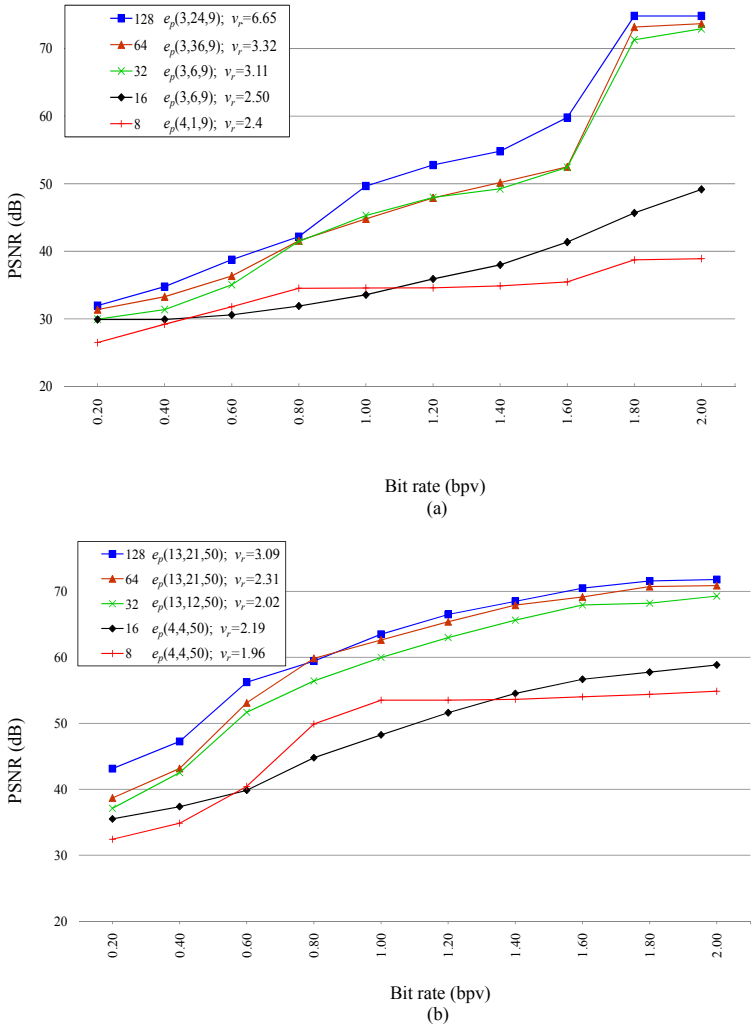
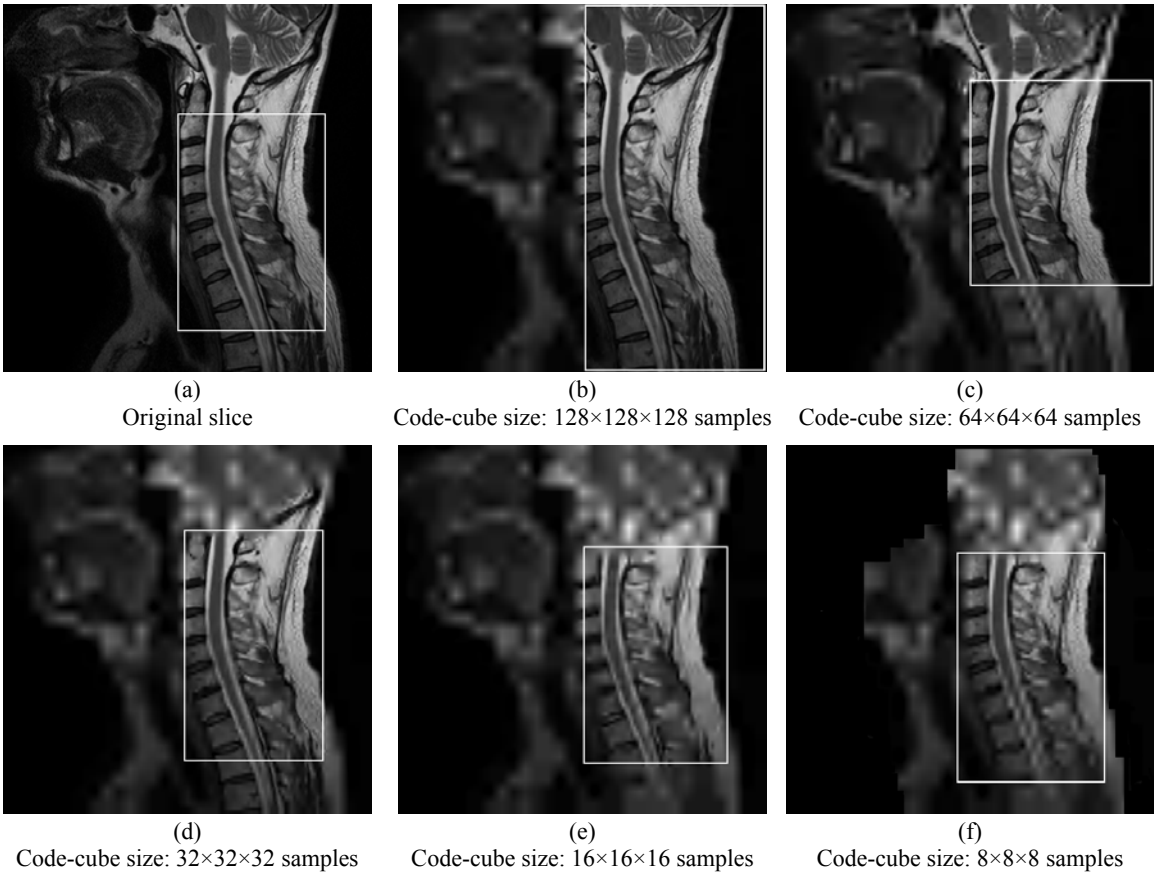


Fig. 3-7: PSNR (in dB) for the VOI and VOI shape decoding quality values of (a) Sequence 1 and (b) Sequence 4 after decoding at a variety of bit-rates using different code-cube sizes (see Table 3.4).

As expected, results in Fig. 3.7 show that as the code-cube size is reduced the coding performance decreases, but the VOI definition quality improves. This can be seen in Fig. 3.8(b), where the VOI seems to be larger than in Fig. 3.8(d)-(f) because the code-cubes are not small enough for the VOI to be accurately defined. Also note the blocky artifacts when employing code-cubes of 8×8×8 samples, which are the result of the low coding performance due to the increased number of independent bit-streams needed to represent all code-cubes. In this case, code-cubes of 32×32×32 and 64×64×64 samples (defined for the first decomposition level) present the best trade-off between coding performance and VOI definition quality. If the VOI coordinates are known *a priori* the coding process, large code-cubes may be employed for large VOIs or if the whole volume needs to be decoded; while small code-cubes may be employed for small VOIs or if a good VOI definition quality is required.

Finally, it is important to mention that the proposed compression method may be also employed to compress any 4D medical image with a 4D VOI (i.e., a VOI defined along a set of volumes of a 4D medical image). This may be done by compressing each volume of the 4D data set independently, while defining a VOI in each volume.



(a) Original slice
(b) Code-cube size: 128×128×128 samples
(c) Code-cube size: 64×64×64 samples
(d) Code-cube size: 32×32×32 samples
(e) Code-cube size: 16×16×16 samples
(f) Code-cube size: 8×8×8 samples

Fig. 3-8: (a) Original slice no. 5 of Sequence 1. The voxels belonging to the desired VOI are delimited by a square area (see Table 3.4). (b)-(f) Slice no. 5 of Sequence 1 reconstructed at. 0.6 bpv after coding using various code-cubes sizes with four level of decomposition (code-cube sizes are defined for the first level of decomposition). The voxels belonging to the decoded VOI are delimited by a square area.

3.3.3 Computational complexity considerations

We conclude our performance evaluation with a brief discussion regarding the complexity of the proposed compression method. Compared to 3D-JPEG2000 with VOI coding (MAXSHIFT and GSB methods), the proposed method presents a higher complexity at the encoder side due mainly to the bit-stream reordering procedure. This augmented complexity is a consequence of the calculation of the code-cube weights and the layer optimization technique, which needs to be performed each time a VOI is to be decoded.

It is important to remember that, in the proposed method, the entropy coding process needs to be performed only once for a 3D medical image, since the decoding of a VOI simply requires the reordering of

the compressed bit-stream. As mentioned earlier, the calculation of the code-cube weights for a requested VOI simply requires the re-computation of two values for each code-cube. Moreover, the MSE required during the layer optimization technique is easily calculated from the information about the code-cube contributions into each quality layer, which is stored as header information during the coding process.

At the decoder side, the complexity of the proposed method is very similar of that of the MAXSHIFT method, since there is no need for the decoder to reorder the bit-stream prior to decoding. Compared to the GSB method, the decoding complexity of the proposed method is lower, since the GSB method requires the generation of a VOI mask prior to decoding.

Finally, it is important to remark that in the proposed method, all information needed to perform the bit-stream reordering procedure and layer optimization technique is stored and transmitted as header information. In the case of the test sequences evaluated in this work, this additional information represents a mere 0.04% - 0.5% of the compressed bit-rate.

3.4 Conclusions

We presented a novel scalable 3D medical image compression method with optimized VOI coding within the framework of interactive telemedicine applications. The method is based on a 3D integer wavelet transform and a modified version of EBCOT that exploits correlations between wavelet coefficients in three dimensions and generates a scalable layered bit-stream. The method employs a bit-stream reordering procedure and an optimization technique to optimally encode any VOI at the highest quality possible in conjunction with contextual background information from a lossy to a lossless representation. We demonstrated the two main novelties of the method; namely, the ability to decode any VOI from the compressed bit-stream without the need to recode the entire 3D image; and the ability to enhance the visualization of the data at any bit-rate by including contextual background information with peripherally increasing quality around the VOI. We evaluated the performance of the proposed method on real 8-bit and 12-bit 3D medical images of various resolutions. We demonstrated that the proposed method achieves higher reconstruction qualities than those achieved by 3D-JPEG2000 with VOI coding at a variety of bit-rates. We also demonstrated that the proposed method attains lossless compression ratios comparable to those attained by 3D-JPEG2000 with VOI coding. Finally, we studied the effect on coding performance and VOI decoding capabilities of the proposed method with different coding parameters.

3.5 References

[1] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto and J. Cornelis, "Wavelet coding of volumetric medical datasets," IEEE Trans. Medical Imaging, vol. 22, no. 3, pp. 441-458, March 2003

[2] Z. Xiong, X. Wu, S. Cheng and J. Hua, "Lossy-to-lossless compression of medical volumetric images using three-dimensional integer wavelet transforms," IEEE Trans. Medical Imaging, vol. 22, no. 3, pp. 459-470, March 2003

[3] X. Wu and T Qiu, "Wavelet coding of volumetric medical images for high throughput and operability," IEEE Trans. Medical Imaging, vol. 24, no. 6, pp. 719-727, June 2005.

[4] G. Menegaz and J.P. Thirian, "Three-dimensional encoding/two-dimensional decoding of medical data," IEEE Trans. Medical Imaging, vol. 22, no. 3, pp. 424-440, March 2003

[5] R. Srikanth and A.G. Ramakrishnan, "Contextual encoding in uniform and adaptive mesh-based lossless compression of MR images," IEEE Trans. Medical Imaging, vol. 24, no. 9, pp. 1199-1206, September 2005

[6] V. Sanchez, R. Abugharbieh and P. Nasiopoulos, "Symmetry-based scalable lossless compression of 3D medical image data," IEEE Trans. Medical Imaging, vol. 28, no. 7, pp. 1062-1072, July 2009

[7] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Trans. Signal Processing, vol. 41, pp. 3445-3462, Dec. 1993

[8] A. Said and W. Pearlman, "A new fast and efficient image coded based on set partitioning in hierarchical trees," IEEE Trans. Circuits and Systems for Video Technology, vol. 6, pp. 243-250, June 1996

[9] D. Taubman, "High performance scalable image compression with EBCOT," IEEE Trans. Image Processing, vol. 9, no. 7, pp. 1158-1170, July 2000

[10] K. Krishnan. M. Marcellin, A. Bilgin and M. Nadar, "Efficient transmission of compressed data for remote volume visualization," IEEE Trans. Medical Imaging, vol. 25, no. 9, pp. 1189-1199, September 2006

[11] Y. Liu and W. A. Pearlman, "Region of interest access with three-dimensional SBHP algorithm," Proc. SPIE 6077, pp. 17-19, 2006

[12] C. Doukas and I. Maglogiannis, "Region of interest coding techniques for medical image compression," IEEE Engineering in Med. And Biol. Magazine, vol. 25, no. 5, pp. 29-35, September-October 2007

[13] I. Ueno and W. Pearlman, "Region of interest coding in volumetric images with shape-adaptive wavelet transform," Proceedings of the SPIE, vol. 5022, pp. 1048-1055, 2003

[14] JPEG 2000 Part I: Final Draft International Standard (ISO/IEC FDIS15444-1), ISO/IECJTC1/SC29/WG1 N1855, August 2000

[15] J. Strom and P. C. Cosman, "Medical image compression with lossless regions of interest," Signal Proc., vol. 59, no. 3, pp. 155-171, June 1997

[16] A. Signoroni and R. Leonardi, "Progressive ROI coding and diagnostic quality for medical image compression," SPIE Proceedings Series, vol. 3309, no. 2, pp. 674-685, 1997

[17] X. Bai, J. S. Jin, and D. Feng, "Segmentation-based multilayer diagnosis lossless medical image compression," ACM International Conference Proceeding Series, vol. 100, pp. 9-14, June 2004

[18] A. R. Calderbank, I. Daubechies, W. Sweldens and B.L, Yeo, "Wavelet transforms that map integers to integers," Appl. Comput. Harmon. Anal., vol. 5, no. 3, pp. 332-369, 1998

[19] I. Daubechies and W. Sweldens, "Factoring wavelet transform into lifting steps," J. Fourier Anal. Appl., vol. 41, no. 3, pp. 247-269, 1998

[20] J. Xu, "Three-dimensional embedded subband coding with optimized truncation (3-D ESCOT)," Applied and Computational Harmonic Analysis, vol. 10, pp. 290-315, 2001

[21] N. Zhang, M. Wu, S. Forchhammer, and X. Wu, "Joint compression-segmentation of functional MRI data sets," Proceedings of the SPIE, vol. 5748, pp. 190-201, 2003

[22] Z. Wang and A. C. Bovik, "Embedded foveation image coding," IEEE Trans. Image Processing, vol. 10, no. 10, pp. 1397–1410, October 2001

[23] V. Sanchez, A. Basu and M.K. Mandal, "Prioritized region of interest coding in JPEG2000," IEEE Trans. on Circuits and system for Video Technology, vol. 14, no. 9, September 2004

[24] K. J. Wiebe and A. Basu, "Improving image and video transmission quality over ATM with fovea prioritization and priority dithering," Pattern Recognition Leters, vol. 22, pp. 905–915, 2001

[25] Z. Wang and A. C. Bovik, "Foveation scalable video coding with automatic fixation selection," IEEE Trans. Image Processing, vol. 12, pp. 243–254, Feb. 2003

[26] B. Usevitch, "Optimal bit allocation for biorthogonal wavelet coding," in Proc. Data Compression Conf., Snowbird, UT, Mar. 1996, pp.387–395. 1996

[27] The National Library of Medicine (NLM): http://www.nlm.nih.gov

[28] Information Technology—JPEG 2000 Image Coding System—Part 2: Extensions, ISO/IEC 15 444-2, 2002

[29] Information Technology—JPEG 2000 Image Coding System—Part 10: Extensions for three dimensional data, ISO/IEC 15 444-10, 2007

[30] T. Bruylants, P. Schelkens, and A. Tzannes "JP3D - Extensions for Three Dimensional Data (Part 10)," in The JPEG2000 Suite, P. Schelkens, A. Skodras, and T. Ebrahimi, Eds. UK: Wiley, 2009, pp. 218-219.

[31] Available on-line: http://www.openjpeg.org.

Chapter 4

4. Efficient Lossless Compression of 4D Medical Images Based on the Advanced Video Coding Scheme[1]

4.1 Introduction

Most current lossless medical image compression methods are designed to handle two dimensional (2D) and three dimensional (3D) data [1-4]. Compression of four dimensional (4D) medical images is still a relatively new area of research. Since 4D images are typically sequences of 3D images (volumes) and 3D images are comprised of 2D images (slices), 2D and 3D compression algorithms may be used to code slices or volumes independently. However, such algorithms ultimately fail to exploit redundancies in all four dimensions.

Few methods that exploit redundancies in all four dimensions have been proposed recently. These methods either use 4D wavelet transforms [5-7] or 3D motion compensation algorithms [8,9] to decorrelate the data. However, these methods still achieve compression ratios comparable to those of the 3D compression techniques such as 3D-JPEG2000.

In this chapter, we propose a novel lossless compression method designed for 4D medical images that efficiently exploits redundancies in all four dimensions. Our method is based on the most advanced features of the H.264/AVC (Advanced Video Coding) standard, namely multi-frame motion compensation, variable block-sizes for motion estimation and sub-pixel motion vector accuracy; as well as a novel differential coding algorithm for motion vectors. We compare our lossless compression method to JPEG2000, 3D-JPEG2000 and standard H.264/AVC [10]. Performance evaluations show that our proposed method provides a significant improvement on compression ratio of up to three times that of 3D compression techniques such as 3D-JPEG2000.

4.2 Proposed Compression Method

A 4D medical image can be denoted as $I(x,y,z,t)$, where the variables x and y denote the dimensions of the slices, the variable z denotes the dimension of the volumes and t denotes time. Our proposed compression method (Fig. 4.1) encodes a 4D medical image $I(x,y,z,t)$ of n volumes of s slices each as detailed below.

[1] A version of this chapter has been published. V. Sanchez, P. Nasiopoulos, and R. Abugharbieh, "Efficient Lossless Compression of 4D Medical Images Based on the Advanced Video Coding Scheme," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 4, pp. 442–446, 2008.

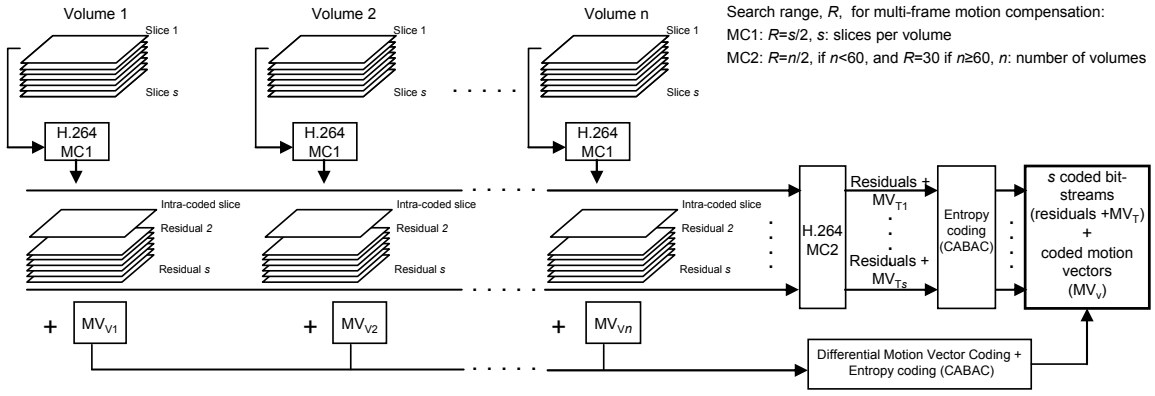


Fig. 4-1: Block diagram of the proposed lossless compression method for a 4D medical image of n volumes of s slices. MC1: first multi-frame motion compensation process. MC2: second multi-frame motion compensation process. MV_{Vi}: motion vector data produced after applying motion compensation on volume i. MV_{Tk}: motion vector data produced after applying motion compensation on set of residual slices k. CABAC: Context-Based Adaptive Binary Arithmetic Coding, the entropy coding method used to compress the data.

1. Spatial redundancies in the z dimension are reduced by processing each volume as a single video sequence using multi-frame motion compensation. The first slice of each volume is encoded as an I-frame using the intra-code mode of H.264/AVC while the remaining $s-1$ slices are encoded as P-frames using the inter-code mode [10]. Slices are encoded using non-overlapping macroblocks of 16×16 pixels, which may be further partitioned into smaller blocks of 16×8, 8×16 and 8×8 pixels. Our experiments on a large set of medical images have shown that block sizes smaller than of 8×8 pixels do not provide a significant coding improvement, but significantly increase the coding time. This first step results in s residual slices per volume and the corresponding motion vector data.
2. Temporal redundancies in the t dimension are then reduced by processing all the residual slices generated in *step 1* using multi-frame motion compensation with variable block-sizes. The residual slices are first arranged into s sets of n residual slices each and then each set is processed as an individual video sequence. The k^{th} set of residual slices to be processed in this step can be expressed as:

$$R_k = \left\{ r_k^1, r_k^2, r_k^3, ..., r_k^{n-1}, r_k^n \right\} \tag{4.1}$$

where r_k^i denotes the k^{th} residual slice of volume i. The first residual slice of each set is encoded as an I-frame while the remaining $n-1$ residual slices are encoded as P-frames. This second step results in the final residual slices and their corresponding motion vector

data, which both are compressed using the context-based adaptive binary arithmetic coder (CABAC) [10].

3. The motion vector data generated in *step 1* is then encoded using a novel differential motion vector coding algorithm described in section 4.2.1 and the resulting information is compressed using CABAC.
4. The final residual slices and the motion vector data compressed in *step 2*, and the motion vector data compressed in *step 3* comprise the final compressed bit-stream.

4.2.1 Proposed differential coding of motion vectors

The motion vector data of two consecutive volumes generated as described in *step 1* (see MV_Vs in Fig. 4.1) are often highly correlated since medical image volumes within dynamic data usually depict the same anatomical region undergoing certain changes (e.g., functional activation or motion) in time. We thus propose a differential coding algorithm that exploits such correlation. Algorithms that calculate the difference between motion vectors have been previously employed in the H.264/AVC standard to improve coding efficiency [10]. Here, we propose an algorithm that calculates the difference between two sets of motion vectors associated with the same spatial region in two consecutive volumes (e.g., MV_{Vi} and MV_{Vi-1}). The calculated difference is then entropy encoded using CABAC.

Let *C* be the current macroblock in volume *i* and slice *k* and let *P* be the previous macroblock in the same spatial position as C but in volume *i*−1 and slice *k*, as exemplified in Fig. 4.2.

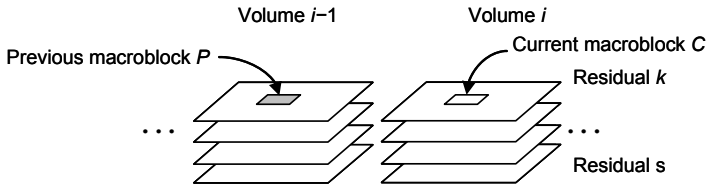


Fig. 4-2: Current macroblock *C* of slice *k* of volume *i*, and previous macroblock *P* in the same spatial position in slice *k* of volume *i*−1.

If *C* and *P* contain no partitions (Fig. 4.3(a)), the differential motion vector of *C* (dMV_C) is the difference between the motion vector of *C* (MV_C) and the motion vector of *P* (MV_P):

$$dMV_C = MV_C - MV_P \tag{4.2}$$

If, however, *C* contains partitions but *P* contains no partitions (Fig. 4.3(b)), then the differential motion vector of the j^{th} partition of *C* (dMV_{Cj}) is the difference between the motion vector of the j^{th} partition of *C* (MV_{Cj}) and MV_P:

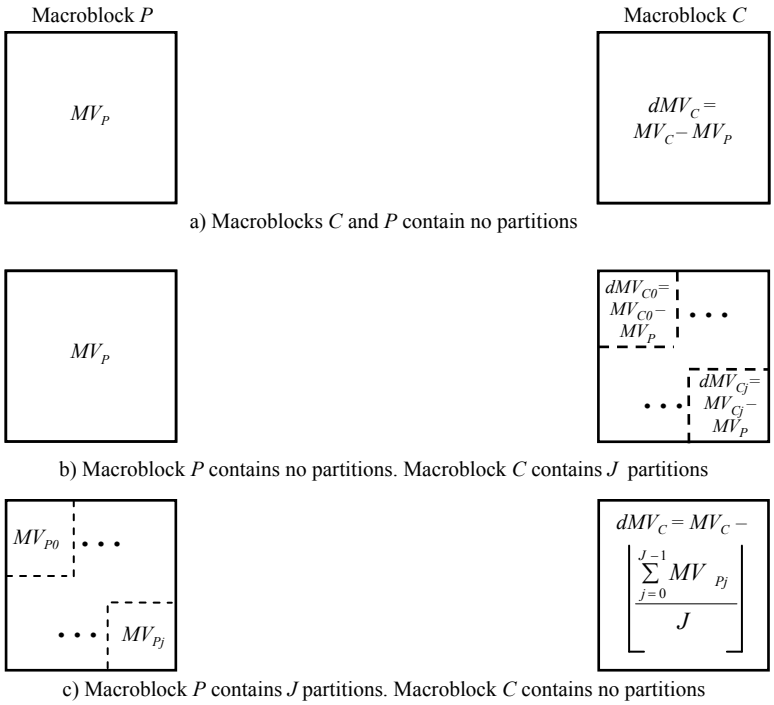


Fig. 4-3: Differential coding of MV_V motion vectors. *C*: current macroblock in volume *i* and slice *k*. *P*: previous macroblock in the same spatial position in volume $i-1$ and slice *k*. MV_C: motion vector of *C*. dMV_C: differential motion vector of *C*. MV_P: motion vector of *P*. $\lfloor\ \rfloor$: largest integer $\leq x$.

$$dMV_{Cj} = MV_{Cj} - MV_P \tag{4.3}$$

Finally, if *C* contains no partitions but *P* contains partitions (Fig. 4.3(c)), then dMV_C is the difference between MV_C and the average value of the motion vectors of all partitions of *P*:

$$dMV_C = MV_C - \left\lfloor \sum_{j=0}^{J-1} MV_{Pj} \Big/ J \right\rfloor \tag{4.4}$$

where MV_{Pj} is the motion vector of the j^{th} partition of *P*, *J* is the total number of partitions in *P* and $\lfloor\ \rfloor$ denotes the largest integer less than or equal to *x*.

Note that when both *C* and *P* contain partitions, each partition is treated as a single macroblock.

4.3 Performance Evaluation and Results

We tested the proposed compression method on twenty functional magnetic resonance imaging (fMRI) sequences, five dynamic magnetic resonance imaging (4D-MRI) sequences and five positron emission

tomography (PET) sequences. The fMRI sequences are comprised of volumes of axial and coronal slices of a human head and depict visual cortex activity. The 4D-MRI sequences describe the structure of a human hand undergoing movement at different time points. The PET sequences depict some brain activity in a rat. The first column of Table 4.1 summarizes the characteristics of the tested sequences, while samples of some of these sequences are shown in Fig. 4.4.

Table 4-1: Compression ratios of 4D medical images of varying modalities using different lossless compression methods

4D medical image	**Lossless compression method**				
Modality: ***volumes: slices per volume: pixels per slice: bits per pixel***	**JPEG2000**	**3D-JPEG2000**	**H.264/ AVC**	**Proposed method**	
				No DCMV	**DCMV**
1. fMRI:150:36:128×128:12	4.13:1	5.00:1	5.59:1	14.10:1	**15.06:1**
2. fMRI:163:45:128×128:12	3.98:1	5.13:1	5:47:1	13.57:1	**14.09:1**
3. fMRI:163:45:128×128:12	4.00:1	5.14:1	5.48:1	13.56:1	**14.11:1**
4. fMRI:163:45:128×128:12	3.95:1	5.08:1	5.37:1	13.37:1	**14.28:1**
5. fMRI:33:60:256×256:12	5.54:1	5.68:1	5.62:1	11.76:1	**11.87:1**
6. fMRI:33:60:256×256:12	5.60:1	5.74:1	5.68:1	11.93:1	**12.11:1**
7. fMRI:33:60:256×256:12	5.57:1	5.71:1	5.66:1	11.88:1	**12.02:1**
8. fMRI:126:36:128×128:12	4.81:1	6.36:1	4.85:1	10.33:1	**10.53:1**
9. fMRI:126:36:128×128:12	3.84:1	6.16:1	5.27:1	12.21:1	**12.34:1**
10. fMRI:126:36:128×128:12	3.71:1	6.03:1	4.86:1	11.21:1	**11.41:1**
11. fMRI:126:36:128×128:12	3.72:1	6.02:1	4.89:1	11.36:1	**11.55:1**
12. fMRI:126:36:128×128:12	3.75:1	6.17:1	4.97:1	11.54:1	**11.65:1**
13. fMRI:195:36:128×128:12	4.18:1	7.64:1	5.74:1	13.27:1	**13.35:1**
14. fMRI:195:36:128×128:12	4.24:1	8.05:1	6.02:1	13.71:1	**13.81:1**
15. fMRI:195:36:128×128:12	4.25:1	7.96:1	5.93:1	13.47:1	**13.62:1**
16. fMRI:195:36:128×128:12	4.09:1	7.25:1	5.33:1	12.40:1	**12.55:1**
17. fMRI:195:36:128×128:12	4.21:1	8.08:1	6.41:1	13.47:1	**13.59:1**
18. fMRI:195:36:128×128:16	4.16:1	12.15:1	8.03:1	14.02:1	**14.34:1**
19. fMRI:180:45:128×128:16	3.75:1	5.15:1	5.11:1	12.67:1	**12.98:1**
20. fMRI:180:45:128×128:16	4.01:1	5.85:1	5.69:1	11.96:1	**12.11:1**
21. 4D-MRI:6:87:512×352:8	2.09:1	2.12:1	2.41:1	3.64:1	**3.69:1**
22. 4D-MRI:6:87:512×352:16	2.36:1	2.86:1	2.98:1	3.75:1	**3.83:1**
23. 4D-MRI:5:87:512×352:8	2.89 :1	3.02 :1	2.95:1	3.47 :1	**3.51 :1**
24. 4D-MRI:5:87:512×352:16	2.73 :1	2.98 :1	3.11:1	3.72:1	**3.79 :1**
25. 4D-MRI:4:87:512×352:8	2.12 :1	2.56 :1	2.65:1	3.51:1	**3.54 :1**
26. PET:14:93:128×128:16	1.97:1	2.12:1	2.10 :1	2.56:1	**2.67 :1**
27. PET:6:93:128×128:8	1.32:1	1.65:1	1.76:1	1.85:1	**1.88:1**
28. PET:9:93:128×128:16	1.35:1	1.64:1	1.81:1	1.87:1	**1.90:1**
29. PET:16:93:128×128:16	1.85:1	1.94:1	2.15:1	2.26:1	**2.33:1**
30. PET:9:93:128×128:16	1.28:1	1.47:1	2.09:1	2.34:1	**2.39:1**

MRI: magnetic resonance imaging. fMRI: functional MRI. PET: positron emission tomography. DCMV: differential coding of MV_V motion vectors (see Fig. 4.1).

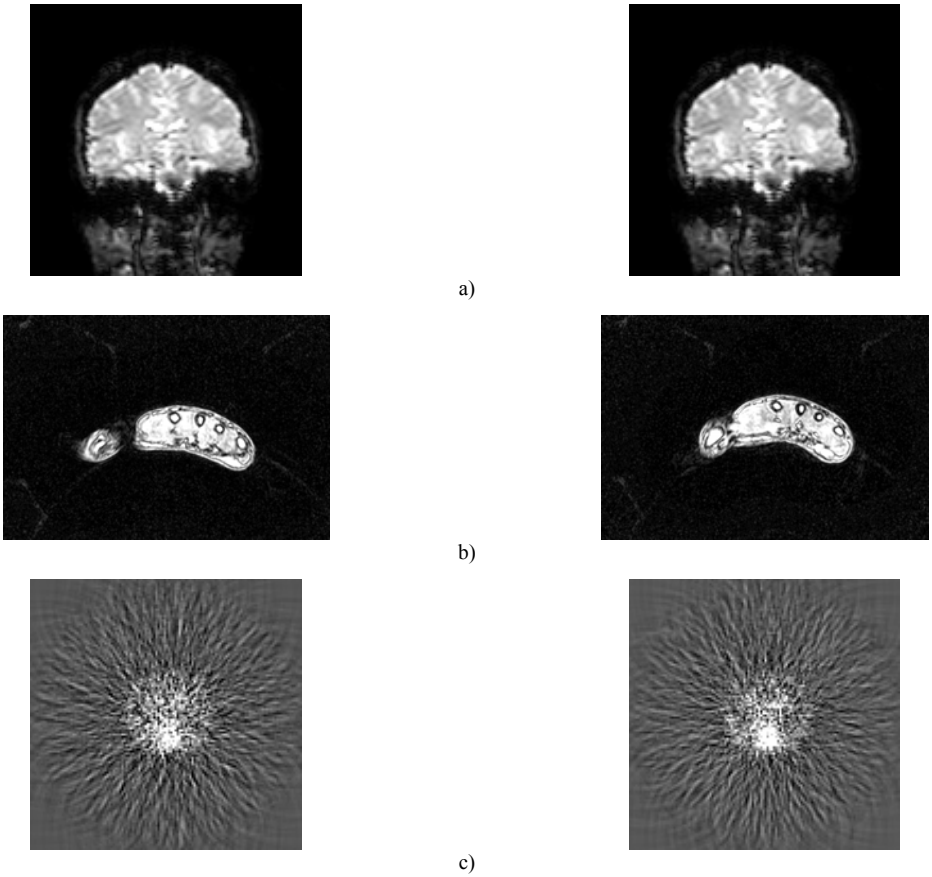


Fig. 4-4: Samples of tested 4D medical image sequences. Each row in the figure shows two slices of two consecutive volumes at the same spatial position of a) an fMRI sequence of the coronal view of a human head (128×128 pixels, 12 bits per pixel); b) a 4D-MRI (structural) sequence of the axial view of a human hand (512×352 pixels, 16 bits per pixel); and c) a PET sequence of the brain activity in a rat (128×128 pixels, 16 bits per pixel).

fMRI sequences typically feature high correlation among slices along the *t* dimension making them suitable for our proposed compression method. fMRI data thus represent the best case test set, for which the improvement on compression ratio over 3D and 2D systems is expected to be the highest. On the other hand, PET sequences can be considered as the worst case for the proposed compression method. PET data have little well-defined structures which makes it more difficult to estimate motion using block-based motion compensation as used in our approach.

For comparison purposes, we have also losslessly encoded the sequences using JPEG2000, 3D-JPEG2000 and standard H.264/AVC. We specifically compared our proposed method to these particular compression methods as their coding algorithms are very similar to those employed in the technical advances in lossless compression of 3D and 4D medical images reported in the literature [1-9]. We encoded the sequences using 3D-JPEG2000 and standard H.264/AVC by first grouping together slices across the *t* dimension into sets and then processing each set as an individual 3D image, in manner similar to our previous

work in [13]. In 3D-JPEG2000 we first applied a one dimensional discrete wavelet transform along the *t* dimension of each set with two levels of decomposition. Subsequently, we compressed all transformed slices using JPEG2000 with two levels of decomposition, which provides a good performance for lossless compression as suggested in [14]. In standard H.264/AVC, we losslessly encoded each set as a monochrome video sequence using an IPPP coding structure with no quantization for residuals to ensure lossless compression [15]. For the case of JPEG2000, we losslessly compressed each slice independently with two levels of decomposition. We used the Kakadu implementation of JPEG2000 and 3D-JPEG2000 [11]. We used version 10.1 of the H.264/AVC reference software [12].

In order to confirm lossless compression, we computed the mean square error (MSE) between the original sequences and the decoded sequences following compression using our proposed method. The MSE measures the variation between the two signals, which for two $u \times v$ monochrome images, O and R, is defined as:

$$MSE = \frac{1}{uv} \sum_{i=0}^{u-1} \sum_{j=0}^{v-1} \left[O(i,j) - R(i,j) \right]^2 \tag{4.5}$$

The MSE was computed for every slice of the test sequences and resulted in a MSE equal to exactly zero for all cases, which confirms that our proposed compression method is fully lossless.

Lossless compression ratios for the test sequences are summarized in Table 4.1. It can be observed that our proposed compression method significantly outperforms the other state-of-the-art methods achieving up to three times the compression ratio of 3D-JPEG2000 (see last column of Table 4.1). By first applying motion compensation in the *z* dimension, our proposed method finds the optimal residual slices and thus reduces the energy contained in each volume. This process results in fewer bits to be encoded after applying motion compensation in the *t* dimension.

The best improvement in compression ratio was achieved on the fMRI sequences. As described earlier, such images are quite suitable for our proposed method. The improvement in compression ratio for the 4D-MRI sequences is lower mainly due to the fact that these particular sequences feature a lower number of volumes (i.e., have lower temporal resolution), which results in a lower correlation between slices in the temporal dimension. However, the compression ratios in these sequences are still up to 50% better than those of 3D-JPEG2000. As expected, the lowest improvement in compression ratio was that achieved for the PET sequences due to their very low spatial resolution. Nevertheless, our method still achieves an improvement of about 7% over 3D-JPEG2000 on these sequences.

Column 5 of Table 4.1 corresponds to the compression ratios achieved without our proposed differential coding algorithm on the MV_V motion vectors. Note how an additional improvement of up to 5% can be observed when our differential coding algorithm is employed.

It is important to mention that the proposed compression method may be employed to compress any

4D medical image, independently of the imaging modality. However, as mentioned earlier, it provides the best coding performance with data that depicts well-defined structures, which makes it easier to estimate motion using block-based motion compensation.

4.4 Conclusions

In this chapter, we proposed a new efficient and fully lossless compression method designed for 4D medical image data. The method is based on the advanced features of the advanced video coding scheme (H.264/AVC) as well as a novel differential coding algorithm for motion vectors. The proposed compression method efficiently exploits data redundancy in all four dimensions thus providing a significantly superior compression performance compared to current state-of-the-art. Redundancies between slices within volumes are exploited by first calculating the motion-compensated residual slices, which, in turn, are encoded using motion compensation across the temporal dimension. Correlations between motion vectors are exploited next by employing a new differential coding algorithm that further improves compression performance. Our quantitative experimental results on a large number of medical image datasets of varying modalities show significant improvements in compression ratio of up to three times that of current 2D and 3D state-of-the-art compression techniques, such as JPEG2000 and 3D-JPEG2000.

4.5 References

[1] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto and J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Trans. on Medical Imag.,* vol. 22, no. 3, pp. 441-458, March 2003.

[2] G. Menegaz and J.P. Thirian, "Three-dimensional encoding/two-dimensional decoding of medical data," *IEEE Trans. on Medical Imag.,* vol. 22, no. 3 , pp. 424-440, March 2003.

[3] Z. Xiong, X. Wu, S. Cheng and J. Hua, "Lossy-to-lossless compression of medical volumetric images using three-dimensional integer wavelet transforms," *IEEE Trans. on Medical Imag.*, vol. 22, no. 3, pp. 459-470, March 2003.

[4] R. Srikanth and A.G. Ramakrishnan, "Contextual Encoding in Uniform and Adaptive Mesh-Based Lossless Compression of MR Images," *IEEE Trans. on Medical Imag.*, vol. 24, no. 9, pp. 1199-1206, Sept. 2005.

[5] L. Zeng, C.P. Jansen, S. Marsch, M. Unser, and P.R. Hunziker, "Four-dimensional wavelet compression of arbitrarily sized echocardiographic data," *IEEE Trans. on Medical Imag.*, vol. 21, no. 9, pp. 1179-1187, Sept. 2002.

[6] H. G. Lalgudi, A. Bilgin, M. W. Marcellin, and M. S. Nadar, "Compression of fMRI and ultrasound images using 4D-SPIHT," *Proc. of 2005 Int. Conf. on Image Processing*, vol. 2, pp. 11-14, Sept. 2005.

[7] L. Ying, and W.A. Pearlman, "Four-dimensional wavelet compression of 4-D medical images using scalable 4-D SBHP," *2007 Data Compression Conf.*, pp. 233-242, March 2007.

[8] P. Yan and A. Kassim, "Lossless and near-lossless motion compensated 4D medical image compression," *IEEE Int. Workshop on Biomedical Circuits and Systems*, pp. 13-16, Dec. 2004.

[9] A. Kassim, P. Yan, and W. S. Lee, "Motion compensated lossy-to-lossless compression of 4-D medical images using integer wavelet transforms," *IEEE Trans. on Information Technology in Biomedicine*, vol. 9, no. 1, pp. 132-138, March 2005.

[10] Joint Video Team of ITU-T and ISO/IEC JTC 1, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification," JVT-L050, July 2004.

[11] Available online: http://www.kakadusoftware.com

[12] Available online: http://iphome.hhi.de/suehring/tml

[13] V. Sanchez, P. Nasiopoulos and R. Abugharbieh, "Lossless compression of 4D medical images using H.264/AVC," *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2006, Toulouse, France.

[14] Information Technology—JPEG 2000 Image Coding System—Part 2: Extensions, ISO/IEC 15 444-2, 2002

[15] G. Sullivan, P. Topiwala and A. Luthra, "The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions", *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 5558, pp. 454-474, August 2004

Chapter 5

5. Novel Lossless fMRI Image Compression Based on Motion Compensation and Customized Entropy Coding[1]

5.1 Introduction

Four dimensional (4D) images are increasingly being collected and used in many clinical and research applications including functional brain imaging and computer assisted intervention. Such 4D medical imaging data have become an important part of modern medical research thanks to the continuous advances in various image acquisition technologies. These technologies include 4D *structural* imaging modalities, such as dynamic magnetic resonance imaging (MRI), dynamic computed tomography (CT), and dynamic three dimensional (3D) ultrasound (US); and 4D *functional* imaging modalities, such as positron emission tomography (PET), dynamic single photon emission computed tomography (SPECT), and functional MRI (fMRI). Four dimensional medical images depict 3D image volumes each comprising two dimensional (2D) image slices depicting cross sections of a region of interest (ROI) that show either anatomy or physiology data over time. These 4D data are huge in file size and thus pose heavy demands on storage and archiving resources. Furthermore, recent advances in telemedicine and tele-consultation and the wide deployment of picture archiving and communications systems (PACS) in clinical settings necessitate that medical images be efficiently transmitted over networks of limited bandwidth.

A typical fMRI scanning session involves temporal acquisition of the 3D volume of the brain every 1-2 seconds and always involves a group of subjects. With at least 8-10 subjects typically scanned, with 200+ megabytes of data per subject, an fMRI study typically results in several gigabytes of imaging data. Lossless compression can reduce the storage and transmission burden of such data, while at the same time avoiding any loss of valuable clinical data which may result in serious clinical and legal implications. Unlike structural medical images, scalable compression is not a main interest in fMRI, as the sequences are usually used in batch to conduct statistical image analysis and are not used individually as single volumes.

Most of the previously reported advances in compression of high-dimensional medical images are aimed at 3D datasets [1-6]. Few methods that exploit data redundancies in the spatial and temporal

[1] A version of this chapter has been published. V. Sanchez, P. Nasiopoulos, and R. Abugharbieh, "Novel Lossless fMRI Image Compression Based on Motion Compensation and Customized Entropy Coding," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 645–655, 2009.

dimensions of 4D medical images have been proposed. Such 4D compression methods use mainly wavelet transforms or prediction coding to decorrelate the data and improve the compression performance. Wavelet-based compression methods that decorrelate the data by applying discrete wavelet transforms were reported in [7-12]. Prediction-coding based methods, which try to minimize the difference between consecutive samples, slices or volumes by using either motion compensation and estimation or differential pulse code modulation, were proposed in [13-15]. Among these, our previously proposed lossless compression method which is based on the H.264/AVC standard reported the highest lossless compression ratios on a wide range of real functional and structural 4D medical data [15].

In this chapter, we extend our previous work presented in Chapter 4 [15] and propose a novel lossless compression method specifically designed for fMRI data. The proposed method employs a new multi-frame motion compensation process which better exploits the spatial and temporal correlations of fMRI data and a new context-based adaptive binary arithmetic coder (CABAC) to losslessly compress the residual and motion vector data generated by the motion compensation process. The proposed multi-frame motion compensation uses a 4D search, bi-directional prediction and variable-size block matching for motion estimation. The proposed CABAC takes into account the probability distribution of the residual and motion vector data in order to assign proper probability models to these data and improve the compression performance.

Performance evaluations of the proposed method on a large sample of real fMRI data of varying resolution demonstrate superior lossless compression ratios compared to two state-of-the-art methods; 4D-JPEG2000 and H.264/AVC [15], with an average improvement of 13%.

The rest of the chapter is organized as follows. We describe our proposed compression method in section 5.2. Performance evaluations and comparisons to the other state-of-the-art are presented in section 5.3 and conclusions are given in section 5.4.

5.2 Proposed fMRI Compression Method

Let us denote an fMRI sequence as $I(x,y,z,t)$, where the variables x and y denote the two spatial dimensions within a slice, the variable z denotes the third spatial dimension within a volume and t denotes the fourth temporal dimension.

Our proposed method is a four stage system as schematically summarized in Fig. 5.1. In *stages I* and *II*, redundancies between slices in the z and t dimensions are reduced by recursively applying multi-frame motion compensation with a 4D search, bi-directional prediction and variable-size block matching for motion estimation. In *stage III*, residuals generated during *stage II* are losslessly encoded using a new CABAC designed for these residual data. In *stage IV*, the motion vector data generated during *stages I* and *II* are first encoded using a differential coding algorithm and the resulting differences are then losslessly compressed using a new CABAC designed for these motion vector data. The coded residuals and motion vector data comprise the final compressed bit-stream. These four stages are explained in detail next.

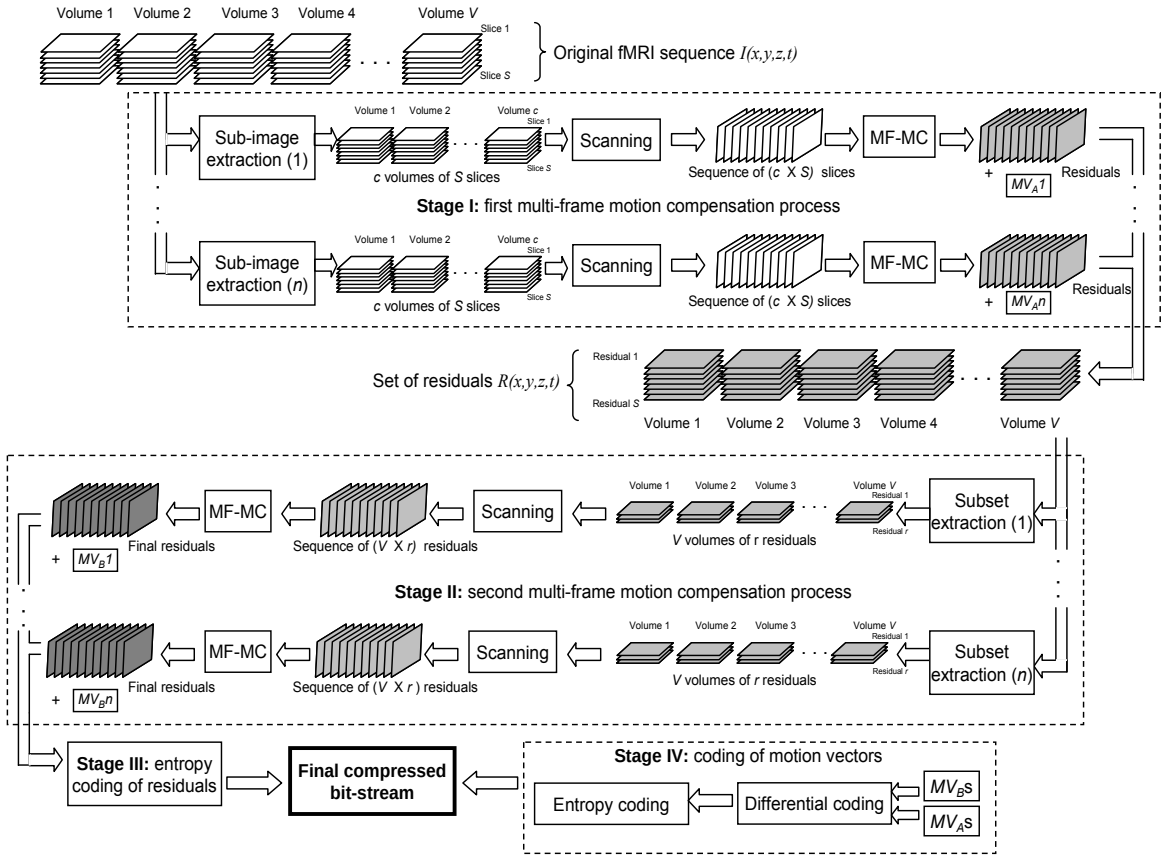


Fig. 5-1: Block diagram of the proposed lossless compression method. Diagram shows the encoding process for an fMRI sequence with V volumes of S slices. MF-MC: multi-frame motion compensation. MV_A: motion vectors produced after applying MF-MC on sub-images (Stage I). MV_B: motion vectors produced after applying MF-MC on subsets of residuals (Stage II).

5.2.1 Stage I: multi-frame motion compensation on slices

Multi-frame motion compensation (MF-MC) is an efficient way to reduce data redundancies in the temporal dimension between frames of video sequences [16-18]. The objective of MF-MC is to estimate the amount of motion on a block-by-block basis which minimizes the difference between two frames.

fMRI sequences represent highly correlated data at consecutive timepoints with limited motion in the temporal dimension t and structural changes in the spatial dimension z. The motion found in the t dimension is due mainly to small head movements of the patient in the scanner and breathing and cardiac cycle related displacements [19]. This motion, which is comparable to the motion found in video sequences of still objects with few scene transitions, may be effectively estimated using MF-MC. Structural changes found in the z dimension, on the other hand, are comparable to the motion found in video sequences with many scene transitions. However, there still exists a high level of redundancy between slices in the z dimension, which may be also exploited using MF-MC. Based on these observations; in *stage I* we first reduce data

redundancies between slices in the spatial dimension z by using a novel MF-MC process. Namely, we improve the coding performance by extending the search area of MF-MC to the t dimension [20]. To achieve this, we divide the fMRI sequence to be coded into smaller sub-images and encode the slices of each sub-image using a specific coding order. Accordingly, sequence $I(x,y,z,t)$ of dimensions $x=X$, $y=Y$, $z=S$ and $t=V$ is divided into sub-images of dimensions $x=X$, $y=Y$, $z=S$ and $t=c$, where $c<V$ and $c<S$. Each sub-image is then processed separately by first scanning it in a raster order (see Fig. 5.2) to create a single sequence of $(c \times S)$ slices denoted as $i(x,y,k)$, where the x and y variables denote the spatial dimensions of the slices and the k variable denotes the position of a slice within the sequence. The slices of sequence i are grouped into groups of slices (GOS) of size g and coded as I-frames, P-frames or bi-directionally as B-frames following coding order W. Coding order W is aimed at exploiting data redundancies between slices of adjacent volumes by coding slices in an order which may differ from the scanning order.

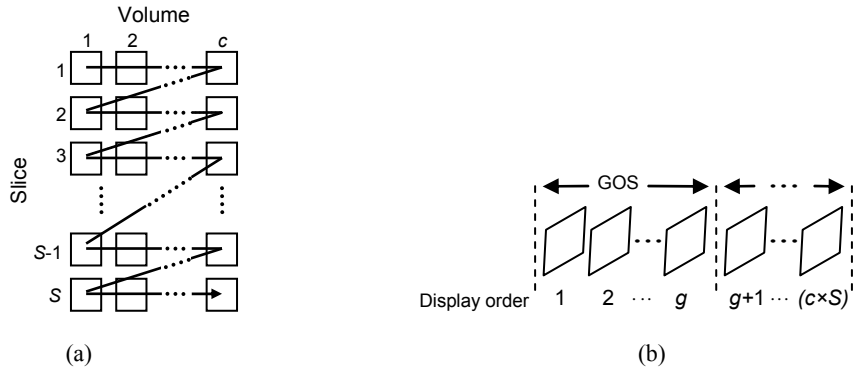


Fig. 5-2: Stage I: coding of slices. Figure shows (a) the raster scanning order followed to scan slices of a sub-image of c volumes of S slices each, and (b) the sequence after scanning with Group of Slices (GOS) of g slices.

A slice is said to be an I-frame if it is used as reference to predict other slices while using no prediction to encode it. Slice k is said to be coded as a P-frame if only previously coded slices of the current or previous GOS at positions preceding slice k are used as reference to predict it. On the other hand, slice k is said to be coded bi-directionally as a B-frame if previously coded slices of the current or previous GOS at positions preceding or exceeding slice k are used as reference to predict it.

Note that by making $c<S$, we force the spatial resolution z (i.e., slices per volume) of the sub-images to be higher than the temporal resolution t (i.e., number of volumes). Thus, when we apply MF-MC to such sub-images, we mainly exploit the redundancies between slices in the z dimension. Depending on the total number of volumes of $I(x,y,z,t)$ the last sub-image may contain fewer volumes than c.

In order to select the values of c and g that best exploit the correlation between slices, we analyze the characteristics of fMRI data. These data usually depict brain activation by measuring the blood-oxygenation-level-dependent (BOLD) signal with a spatial resolution in the region of 3-6 millimeters and a temporal

resolution in the order of seconds. A single fMRI volume is usually acquired within a time of repetition (TR). For example, for 150 fMRI volumes with a typical TR of 2 s, we have a timecourse of 150 volumes of the brain (each consisting of 40-50 slices) acquired over 500 s. Compared to structural MRI, fMRI data present a relatively poorer spatial resolution with a higher temporal resolution [19]. We claim that any slice can be effectively predicted by using its immediate eight neighbor slices as illustrated in Fig. 5.3.

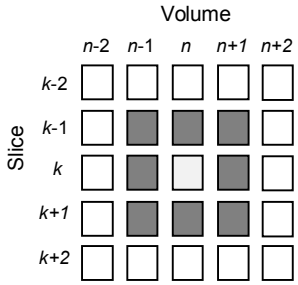


Fig. 5-3: The immediate eight neighbor slices (in gray) of slice *k* of volume *n*.

Let us define *sf* as a measure of how similar slice *a* is to slice *b*:

$$sf = 1 - \frac{e(a-b)}{e(b)} \tag{5.1}$$

where *e(n)* denotes the energy of slice *n* as defined by Eq. (5.2):

$$e(n) = \sum_{x=1}^{X} \sum_{y=1}^{Y} n(x,y)^2 \tag{5.2}$$

where $n(x, y)$ is the value of sample at position (x, y).

If e(a - b) (i.e., the energy of the difference between slices a and b) is much lower than e(b) (i.e., the energy of slice b), sf tends to one and slice a is said to be highly similar to slice b. If e(a─b) is close to e(b), sf tends to zero and slice a is said to be dissimilar to slice b. Finally, if e(a─b) is much higher than e(b), sf tends to negative values and slice a is also said to be dissimilar to slice b.

After computing the similarities between slice *k* of volume *n* and its immediate eight neighbor slices according to Eq. (5.1), we find that for a large set of fMRI sequences of different spatial and temporal resolutions, the value of *sf* for the eight neighbor slices lies in the range of [0.65, 0.75], which substantiates our claim. Therefore, we set the value of *c* to three volumes and the value of *g* to nine slices to exploit these similarities. Note that the last GOS may contain fewer slices than *g* depending on the total number of slices per volume.

In order to define the coding order W that exploits most of the redundancies between slices of each GOS, we take advantage of the bi-directionality of B-frames which allows prediction of slice k using slices at positions preceding or exceeding slice k. Moreover, we allow slices to be coded in any particular order, which may differ from the order they appear in the GOS. After evaluating several coding orders on GOS's of g= 9 slices of a large set of fMRI sequences, the following coding order has shown to produce the smallest residuals (i.e., the residuals with the least amount of energy):

$$W = \begin{cases} \{I^1, B^4, B^6, B^8, B^3, B^7, B^9, B^5, P^2\} & \text{for the } 1^{st} \text{ GOS} \\ \{B^1, B^4, B^6, B^8, B^3, B^7, B^9, B^5, P^2\} & \text{elsewhere} \end{cases} \tag{5.3}$$

Coding order W is a vector of g elements where g is the size of the GOS to be coded. The gth element of W specifies the type of prediction and coding order for the gth slice of the GOS, where I denotes an I-fame, B denotes a B-frame and P denotes a P-frame; while the superscript of each element denotes the coding order of the corresponding slice. For example, the fourth slice of the 1st GOS (B8, fourth element of W) is the eighth slice to be predicted and is encoded as a B-frame. Note that in this particular coding order, slices are not coded in the order they are scanned, but rather following a pattern aimed at exploiting data redundancies between slices of adjacent volumes. The complete scanning process of a sub-image of c = 3 volumes, the 2D sequence generated after the scanning process, the GOS's of g = 9 slices, and coding order W are illustrated in Fig. 5.4.

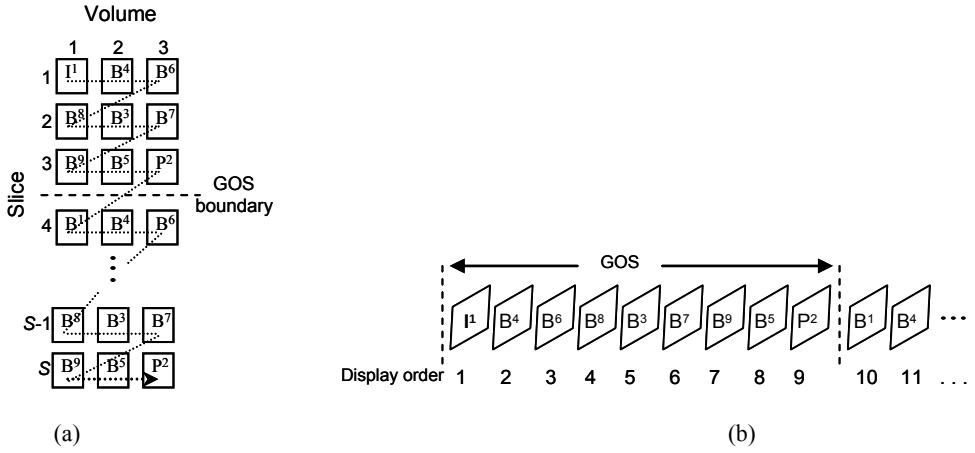


Fig. 5-4: Stage I: coding of slices. (a) Scanning order of a sub-image with three volumes (c=3) of S slices. (b) Sequence after scanning with Group of Slices (GOS) of g=9 slices. I: I-frame. P: P-frame B: B-frame. Superscript of each slice indicates coding order.

It is important to note that coding order W increases the search area of MF-MC to adjacent volumes of fMRI sequences, hence extending the search area to four dimensions. For example, according to Eq. (5.3),

the eighth slice of the 1st GOS (B5, eighth element of W) is the fifth slice to be predicted (see Fig. 5.4(b)) and is encoded as a B slice. This particular slice may use as reference slice I1, from the immediate previous volume, or slices {B3, B4} from the same volume, or slice P2, from the immediate subsequent volume. Therefore, the MF-MC process searches a section of a 4D image for potential references.

According to Eq. (5.3), only the first slice of each sub-image is coded as the reference slice, or I-frame. By limiting the use of I-frames we improve the coding performance, as I-frames usually contain more energy than residuals of P- and B-frames. Slices coded as P- and B-frames are predicted using MF-MC on a block-by-block basis by first dividing them into non-overlapping macroblocks of 16×16 pixels. We employ variable-size block matching (VSBM) to partition macroblocks into smaller blocks of 16×8, 8×16 and 8×8 pixels if that shows improvement in coding performance. A VSBM scheme allows us to use larger blocks in regions of less motion or changes and smaller blocks to represent more complex motion or changes. Our experiments on a large set of fMRI sequences have shown that block sizes smaller than 8×8 pixels do not provide a significant coding improvement, but significantly increase the coding time. This is further discussed in section 5.3.1.

The difference between the position (i,j) of a block in the current slice and the position (i',j') of the matched block in the reference slice is denoted by a motion vector. The matched block is defined as the block that yields the minimum sum-of-absolute differences (SAD) with the current block. Matched blocks of the current slice comprise the current predicted slice, which is subtracted from the current slice to calculate the residual.

Stage I results in I-frames, residuals, motion vectors and the information about the block sizes and references used for MF-MC. The I-frames and residuals are re-arranged back to the original order to create a 4D set of residuals, denoted as $R(x,y,z,t)$, of dimensions $x=X$, $y=Y$, $z=S$ and $t=V$ (see Fig. 5.1). The information about the block sizes and references used for MF-MC is coded using variable length coding and sent separately to the decoder.

5.2.2 Stage II: multi-frame motion compensation on residuals

In *stage I*, most of the data redundancies between slices in the z dimension are reduced by increasing the search area of MF-MC to the t dimension. However, data redundancies may still exist between volumes of $R(x,y,z,t)$, as volumes depict the same ROI at different time points. For this reason, in this second stage we apply a second MF-MC process with VSBM to reduce data redundancies between residuals in the t dimension. Here, we also increase the search area of MF-MC to four dimensions by dividing the set of residuals $R(x,y,z,t)$ into smaller subsets and encode the residuals of each subset using a specific coding order (see Fig. 5.1) [20]. Accordingly, $R(x,y,z,t)$ is divided into subsets of dimensions $x=X$, $y=Y$, $z=r$ and $t=V$, where $r<S$ and $r<V$. Each subset is then processed separately by first scanning it in a raster order (see Fig. 5.5) to create a single sequence of $(V \times r)$ residuals denoted as $j(x,y,k)$, where the x and y variables denote the spatial dimensions of the residuals and the k variable denotes the position of a residual within the sequence. The

residuals of sequence *j* are then grouped into GOS's of size *g* and coded as I-, P- or B-frames following coding order *Y*. Note that by making *r*<*V*, we force the temporal resolution *t* of the subsets to be higher than the spatial resolution *z*. Thus, when we apply MF-MC to such subsets, we mainly exploit the redundancies between residuals in the *t* dimension. Depending on the total number of residuals per volume of *R(x,y,z,t)* the last subset may contain fewer residuals than *r*.

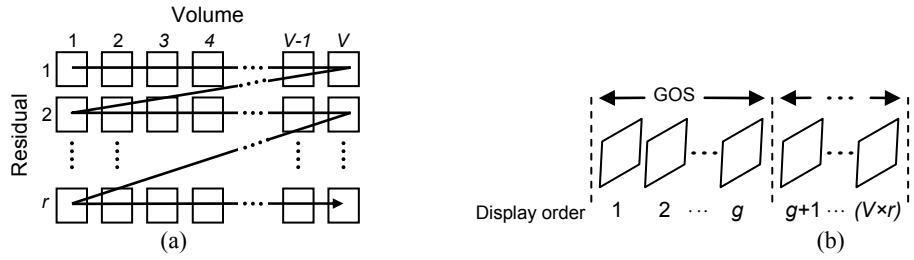


Fig. 5-5: Stage II: coding of residuals. Figure shows (a) the raster scanning order followed to scan residuals of a sub-set of *V* volumes of *r* residuals each, and (b) the sequence after scanning with Group of Slices (GOS) of *g* slices.

After computing the similarities between residual *k* of volume *n* and its immediate eight neighbor residuals according to Eq. (5.1), we find that for a large set of fMRI sequences of different spatial and temporal resolutions, the value of *sf* for the immediate eight neighbor residuals lies in the range of [0.70, 0.85]. Based on these observations, we set the value of *r* to three residuals and the value of *g* to nine residuals to exploit these similarities. Note that the last GOS may contain fewer residuals than *g* depending on the total number of residuals per volume.

Similarly to *stage I*, after the evaluation of several coding orders on GOS's of *g*= 9 residuals of a large set of residuals *R(x,y,z,t)*, the coding order specified by Eq. (5.3) have also shown to produce the residuals with the least amount of energy. Therefore, in *stage II* we employ coding order *Y*=*W* to code residuals as I-, P- or B-frames in a manner similar to that explained in *stage I*.

The complete scanning process of a subset of *r* = 3 residuals, the 2D sequence generated after the scanning process, the GOS's of *g* = 9 residuals, and coding order *Y* are illustrated in Fig. 5.6.

Stage II results in the final residuals, motion vectors and the information about the block sizes and references used for MF-MC. The latter is coded using variable length coding and sent separately to the decoder.

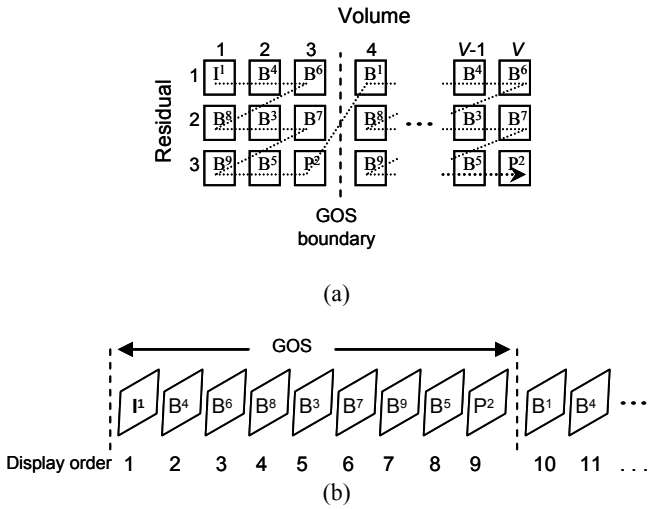


Fig. 5-6: Stage II: coding of residuals. (a) Scanning order of a subset with V volumes of 3 residuals (r=3). (b) 2D sequence after scanning with Group of Slices (GOS) of g=9 residuals. I: I-frame. P: P-frame B: B-frame. Superscript in each type of residual indicates coding order.

5.2.3 Stage III: entropy coding of final residuals

It is common practice to compress the residuals obtained from a MF-MC process using an entropy coding algorithm. One of the most advanced entropy coding algorithms is the context-based adaptive binary arithmetic coder (CABAC) [21]. However, this algorithm is optimized to work with small-sized quantized residual values of video sequences. In order to efficiently compress the final residuals generated in *stage II*, we have developed a new CABAC which is specifically designed for lossless compression of residuals of fMRI data [20].

As described in Chapter 1, section 1.5.3, in the first *code assignment* step of CABAC, a unique binary code is assigned to the magnitude of a decimal value (level, hereafter), which consists of two parts: the prefix and the suffix

Table 5.1 shows the corresponding binary codes for levels from 1 to 19. For levels smaller than 15, the binary codes consist of only a prefix part, while for levels greater than or equal to 15, the binary codes consist of a prefix and a suffix part.

In our proposed compression method no quantization is employed (to ensure lossless compression) and the final residuals may contain high-valued levels, especially in fMRI sequences with low correlation in the z dimension. In this scenario, the original CABAC assigns to levels greater than or equal to 15, binary codes that are comprised of a prefix and a suffix part, resulting in more binary symbols to encode. Additionally, the binary codes used for the suffix part (i.e., zero-order Exp-Golomb codes) minimize the overall code length for quantized residuals [22, 23].

Table 5-1: Binary codes for levels in the **original** CABAC for residual data

Level	**Binary code**	
	Prefix	**Suffix (*zero-order Exp-Golomb code*)**
1	0	
2	1 0	
3	1 1 0	
…	……	
…	……	
13	1 1 1 1 1 1 1 1 1 1 1 1 0	
14	1 1 1 1 1 1 1 1 1 1 1 1 1 0	
15	1 1 1 1 1 1 1 1 1 1 1 1 1 1	0
16	1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 0 0
17	1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 0 1
18	1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 0 0 0
19	1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 0 0 1
…	…… …… ……	……
Bit index (*idx*)	1 2 3 4 5 6 7 8 9 10 11 12 13 14	15 16 17 18 19 20

Careful analysis of the probability distribution of residual values produced in stage II for several fMRI sequences of various spatial and temporal resolutions, led us to two important observations: 1) only 35% of the levels are within the range [1,14], with most of them being greater than or equal to 15; and 2) the probability distribution of the levels greater than or equal to 15 tends to follow a highly peaked Laplacian distribution (more peaked around 15 and with longer tails). Based on these observations, we propose a new CABAC that assigns smaller binary codes to high-valued levels while taking advantage of the original code assignment for small-valued levels. The objective is to improve the coding performance by reducing the amount of the binary symbols that need encoding.

Accordingly, we separate the levels into two sets (type-*A* and type-*B*) according to their value:

$$Type(R) = \begin{cases} A, & if\ R < 15 \\ B, & otherwise \end{cases} \tag{5.4}$$

where R is the current level.

We assign type-*A* levels a code that consists of $m-1$ "1" bits plus a terminating "0" bit, where m is the value of the level, while we assign a second order Exp-Golomb code (EG2) for the value of level−15 of type-*B* levels, which are optimal binary codes for sources with a Laplacian distribution (i.e., they minimize the overall code length) [22, 23].

In order to identify the two types of levels, we introduce an extra bit, where a "0" bit indicates a type-*A* level and a "1" bit indicates a type-*B* level. Table 5.2 shows the corresponding binary codes for levels from 1 to 19.

In the *context modeling and binary arithmetic coding* steps of our proposed CABAC for residuals, we arithmetic code the bits of the binary codes of type-*A* levels with index $idx \in [1,14]$ using the probability models employed in [21], as they provide a good approximation to the probability distributions of these bits. The bits of the binary codes of type-*B* levels with index $idx \geq 1$ are uniformly distributed, as the probability

of encountering a "1" or a "0" bit is roughly the same. In this case, we use a binary arithmetic coder designed to work with a uniformly distributed source.

Table 5-2: Stage III: coding of residual data. Binary codes for levels in the **proposed** CABAC for residual data

Level	**Type**	**Binary code**
		Unary code
1	A	0 0
2	A	0 1 0
3	A	0 1 1 0
…	…	…
…	…	…
13	A	0 1 1 1 1 1 1 1 1 1 1 1 1 1 0
14	A	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
		2nd order Exp-Golomb code
15	B	1 0 0 0
16	B	1 0 0 1
17	B	1 0 1 0
18	B	1 0 1 1
19	B	1 1 0 0 0 0
…	…	…
Bit index (*idx*)		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 …

Bit with index = 0, highlighted in grey, indicates the type of residual value.

Since the extra bit used to indicate the type of level increases the overall bit-rate, we also compress this bit using a binary arithmetic coder. In order to improve the coding performance of this bit, we collect information about the probability of encoding a type-*A* or type-*B* level, based on previously coded levels. We collect this information through a *context modeling stage* where we assign this bit one of three different probability models (indexed probability model 1, 2 and 3). These models exploit the correlation between neighbouring levels, as levels of the same type tend to appear in clusters. To keep the number of probability models as small as possible, instead of employing all surrounding previously coded levels, we only employ the previously coded level to the left and on top of the current level to select the models, as illustrated in 5.7. When a neighbour is not available, as will be the case for levels along the left and top edges of the residuals, or it is equal to zero, a type-*A* level is assumed. Thus, the probability model for encoding the bit with index *idx* = 0 is selected as follows:

$$Model(bit\_index = 0) = \begin{cases} 1, & if\ Type(L) = Type(U) = A \\ 2, & if\ Type(L) \neq Type(U) \\ 3, & if\ Type(L) = Type(U) = B \end{cases} \tag{5.5}$$

where *L* is the immediate level to the left of the current level *R* and *U* is the immediate level on top of the current level *R*.

The performance of our proposed CABAC for residuals is discussed in section 5.4.

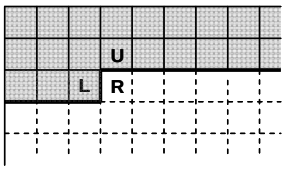


Fig. 5-7: Stage III: coding of residual data. Levels *L* and *U* to the left and on top of the current level *R*. Previously coded levels are highlighted in gray.

5.2.4 Stage IV: coding of motion vectors

The coding gains achieved by employing VSBM in stages I and II may be curtailed by the cost of extra motion vector data. In order to reduce the number of bits needed to represent the motion vector data generated in these two stages, we extend our differential coding algorithm previously proposed in [15] to exploit the correlation in space and time of the motion vectors produced in stages I and II. The spatial correlations are exploited by calculating the difference between motion vectors of neighboring blocks within a single slice, while temporal correlations are exploited by calculating the difference between motion vectors of blocks of two consecutive volumes in the same spatial position. The resulting differences are then entropy coded using a new CABAC for motion vectors (see Fig. 5.1). In the following subsections, we first explain in detail the improvements to our differential coding algorithm, followed by our new CABAC for motion vectors.

Differential coding algorithm for motion vectors: Spatial correlations of motion vectors are first exploited by calculating the spatial motion vector difference (*SMVD*) of the horizontal and vertical components of the motion vector of the current block as defined by Eq. (5.6):

$$SMVD(cmp)=MV(cmp)-MV_{median}(cmp) \qquad cmp\in\{horizontal,vertical\} \tag{5.6}$$

where MV is the motion vector of the current block and MV_{median} is the median of the motion vectors of the blocks immediately above, diagonally above and to the left, and immediately to the left of the current block, as exemplified in Fig. 5.8. If some of the neighbours are not available, for example for blocks along the top and left slice edges, only the available blocks are used to calculate the median.

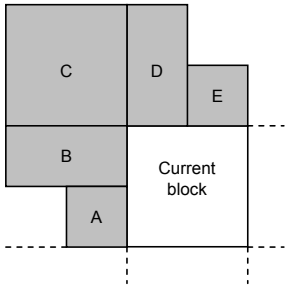


Fig. 5-8: Stage IV: coding of motion vectors. A-E: neighboring blocks used to calculate the spatial motion vector difference, *SMVD,* for the current block.

Volumes of fMRI data usually depict the same anatomical region undergoing certain functional activation in time. If these functional changes are small or constant across volumes, the correlation between motion vectors of two consecutive volumes increases considerably. In order to further reduce the motion vector rate we exploit this temporal correlation by employing our differential coding algorithm proposed in [15]. To this end we first calculate the *SMVD*s of all volumes as defined by Eq. (5.6), and then we calculate the difference between the *SMVD*s of two consecutive volumes.

Let C be the current macroblock of slice k of volume n and let P be the previous macroblock in the same spatial position as C but in slice k of volume $n-1$. The temporal motion vector difference of the qth partition of C ($TMVD_{Cq}$) may be calculated as follows [15]:

$$TMVD_{Cq}(cmp) = SMVD_{Cq}(cmp) - \left\lfloor \sum_{j=1}^{J} SMVD_{Pj}(cmp) \Big/ J \right\rfloor \qquad cmp \in \{horizontal, vertical\} \tag{5.7}$$

where $SMVD_{Cq}$ is the spatial motion vector difference of the qth partition of C, $SMVD_{Pj}$ is the spatial motion vector difference of the jth partition of P located in the same spatial region as the qth partition of C, J is the total number of partitions in P located in the same spatial region as the qth partition of C and $\lfloor x \rfloor$ denotes the largest integer less than or equal to x, as exemplified in Fig. 5.9.

Note that for the first volume, the *TMVD*s are equal to the *SMVD*s.

Proposed CABAC for motion vectors: For the compression of the *TMVD*s, we propose a new CABAC for motion vectors. In the *code assignment* step, we assign the magnitude of a *TMVD* component $cmp \in \{horizontal, vertical\}$ a unique binary code as illustrated in Table 5.3.

Fig. 5-9: Stage IV: coding of motion vectors. *C*: current macroblock in volume *n* and slice *k* with two partitions. *P*: previous macroblock in the same spatial position as *C* in volume $n-1$ and slice *k* with four partitions. The first and third partitions of *P* (highlighted in gray) are used to calculate the temporal motion vector difference (*TMVD*) of the first partition of *C* (highlighted in gray) according to Eq. (7).

Table 5-3: Stage IV: coding of motion vectors. Binary codes for motion vector values in the proposed CABAC for motion vectors

Magnitude of a *TMVD* component *cmp* ∈ {*horizontal*,*vertical*}	**Binary code**
0	0
1	1 0
2	1 1 0
3	1 1 1 0
4	1 1 1 1 0
5	1 1 1 1 1 0
6	1 1 1 1 1 1 0
…	… ……
Bit index (*idx*)	1 2 3 4 5 6 7 8 9 .

In the *context modeling and binary arithmetic coding* steps, we compress the bits of these binary codes using a binary arithmetic coder and a probability model. The first bit (i.e., bit with index *idx* = 1) is compressed using one of three different probability models (indexed probability model 1, 2 and 3) which are selected according to Eq. (5.8) [21]:

$$Model(bit\_index = 1, cmp) = \begin{cases} 1, & if\ AVG(T,L,cmp) < 3 \\ 2, & if\ 3 \le AVG(T,L,cmp) < 8 \\ 3, & if\ AVG(T,L,cmp) \ge 8 \end{cases} \quad cmp \in \{horizontal, vertical\} \tag{5.8}$$

where *AVG(T,L,cmp)* is defined as the average of the absolute value of the *TMVD*s of block *T* and block *L, T* being the topmost block on the left of the current block and *L* being the leftmost block on top of the current block (see Fig. 5.10). When a neighbouring block is not available, as it is the case for blocks along the left and top slice edges, only the available blocks are used.

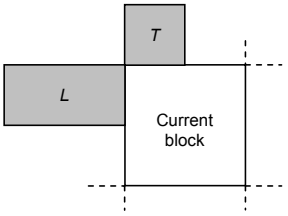


Fig. 5-10: Stage IV: coding of motion vectors. Neighboring blocks at the topmost position on the left (*T*) and at the leftmost position (*L*) of the current block.

For the compression of the bits with indices *idx* ≥2, we define new probability models. Careful analysis of the probability distribution of the *TMVD*s of a large set of fMRI sequences with various spatial and temporal resolutions showed that this probability distribution can be very accurately modeled as a Laplacian distribution with 90% of the magnitudes for both the *horizontal* and *vertical* components in the range [0, 4]. As illustrated in Table 5.3, the magnitudes in this range are assigned binary codes of up to five bits in length, which correspond to the bits with indices 1, 2, 3, 4, and 5. In order to improve the coding performance of the *TMVD*s of fMRI sequences we use two different probability models for each of the bits with index *idx* ∈ [2, 5]. In a *context modeling stage* we use the average value of the *TMVD*s of neighboring blocks to determine the probability of these bits to be "0" or "1" by checking if the corresponding average value is above a predefined threshold. We then use a different probability model to encode the given bit. Let us take for example the bit with index *idx* = 2 of the binary codes illustrated in Table 5.3. This particular bit has a value of "1" for all the binary codes greater than or equal to 2. Similarly, the bit with index *idx* = 3 has a value of "1" for all the binary codes greater than or equal to 3. Based on this observation, we define the threshold to select the appropriate probability model as the value of the index *idx*, as specified by Eq. (5.9):

$$Model(bit\_index = idx, cmp) = \begin{cases} 1, & if\ AVG(T,L,cmp) < idx \\ 2, & if\ AVG(T,L,cmp) \geq idx \end{cases}$$
$$cmp \in \{horizontal, vertical\}$$
$$idx \in [2,5] \tag{5.9}$$

where *AVG(T,L,cmp)* is defined as the average of the absolute value of the *TMVD*s of block *T* and block *L, T* being the topmost block on the left of the current block and *L* being the leftmost block on top of the current block, as illustrated in Fig. 5.10. The performance of our proposed CABAC for motion vectors is discussed in section 5.4.

5.3 Performance Evaluation

We have tested the proposed compression method on eighteen fMRI sequences of various spatial and temporal resolutions. The tested sequences are comprised of volumes of coronal and axial slices of a human brain and depict visual cortex (sequences 1-13) and motor cortex (sequences 14-18) activity. Columns 1 and 2

of Table 5.4 summarize the characteristics of the tested fMRI sequences, while samples of some of these sequences are illustrated in Fig. 5.11.

For comparison purposes, we have also losslessly encoded the fMRI sequences using the H.264/AVC-based method reported in [15] and 4D-JPEG2000, which is a wavelet-based compression method for 4D images and is based on the JPEG2000 standard [24]. We specifically compared our proposed method to these particular compression methods as their coding algorithms are very similar to those employed in the technical advances in lossless compression of 4D medical images reported in the literature [7-15].

For the case of 4D-JPEG2000, we employed two levels of decomposition across all four dimensions. We first applied a one dimensional discrete wavelet transform (1D-DWT) across the t dimension with two levels of decomposition, followed by a 3D-JPEG2000 with two levels of decomposition. We used the Kakadu implementation of 3D-JPEG2000 [25].

a)

b)

Fig. 5-11: Samples of the test sequences. Two slices of two consecutive volumes at the same spatial position of an fMRI sequence of the a) axial view of a human head (128×128 pixels, 12 bits per pixel); and b) coronal view of a human head (128×128 pixels, 12 bits per pixel).

In our proposed compression method, we divided the fMRI sequences into sub-images of three volumes of S slices (c=3, as described in section 5.2.1). Similarly, we divided the 4D set of residuals into subsets of V volumes of three slices (r=3, as described in section 5.2.2). The coding parameters of our proposed compression method are summarized in Table 5.5.

Table 5.4 shows the compression performance of our proposed compression method as well as that of 4D-JPEG2000 and the H.264/AVC-based method. We observe that our proposed compression method outperforms the other compression methods. Although the H.264/AVC-based method is also based on MF-MC, the motion compensation process and entropy coders are designed for compression of video sequences and not fMRI sequences. Our method, on the other hand, is able to exploit data redundancies in the *z* and *t* dimensions of fMRI sequences by applying a MF-MC process with a 4D search specifically designed for these data.

Table 5-4: Compression ratios of fMRI sequences using a H.264/AVC-based method [15], 4D-JPEG2000 and our new compression method.

4D sequence		**Compression method**					
				Proposed compression method			
Modality: ***bits per pixel***	**Size** ***volumes: slices per volume: pixels per slice***	**H.264/AVC-based method [15]**	**4D-JPEG2000**	**New MF-MC, no DCMV and original CABAC for MVs and residual data**	**New MF-MC, DCMV and original CABAC for MVs and residual data**	**New MF-MC, DCMV, proposed CABAC for MVs and original CABAC for residual data**	**New MF-MC, DCMV, proposed CABAC for MVs and residual data**
1. fMRI:12	51:36:128×128	5.95:1	5.51:1	5.93:1	5.97:1	6.08:1	**6.22:1**
2. fMRI:12	60:45:128×128	6.11:1	5.71:1	6.07:1	6.13:1	6.16:1	**6.46:1**
3. fMRI:12	60:45:128×128	6.03:1	5.42:1	5.81:1	5.97:1	6.01:1	**6.37:1**
4. fMRI:12	45:60:256×256	6.09:1	5.78:1	6.08:1	6.12:1	6.15:1	**6.45:1**
5. fMRI:12	45:60:256×256	5.89:1	5.67:1	5.82:1	5.9:1	5.92:1	**6.34:1**
6. fMRI:12	126:36:128×128	10.53:1	11.95:1	12.15:1	12.39:1	12.47:1	**13.14:1**
7. fMRI:12	126:36:128×128	11.41:1	12.54:1	13.19:1	13.42:1	13.89:1	**14.34:1**
8. fMRI:12	126:36:128×128	11.55:1	12.82:1	13.37:1	13.59:1	13.97:1	**14.47:1**
9. fMRI:12	195:36:128×128	13.35:1	14.89:1	15.61:1	15.71:1	16.12:1	**16.51:1**
10. fMRI:12	195:36:128×128	13.81:1	14.27:1	14.59:1	14.76:1	14.89:1	**15.63:1**
11. fMRI:16	198:45:128×128	13.15:1	12.79:1	13.05:1	13.74:1	13.89:1	**14.10:1**
12. fMRI:16	198:45:128×128	12.25:1	11.59:1	11.98:1	12.32:1	12.56:1	**13.21:1**
13. fMRI:16	198:45:128×128	12.38:1	11.05:1	11.11:1	12.53:1	12.78:1	**13.78:1**
14. fMRI:12	130:36:128×128	10.11:1	11.16:1	11.43:1	11.74:1	11.93:1	**12.47:1**
15. fMRI:12	130:36:128×128	10.25:1	10.78:1	11.02:1	11.31:1	11.51:1	**12.09:1**
16. fMRI:12	130:36:128×128	10.18:1	11.01:1	11.25:1	11.59:1	11.82:1	**12.35:1**
17. fMRI:12	130:36:128×128	10.21:1	10.86:1	11.11:1	11.46:1	11.64:1	**12.18:1**
18. fMRI:12	130:36:128×128	10.13:1	10.72:1	10.96:1	11.31:1	11.46:1	**12.01:1**

MF-MC: multi-frame motion compensation. DCMV: differential coding of motion vectors. MV: motion vector. CABAC: context-based adaptive binary arithmetic coder.

Table 5-5: Coding parameters for the proposed compression method

Stage	Parameter	Value
I	*c*: volumes per sub-image	3
	g: size of GOS	9
	W: coding order	$\{I^1, B^4, B^6, B^8, B^3, B^7, B^9, B^5, P^2\}$ for 1st GOS $\{B^1, B^4, B^6, B^8, B^3, B^7, B^9, B^5, P^2\}$ elsewhere
II	*r*: residuals per subset	3
	g: size of GOS	9
	Y: coding order	$\{I^1, B^4, B^6, B^8, B^3, B^7, B^9, B^5, P^2\}$ for 1st GOS $\{B^1, B^4, B^6, B^8, B^3, B^7, B^9, B^5, P^2\}$ elsewhere

GOS: group of slices. I: I-frame. P: P-frame. B: B-frame.
Superscript of each slice indicates coding order.

Column 5 of Table 5.4 shows the compression ratios obtained by our method employing only our MF-MC process, while using the original CABAC for residuals and motion vectors, and without our differential coding method for motion vectors (motion vectors were directly entropy coded using the original CABAC for motion vectors). We observe that the resulting compression ratios are up to 7% better than those achieved by 4D-JPEG2000 (sequences 1 and 3).

Column 6 corresponds to the compression ratios achieved when our compression method employs, in addition to our MF-MC process, our differential coding method for motion vectors (while employing the original CABAC for compressing the residuals and the motion vectors). Compared to the compression ratios shown in column 5, these results show an improvement of up to 12%. As the number of volumes increases, the correlation between motion vectors of two consecutive volumes increases (for example, sequence 13) and the amount of bits needed to compress these motion vectors is reduced by applying our differential coding algorithm.

Column 7 shows the compression ratios obtained by our compression method employing our MF-MC process, our differential coding method for motion vectors and our proposed CABAC for motion vectors (while employing the original CABAC for compressing the residuals). It can be observed that our proposed CABAC for motion vectors provides bit savings of up to 3% (sequence 7) compared to the original CABAC for motion vectors (column 6).

Finally, column 8 shows the compression ratios obtained when employing our MF-MC process, our differential coding method for motion vectors and our proposed CABAC for motion vectors and residuals. Note that, compared to results on column 7, our proposed CABAC for residuals provide an improvement of up to 7% (sequences 5 and 13). Overall, our proposed method achieves an average improvement on lossless compression ratio of 13% compared to 4D-JPEG2000 and the H.264/AVC-based method.

5.3.1 Complexity of the proposed compression method

We conclude our performance evaluation with a brief discussion regarding the complexity of the evaluated compression methods. In terms of memory requirements, 4D-JPEG2000 requires a buffer capable to store *V*

samples (where V is the total number of volumes of the fMRI sequence) to perform the 1D-DWT across the t dimension, and a buffer capable to store S samples (where S is the total number of slices in a volume) to perform the 1D-DWT across the z dimension. Our proposed compression method requires a buffer capable to store $2 \times g$ slices (where g is the size of the GOS); since slices may be predicted using previously coded slices of the current or previous GOS.

In terms of computational requirements, the number of arithmetic operations needed to perform a 1D-DWT increases as the number of decomposition levels increases. 4D-JPEG2000 performs these operations for each 1D-DWT applied across the t, z, x and y dimensions. In contrast, the proposed method performs an exhaustive search (based on SAD) amongst all previously coded slices in the current and previous GOS to select the best predictor. This search, which is performed in *stages I* and *II* for each 16×16 block and the corresponding 8×16 and 8×8 partitions, considerably increases the complexity and coding time of the proposed method. The complexity of the decoder and the decoding time, on the other hand, are much lower since no exhaustive search is performed at the decoder. In medical imaging applications, the compression efficiency as well as the complexity of the decoder and the decoding time play an important role, since compressed medical images are usually stored and maintained on a server, so they can be accessible by a number of different clients.

Performance evaluations on a large set of fMRI sequences have shown that using blocks smaller than 8×8 pixels result in a mere 0.04% bit rate reduction on average. Based on these findings, and the fact that this is a very small improvement compared to the complexity added by the exhaustive search needed to perform MF-MC for the smaller block partitions, we decided to terminate partitioning at the 8×8 pixel level.

5.4 Conclusions

We proposed a new lossless compression method for fMRI data based on multi-frame motion compensation process. The proposed method effectively reduces data redundancies in the spatial and temporal dimensions by employing a 4D search, variable-size block matching and bi-directional prediction for motion estimation. Correlations between motion vectors in the spatial and temporal dimensions are exploited by employing a differential coding algorithm. Residual and motion vector data are losslessly compressed using a new context-based adaptive binary arithmetic coder designed based on the probability distribution of the data. Evaluation results show an average improvement on lossless compression ratio of 13% on real fMRI data when compared to 4D-JPEG2000 and H.264/AVC. Future work includes the design of a multi-frame motion compensation process and entropy coders for various 4D medical imaging modalities.

5.5 References

[1] Z. Xiong, X. Wu, S. Cheng and J. Hua, "Lossy-to-lossless compression of medical volumetric images using three-dimensional integer wavelet transforms," *IEEE Trans. on Medical Imaging*, vol. 22, no. 3, pp. 459-470, March 2003.

[2] X. Wu and T Qiu, "Wavelet coding of volumetric medical images for high throughput and operability," *IEEE Trans. on Medical Imaging*, vol. 24, no. 6, pp. 719-727, June 2005.

[3] E. Siegel, K. Sidddiqui and J. Johnson, "Compression of multi-slice CT: 2D vs. 3D JPEG2000 and effects of slice thickness", *Proceedings of SPIE Int. Soc. Opt. Eng.,* vol. 5748, pp. 162-170, 2005.

[4] G. Menegaz and J.P. Thirian, "Three-dimensional encoding/two-dimensional decoding of medical data," *IEEE Trans. on Medical Imaging,* vol. 22, no. 3 , pp. 424-440, March 2003.

[5] M. Benetiere, V. Bottreau, A. Collet-Billon and T. Deschamps, "Scalable compression of 3D medical datasets using a (2D+T) wavelet video coding scheme," *Sixth International Symposium on Signal Processing and its Applications,* vol. 2, pp. 537 - 540, Aug. 2001.

[6] R. Srikanth and A.G. Ramakrishnan, "Contextual Encoding in Uniform and Adaptive Mesh-Based Lossless Compression of MR Images," *IEEE Trans. on Medical Imaging*, vol. 24, no. 9, pp. 1199-1206, September 2005.

[7] L. Zeng, C. Jansen, M. Unser and P. Hunziker, "Extension of wavelet compression algorithms to 3D and 4D image data: exploitation of data coherence in higher dimensions allows very high compression ratios", *Proceedings of SPIE Int. Soc. Opt. Eng.,* vol. 4478, pp. 427-433, 2001.

[8] L. Zeng, C.P. Jansen, S. Marsch, M. Unser, and P.R. Hunziker, "Four-dimensional wavelet compression of arbitrarily sized echocardiographic data," *IEEE Trans. on Medical Imaging,* vol. 21, no. 9, pp. 1179-1187, September 2002.

[9] H. G. Lalgudi, A. Bilgin, M. W. Marcellin, and M. S. Nadar, "Compression of fMRI and ultrasound images using 4D-SPIHT," *Proceedings of 2005 International Conference on Image Processing*, vol. 2, pp. 11-14, September 2005.

[10] L. Ying, and W.A. Pearlman, "Four-dimensional wavelet compression of 4-D medical images using scalable 4D-SBHP," *2007 Data Compression Conference*, pp. 233-242, March 2007.

[11] N. Zhang, M. Wu, S. Forchhammer and X. Wu, "Joint compression-segmentation of functional MRI data sets," *Proceedings of SPIE Int. Soc. Opt. Eng.,* vol. 5748, pp. 190-201, 2005

[12] L. Zhang and X. Wu, "An efficient lossless compression algorithm for fMRI data volume," *2005 IEEE Engineering in Medicine and Biology,* pp. 3093-3096, September 2005.

[13] F. J. Theis and T. Tanaka, "A fast and efficient method for compressing fMRI data sets," *Lecture Notes in Computer Science,* vol. 3697, pp. 769-777, August 2005

[14] A. Kassim, P. Yan, and W. S. Lee, "Motion compensated lossy-to-lossless compression of 4-D medical images using integer wavelet transforms," *IEEE Trans. on Information Technology in Biomedicine,* vol. 9, no. 1, pp. 132-138, March 2005

[15] V. Sanchez, P. Nasiopoulos and R. Abugharbieh, "Efficient lossless compression of four dimensional medical images based on the advanced video coding scheme," *IEEE Trans. on Information Technology in Biomedicine,* vol. 12, no. 4, pp. 442-446, July 2008

[16] T. Sikora, "MPEG digital video-coding standards," *IEEE Signal Preocessing Magazine,* vol. 14, no. 5, pp. 82-100, September 1997.

[17] G. Sullivan, P. Topiwala and A. Luthra, "The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions", *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 5558, pp. 454-474, August 2004.

[18] M. Budagavi and J.D. Gibson, "Multiframe video coding improved performance over wireless channels", *IEEE Trans. On Image Processing,* vol. 10, no. 2, pp. 252-265, February 2001.

[19] J. J. Pekar. "A Brief Introduction to Functional MRI," *IEEE Engineering in Medicine and Biology,* vol. 25, no. 2, pp. 24-26, March/April 2006.

[20] V. Sanchez, P. Nasiopoulos and R. Abugharbieh, "Efficient 4D Motion Compensated Lossless Compression of Dynamic Volumetric Medical Image Data," *2008 International Conference of Acoustics, Speech and Signal Processing (ICASSP)*, pp. 549-552, April 2008.

[21] D. Marpe, H. Schwarz and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard", *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 1, no. 7, pp. 620-623, July 2003.

[22] R. Gallager and D.Van Voorhis, "Optimal source codes for geometrically distributed integer alphabets", *IEEE Transactions on Information Theory*, vol. 21, pp. 228-230, March 1975

[23] J. Teuhola, "A compression method for clustered bit-vectors," *Information Processing Letters*, vol. 7, pp. 308-311, October 1978.

[24] Information Technology—JPEG 2000 Image Coding System—Part 2: Extensions, ISO/IEC 15 444-2, 2002.

[25] Available online: http://www.kakadusoftware.com

[26] O.K. Al-Shaykh and R. M. Mersereau, "Lossy compression of noisy images," *IEEE Trans. on Image Processing,* vol. 7, pp. 1641-1652, December 1998

Chapter 6

6. Discussion and Conclusions

6.1 Significance of the Research

In this thesis, we proposed several lossless medical image compression techniques for efficient storage, transmission and on demand scalable access for 3D and 4D data [1-3]. We approached the problem from an energy reduction perspective, in which the main objective is to reduce the overall energy of the medical imaging data, and thus attain a higher compression performance.

In the introduction, we discussed how the continuously improving visual quality of 3D and 4D medical imaging data poses a big burden on computational resources needed to store, access and transmit massive amounts of data for clinical use and research studies. We also discussed the importance of lossless compression to reduce the storage and transmission burden of such data, while at the same time avoiding any loss of valuable clinical data, which may result in serious clinical and legal implications. We introduced the basic components and requirements of lossless medical image compression methods and described state-the-art compression principles suitable for 3D and 4D medical images.

We first addressed the issue of efficient scalable lossless compression of 3D medical images. In Chapter 2, we showed that the integer wavelet transform (IWT) is a suitable data decorrelation technique to provide scalable compression by resolution and quality. Here, we presented a scalable lossless compression method based on a block-based intra-band prediction method that reduces the energy of wavelet sub-bands according to the symmetrical features of the region of interest (ROI) depicted in most 3D medical images. The significance of this compression method is the ability to produce a scalable bit-stream that offers the possibility to access and transmit the data at different qualities or resolutions to clients with limited bandwidth using, for example, a remote image retrieval system.

Next, in Chapter 3, we addressed the problem of ROI coding in 3D medical image compression. Here, we presented a lossless compression method that optimizes the reconstruction quality of a volume of interest (VOI) at any bit-rate, while incorporating partial background information and allowing for gradual increase in possibility peripheral quality around the VOI. The significance of this compression method resides in the possibility of encoding a VOI at a higher quality than the rest of the image, while improving the overall reconstruction quality and allowing for placement of the VOI into the context of the 3D image by including partial background information along with the VOI.

In Chapter 4 and 5, we addressed the issue of efficient lossless compression of 4D medical images, in particular functional Magnetic Resonance Imaging (fMRI) sequences, which are increasingly being collected and used in many clinical and research applications. First, we showed that motion compensation and

estimation, a video compression method, is an efficient technique to exploit redundancies in all four dimensions of 4D medical imaging data. In particular, we proposed a lossless compression method based on the most advanced features of the H.264/AVC (Advanced Video Coding) standard, namely multi-frame motion compensation, variable block-sizes for motion estimation and sub-pixel motion vector accuracy. We then presented a novel lossless compression method specifically designed for fMRI data. This particular method employs a new multi-frame motion compensation process which better exploits the spatial and temporal correlations of fMRI data and a new context-based adaptive binary arithmetic coder (CABAC) to losslessly compress the residual and motion vector data generated by the motion compensation process. The significance of this compression method is the higher lossless compression ratios attained when compared to the state-of-the-art methods 4D-JPEG2000 and H.264/AVC.

6.2 Contributions

The main contributions of this thesis are summarized as follows:

- We showed that the IWT is an efficient way to attain resolution and quality scalability for 3D medical image lossless compression.
- We proposed a scalable lossless compression method for 3D medical images that exploits the symmetrical characteristics of the data to achieve a higher lossless compression ratio. The proposed method employs 2D wavelet-based compression of slices within a 3D medical image. Specifically, it encodes slices by first applying a 2D-IWT, followed by block-based intra-band prediction of the resulting sub-bands. The block-based intra-band prediction method exploits the structural symmetry of the ROI depicted by the image data to reduce the energy of the sub-bands. Residual data generated by the intra-band prediction method are then compressed using a modified version of the EBCOT algorithm designed according to the characteristics of the residual data.
- We proposed a 3D scalable medical image compression method with optimized VOI coding. The proposed method reorders the output bit-stream after encoding, so that those bits with greater contribution to the distortion of a VOI and are included earlier while simultaneously maximizing the overall reconstruction quality of the 3D image. In other words, the proposed method is designed to optimize the reconstruction quality of a VOI at any bit-rate, while incorporating partial background information and allowing for gradual increase in peripheral quality around the VOI. The method employs the 3D-IWT and a modified EBCOT algorithm with 3D contexts. The bit-stream reordering procedure is based on a weighting model that incorporates the position of the VOI and the mean energy of the wavelet coefficients to create an optimized scalable layered bit-stream.
- We showed that multi-frame motion compensation and estimation is an efficient way to reduce the data redundancies of slices of 4D medical images within each volume and between volumes. Specifically,

we proposed a lossless compression method based on the motion compensation process of the H.264/AVC standard.

- We proposed a lossless compression method specifically designed for fMRI data. The proposed method employs a new multi-fame motion compensation process, which efficiently exploits the spatial and temporal correlations of fMRI data, and a new CABAC to losslessly compress the residual and motion vector data generated by the motion compensation process. The proposed multi-frame motion compensation uses a 4D search, bi-directional prediction and variable-size block matching for motion estimation. The proposed CABAC takes into account the probability distribution of the residual and motion vector data in order to assign proper probability models to these data and improve the compression performance.

6.3 Future Research

We envision several ideas with which this research can be extended in order to improve and complement the current proposed compression algorithms.

6.3.1 Coding and transmission of 3D medical images over wireless networks

In the case of the 3D medical image compression, the proposed compression methods presented in Chapters 2 and 3 can be extended for transmission over error-prone wireless networks commonly encountered in telemedicine applications. With the advent use of telemedicine, mobile devices, such as Personal Digital Assistants (PDAs), have been recently incorporated into current practice, picture archiving and communication systems (PACS) in order to allow immediate diagnosis by a doctor at any time and in any place [4]. Consequently, telemedicine applications require that medical imaging data be efficiently accessed and transmitted over error-prone wireless networks of various bandwidth capacities.

Most of the work on medical imaging data access using mobile devices has focused on enabling the visualization of such data at the mobile devices, and little work has been done on designing coding and error-protection techniques to access and transmit such data over error-prone wireless networks [5-7]. Scalable object-oriented coding in conjunction with error- protection techniques are suitable coding techniques for medical images, as these data usually depict one or more clinically relevant ROIs over a clinically irrelevant background. In this way, the ROIs and background may be separately coded at different qualities, and then transmitted using different error-protection levels. The latter can guarantee the reconstruction of the ROIs even in the presence of transmission errors.

6.3.2 Customized motion compensation and entropy coding for 4D medical imaging data

In the case of 4D medical image compression, the proposed compression method presented in Chapter 5 can be extended for different types of 4D medical images, e.g., dynamic positron emission tomography (PET) sequences or dynamic computed tomography (CT). Specifically, it would be worth investigating the feasibility of designing a motion compensation process capable to exploit the spatial and temporal correlations of different types of 4D medical images by adapting the search and prediction algorithms according to the data.

Another interesting improvement to the proposed 4D medical image compression method is the introduction of resolution and quality scalability. Resolution scalability may be achieved by employing a wavelet transform on the residual data obtained after the motion compensation process. Quality scalability, on the other hand, may be achieved by employing a quantization process to generate a number of quality layers.

6.4 References

[1] V. Sanchez, R. Abugharbieh and P. Nasiopoulos, "Symmetry-Based Scalable Lossless Compression of 3D Medical Image Data," *IEEE Trans. on Medical Imaging*, vol. 28, no. 7, pp. 1062-1072, July 2009

[2] V. Sanchez, P. Nasiopoulos, R. Abugharbieh, "Efficient Lossless Compression of 4D Medical Images Based on the Advanced Video Coding Scheme," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 4, pp. 442–446, 2008

[3] V. Sanchez, P. Nasiopoulos, R. Abugharbieh, "Novel Lossless fMRI Image Compression Based on Motion Compensation and Customized Entropy Coding," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 645–655, 2009

[4] N. Strickland, "PACS (picture archiving and communication systems): filmless radiology," *Archives of Disease in Childhood,* vol. 1, no. 83, pp. 82–86, July 2000

[5] R. Andrade, A. Wangenheim, M.K. Bortoluzzi, "Wireless and PDA: a novel strategy to access DICOM-compliant medical data on mobile devices," *International Journal of Medical Informatics*, vol. 71, no. 3, pp. 157-163, 2003

[6] R.B. Jones, S.M. McGhee, D. McGhee, "Patient on-line access to medical records in general practice," *Health Bulletin*, vol. 2, no. 50, pp. 143-50, March 1992

[7] C. Pyper, J. Amery, M. Watson, C. Crook, "Patients' experiences when accessing their on-line electronic patient records in primary care," *British Journal of General Practice*, vol. 54, no. 498, pp. 38-43, January 2004