IPL
escola superior de tecnologia e gestão
instituto politécnico de leiria

DISSERTATION

Master in Electrical and Electronic Engineering

Improving Minimum Rate Predictors Algorithm for Compression of Volumetric Medical Images

João Miguel Pereira da Silva Santos

Leiria, March 2016

escola superior de tecnologia e gestão
instituto politécnico de leiria

DISSERTATION

Master in Electrical and Electronic Engineering

Improving Minimum Rate Predictors Algorithm for Compression of Volumetric Medical Images

João Miguel Pereira da Silva Santos

Master dissertation performed under the guidance of Doctor Sérgio Manuel Maciel Faria, Professor at Escola Superior de Tecnologia e Gestão of Instituto Politécnico de Leiria and with the co-orientation of Doctor Nuno Miguel Morais Rodrigues, Professor at Escola Superior de Tecnologia e Gestão of Instituto Politécnico de Leiria.

Leiria, March 2016

If you can dream it,
you can do it.

(Walt Disney)

Acknowledgments

I would like to thank everyone that helped during this research work, and that made it possible to accomplish.

First to my advisers, Dr. Sérgio Faria and Dr. Nuno Rodrigues, that have chosen to introduce me to the research environment. I am thankful for their guidance and support that were essential for the proper development of this research. Also, I would like to thank their patience and availability to review my written works, both in papers and this dissertation. Their valuable insights helped improve the quality of this work.

This work was a part of the UDICMI project funded by CENTRO-07- ST24-FEDER-002022 of QREN and project Medicomp in the scope of R&D Unit 50008, financed by the applicable financial framework (FCT/MEC through national funds and when applicable co-funded by FEDER – PT2020 partnership agreement). I would like to thank to the members of these projects. Namely, Dr. Luís Cruz whose outside insights often helped to focus this work.

To Instituto de Telecomunicações and Escola Superior de Tecnologia e Gestão of Instituto Politécnico de Leiria for the excellent working conditions that certainly made this work much more productive. And to the local Multimedia Signal Processing group. Namely, André Guarda, Filipe Gama, Gilberto Jorge, João Carreira, Lino Ferreira, Luís Lucas, Ricardo Monteiro and Sylvain Marcelino. They made my time here much more enjoyable, with their friendship, great working environment and fruitful discussions. A special thanks must be given to Luís Lucas for his patience in Linux related questions and overall cluster management.

To all my friends, their patience and support during this time was amazing and helped me to focus on this work. A special mention is due to some of the friends I made at IPL. Namely, David Cruz, Hélder Simões, José Ricardo, Miguel Rasteiro and Ricardo Santos. We were always there for each other in the best and worse moments, and certainly have the memories to prove it.

To my family, their love and support were invaluable in the length of this work. They patience is finally being rewarded.

Last but not the least, to Diana. Her unconditional friendship, love and patience are beyond everything I could possible expect. She always had the right words in the most difficult moments, even when this work did not allow us to be together as often as we would like.

Abstract

Medical imaging technologies are experiencing a growth in terms of usage and image resolution, namely in diagnostics systems that require a large set of images, like CT or MRI. Furthermore, legal restrictions impose that these scans must be archived for several years. These facts led to the increase of storage costs in medical image databases and institutions. Thus, a demand for more efficient compression tools, used for archiving and communication, is arising.

Currently, the DICOM standard, that makes recommendations for medical communications and imaging compression, recommends lossless encoders such as JPEG, RLE, JPEG-LS and JPEG2000. However, none of these encoders include inter-slice prediction in their algorithms.

This dissertation presents the research work on medical image compression, using the MRP encoder. MRP is one of the most efficient lossless image compression algorithm. Several processing techniques are proposed to adapt the input medical images to the encoder characteristics. Two of these techniques, namely changing the alignment of slices for compression and a pixel-wise difference predictor, increased the compression efficiency of MRP, by up to 27.9%.

Inter-slice prediction support was also added to MRP, using uni and bi-directional techniques. Also, the pixel-wise difference predictor was added to the algorithm. Overall, the compression efficiency of MRP was improved by 46.1%. Thus, these techniques allow for compression ratio savings of 57.1%, compared to DICOM encoders, and 33.2%, compared to HEVC RExt Random Access. This makes MRP the most efficient of the encoders under study.

Keywords: DICOM, Lossless Compression, Medical Imaging, MRP.

Resumo

As tecnologias de imagens médicas têm vivido um crescimento, quer em termos do uso de imagens quer em termos de resolução das mesmas, nomeadamente em sistemas de diagnóstico que requerem um grande conjunto de imagens, como a tomografia computorizada ou a ressonância magnética. Além disso, restrições legais impõem que este tipo de exames devam ser arquivados durante vários anos. Estes factos levam ao aumento dos custos de armazenamento em bases de dados ou instituições de imagens médicas. Portanto, existe uma necessidade crescente de ferramentas de compressão mais eficientes, usadas quer para armazenamento, quer para comunicação.

Atualmente, a norma DICOM, que faz recomendações para comunicações e compressão de imagens médicas, recomenda codificadores sem perdas, como o JPEG, RLE, JPEG-LS e JPEG2000. No entanto, nenhum destes codificadores inclui técnicas de predição *inter-slice* nos seus algoritmos.

Esta dissertação apresenta o trabalho de pesquisa efetuado sobre compressão de imagens médicas, usando o codificador MRP. O MRP é um dos algoritmos mais eficientes para compressão sem perdas de imagem. Inicialmente, são propostas várias técnicas de processamento para adaptar as imagens médicas às características do codificador. Duas destas técnicas (mudar o alinhamento das fatias para a compressão e o preditor de diferença pixel a pixel) aumentaram a eficiência da compressão do MRP até 27,9%.

Técnicas de predição *inter-slice*, tanto uni como bi-direccionais, foram desenvolvidas para o algoritmo MRP. Para além disso, o preditor de diferença pixel a pixel foi também adicionado ao algoritmo. Assim, a eficiência da compressão do MRP foi melhorada em 46,1%. Estas técnicas permitem ganhos na taxa de compressão de 57,1%, em comparação com os codificadores DICOM, e 33,2%, em comparação com o HEVC RExt Random Access. Estes resultados fazem do MRP o mais eficiente dos codificadores em estudo.

Palavras-chave: DICOM, Compressão sem Perdas, Imagens Médicas, MRP.

Contents

List of Figures

List of Tables

List of Abbreviations

2D:	Two Dimensional, p. 9
3D:	Three Dimensional, p. 15
bpp:	bits-per-pixel, p. 25
CALIC:	Context based Adaptive Lossless Image Codec, p. 2
CC:	Correlation Coefficient, p. 43
CIPR:	Center for Image Processing Research, p. 8
CR:	Computed Radiography, p. 55
CT:	Computed Tomography, p. 1
CTU:	Coding Tree Unit, p. 21
DCT:	Discrete Cosine Transform, p. 20
DICOM:	Digital Imaging and Communications in Medicine, p. 1
DPCM:	Differential Pulse Code Modulation, p. 17
DWT:	Discrete Wavelet Transform, p. 15
EBCOT:	Embedded Block Coder with Optimized Truncation, p. 16
GAP:	Gradient-Adjusted Predictor, p. 11
HEVC:	High Efficiency Video Coding, p. 2
HEVC RA:	HEVC Random Access, p. 25
HEVC RExt:	HEVC Range Extension, p. 22
HOP:	Hierarchical Oriented Prediction, p. 30
ISO:	International Organization for Standardization, p. 9
ITU:	International Telecommunication Union, p. 9
JP3D:	JPEG2000 Part 10, p. 2
JPEG:	Joint Photographic Experts Group, p. 2
JPEG-LS:	Lossless and near-lossless compression of continuous-tone still images, p. 2

LOCO-I:	LOw COmplexity LOssless COmpression for Images, p. 11
LSP:	Least Squares Prediction, p. 17
MDL:	Minimum Description Length, p. 33
MED:	Median Edge Detector, p. 13
MMP:	Multi-scale Multidimensional Parser, p. 2
MPEG:	Motion Picture Experts Group, p. 63
MRI:	Magnetic Resonance Imaging, p. 1
MRP:	Minimum Rate Predictors, p. 2
OCP:	Optimal Compression Plane, p. 43
RLE:	Run Length Encoding, p. 2
ROI:	Region of Interest, p. 27
TSGD:	Two-Sided Geometric Distribution, p. 13
US:	Ultrasound, p. 1
VBS:	Variable Block Size, p. 22

Chapter 1

Introduction

1.1 Context and motivation

Since medical imaging was first used for diagnostics purposes it has been an ever expanding field. The need for better medical diagnosis has driven the medical imaging technologies in the last decades. The expansion in this technology has led to an increasing demand for the use of such diagnostic tools, revolutionising the practice of medicine.

Concurrently, the advance in technology has led not only to new medical imaging technologies, but also to better image quality in several imaging types. Therefore, currently medical images have higher resolution and bit depths. Nowadays, resolutions of 512×512 pixels are considered to be the minimum, although most recent scanning systems are able to capture slices with spatial resolutions up to 1024×1024 pixels. Also, in such exams as Computed Tomography (CT), Magnetic Resonance Imaging (MRI) or Ultrasound (US), that are comprised by several slices, thin-slice scanning techniques have been arising. Thus, leading to the increase of the number of slices in volumetric sequences, as the inter-slice resolution has improved from typically 5mm to 0.6mm over the years [1].

Due to medical and legal reasons [2] the results of these scans need to be kept and archived for several years. Moreover, the archiving process needs to guarantee that the scans are kept in same state as during the diagnosing process. This is relevant, for instance, for cases where judicial proceedings due to medical malpractice, or others, might arise, but also for patient record keeping.

Consequently, medical images archiving databases are facing a quasi-exponential growth in their contents [2]. Therefore, a new field in medical imaging compression has emerged, in order to efficiently archive and transmit the ever growing medical imaging databases information. Currently, Digital Imaging and Communications in Medicine (DICOM), the organism that makes recommendations for medical communications and imaging archiving, recommends lossless encoders such as JPEG, Run Length Encoding (RLE), Lossless and near-lossless compression of continuous-tone still images (JPEG-LS) [3, 4] and JPEG2000 [5–7] for this purpose.

Although, these are the currently recommended encoders for medical imaging compression there are several other state-of-the-art lossless encoders. These encoders show higher compression efficiency rates than the ones proposed by DICOM. Some of these encoders were used in this work, namely: Context based Adaptive Lossless Image Codec (CALIC) [8,9], JPEG2000 Part 10 (JP3D) [10–12], Multi-scale Multidimensional Parser (MMP) [13–16], 3D-MMP [15,17], H.264 [18–20], High Efficiency Video Coding (HEVC) [21–23] and Minimum Rate Predictors (MRP) [24–28] encoder. MRP was the focus of this research, due to having one of the highest lossless compression efficiencies for still images [16]. In this work a comparison between these encoders applied to medical images is performed.

All DICOM recommended lossless encoders fail to exploit inter-slice redundancy in medical volumetric datasets. Considering this, and the fact that MRP has a high lossless compression efficiency for continuous-tone still images, in this work we propose to add inter-slice prediction support to this encoder.

The experimental results show that the MRP encoder achieves the highest compression efficiency performance for the used medical datasets for the considered encoders. These results show that this encoder is a good candidate for standardisation in the DICOM scope.

1.2 Objectives

The main topic of this research was to develop more efficient techniques to improve the reversible compression of medical images. As one of the most efficient state-of-the-art lossless encoders for the compression of continuous-tone images the MRP algorithm was chosen as a starting point to this work. Therefore, the following topics were developed:

- Study and comparison of state-of-the art lossless encoders present in the literature, when applied to medical image compression.
- Development of techniques to improve the compression efficiency of medical images by exploiting the characteristics of the images and of the lossless encoders, namely MRP.
- Extension and improvement of the MRP algorithm for the compression of medical images, by adding inter-slice prediction support to the encoder.

1.3 Dissertation structure

This dissertation is organized in five chapters and three appendixes. This chapter introduces this research work through its context, motivation and objectives.

Chapter 2 is dedicated to the state-of-the-art analysis relevant to this research work. Initially, medical imaging technologies, focusing in CT and MRI, and the DICOM standard are described. Then, a brief summary of the used lossless encoder algorithms is given, with a full description of MRP, and their compression efficiency is compared. The chapter is concluded with a review of some relevant state-of-the-art works.

In Chapter 3, processing techniques are proposed to enhance the compression efficiency of the MRP algorithm. This chapter focuses on techniques applied prior to the compression process in the encoder algorithms. Chapter 4, describes the proposed techniques added to the MRP algorithm, as well as the compression efficiency provided by those contributions. The chapter is concluded with a comparison of these proposed contributions. Chapter 5 draws some conclusions on the work presented in this dissertation and presents suggestions for future work in this field.

Appendix A, exhibits the medical images used in the experiments throughout this work. Appendix B, shows more detailed results of tables present in Section 3.5 of Chapter 3. Finally, in Appendix C a list of published and submitted papers, that resulted of this research work, is provided.

Chapter 2

Related State-of-the-Art

In this chapter the current state-of-the-art on medical images compression will be presented and discussed. We will start by giving a brief overview on medical imaging technologies, focusing on the used medical sequence types. Then we will proceed to describe the DICOM standard and the lossless encoders that are used in this research. Finally, recent developments in the field of lossless compression, both of continuous-tone and medical images, will be described.

2.1 Medical imaging

Medical imaging emerged with the increasing comprehension of several physical phenomena. Recent advances in understanding and use of these phenomena, such as X-ray, γ-ray, ultrasound waves and positron emission production, propagation and recording are strongly linked to the progress in different fields of medical imaging technologies. The availability of digital acquisition technologies and powerful computational resources are also driving new developments and useful solutions, e.g., tomography reconstruction both static and time-varying, etc.

There are several types of medical imaging modalities, for example Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Ultrasound (US) imaging, Positron Emission Tomography, Elastography, etc. Each of these modalities have their own characteristics [29,30], that depend on the associated physical processes and acquisition methods. Figure 2.1 shows four sagittal slices of a brain acquired with different technologies, where the distinguishable characteristics of various imaging modalities are clearly observable.

The characteristics of each medical imaging modality are also dependent on the domain of application. Thus, studies are necessary to assess where, when and how each type of image

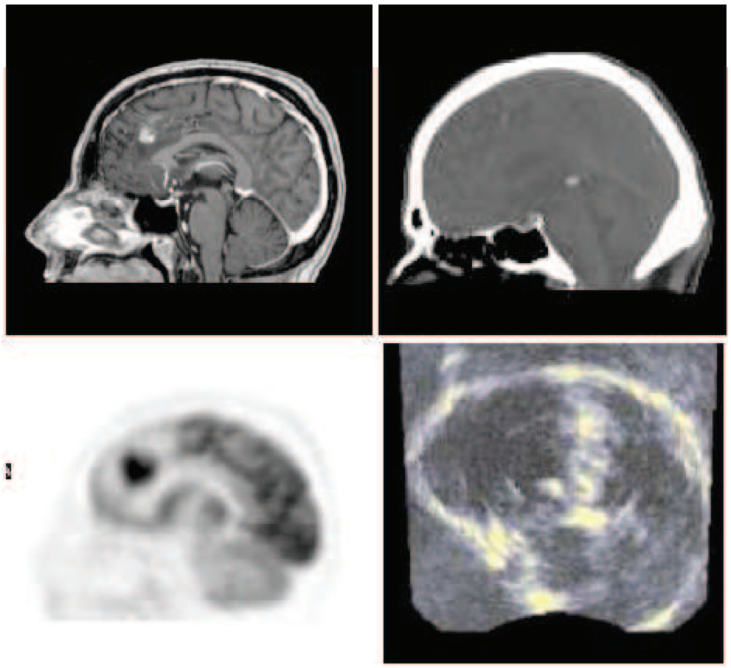


Figure 2.1: Sagittal slices of the brain by different imaging modalities. From top-left to down-right: Magnetic Resonance Imaging, Computed Tomography, Positron Emission Tomography and Ultrasound [29].

can be adequately used. For instance, X-rays are more adequate to image bone structures, while Magnetic Resonance provide higher definition and represents more accurately the softer tissues. In this work, we will focus on two of the most common volumetric medical imaging types, CT [30, 31] and MRI [30, 32] scans.

2.1.1 Computed Tomography

Computed Tomography (CT) has been used since the mid-1970's when it introduced a revolution in medical imaging [31]. Nowadays, millions of scans are performed around the world. A CT scanner consists of a ring housing a rotating X-ray source and a axially opposed corresponding detector, as depicted in Figure 2.2. The body is stationary inside the ring, and thousands of images are taken while the source-detector assembly rotates around the ring axis. If the body is moved longitudinally then a set of scans can be acquired to produce a CT volume.

CT scans can produce more detailed images than those of traditional X-rays, especially for the internal organs. The physical process behind computed tomography is the attenuation of X-rays in the body, where the amount of radiation absorption depends on the characteristics of the tissues in the path of the X-rays, so that different tissues will lead to different intensity readings by the detector [30].

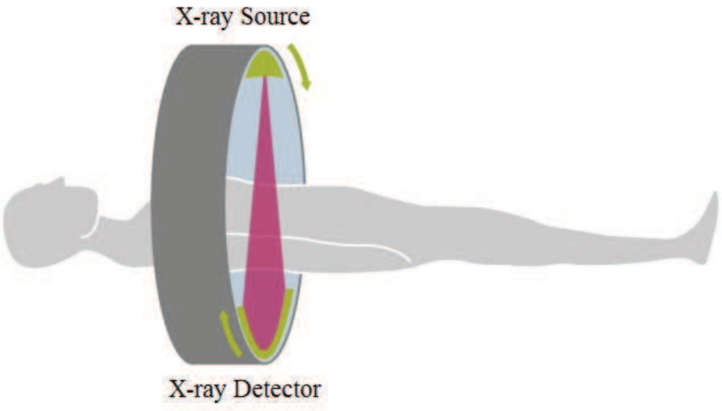


Figure 2.2: The principle of computed tomography with an X-ray source and detector unit rotating synchronously around the patient [31].

One of the advantages of CT is the speed of a complete scan, typically on the order of seconds or less. With the addition of a contrast agent further studies in blood vessels and organs can be made, but the preferred domain of application of CT is in bones imaging.

Due to its capture speed, CT imaging is very valuable in emergencies and have an important role in the diagnosing strokes, brain injuries, and others. Despite this, a CT scan exposes the body to more radiation than a conventional X-ray, and so CT are not recommended to some patients, such as pregnant women, or for repeated use.

2.1.2 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) uses magnetic fields and radio-frequency pulses, in order to obtain images of the internal structures from the human body. MRI scanners produce high contrast and very detailed images of the soft tissues and internal organs structure, for instance the brain, as can be seen in Figure 2.3 [32].

MRI scans are used in various examinations, such as brain exploration, detection of tumours or assessing the blood flow in different organs. MRI have the advantage of being able to reproduce high quality and high contrast images of the human body, without the need for contrast agents or the use of harmful radiations. The disadvantage of MRI is that patients with metallic implants or pacemakers might be put at risk by the magnetic fields that are involved in the imaging process.

An MRI system makes use of a magnet that creates a static magnetic field and which is "focused" on a specific body part. The signal is switched on and off, and the reflected radio-waves are processed to compute the absorption and reflection of these waves [30]. This information is used to render images with a direct correspondence to the physical

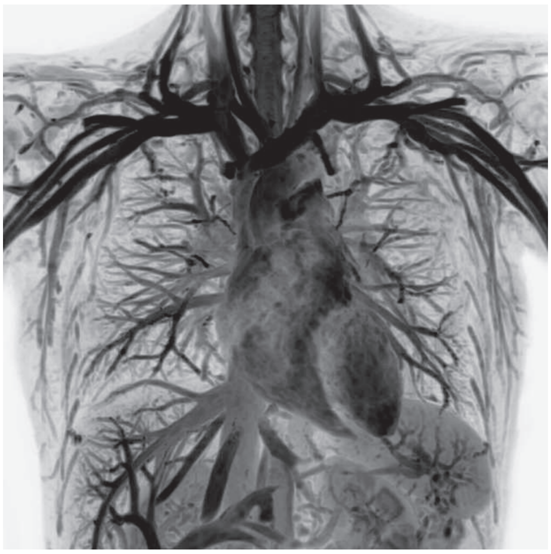


Figure 2.3: MRI of the thorax [32].

characteristics and location of the tissues.

MRI scanners can be used for a wide range of body parts including injuries of the joints, blood vessels, breasts, as well as abdominal and pelvic organs such as the liver or reproductive organs. Many diseases, such as brain tumours, can be visualized using this type of images because of the high contrast definition.

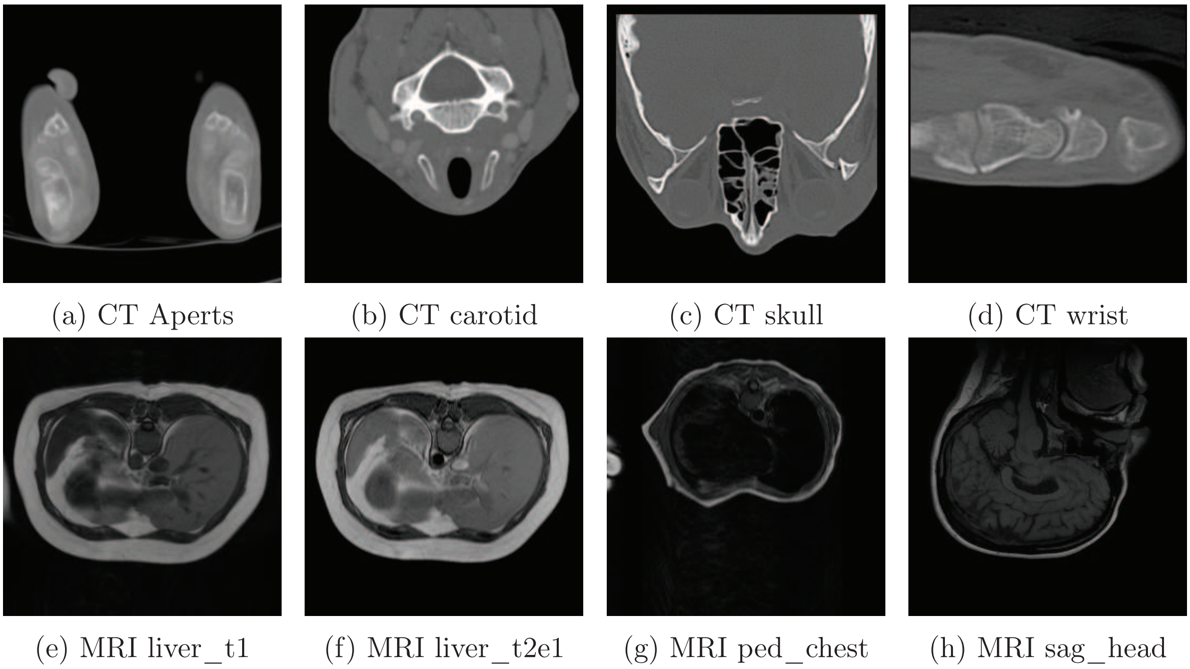
2.1.3 Image Dataset

The image dataset used to assess the performance of the encoders during this research is composed of eight volumetric medical sequences: four CT and four MRI scans, all available in [33]. These images are available on the repository of the Center for Image Processing Research (CIPR) of the Rensselaer Polytechnic Institute. The spatial resolution, bit depth and number of slices of each volume are shown in Table 2.1.

Table 2.1: Description of the used medical images datasets.

Datasets	Type	Slices	Resolution	Bits-per-pixel
Aperts	CT	97	256 × 256	8
carotid		74	256 × 256	8
skull		203	256 × 256	8
wrist		183	256 × 256	8
liver_t1	MRI	58	256 × 256	8
liver_t2e1		58	256 × 256	8
ped_chest		77	256 × 256	8
sag_head		58	256 × 256	8

These data volumes are sets of spatially adjacent slices that require a large number of bits to be represented, due to their resolution and number of slices, as well as bit depth. Figure 2.4 shows only the midpoint slice of each volume of the dataset, which are completely shown in Appendix A.



(a) CT Aperts (b) CT carotid (c) CT skull (d) CT wrist

(e) MRI liver_t1 (f) MRI liver_t2e1 (g) MRI ped_chest (h) MRI sag_head

Figure 2.4: Middle slice of each of the used medical images [33].

2.2 Digital imaging and communications in medicine

Digital Imaging and Communications in Medicine (DICOM) [5–7] is an international standard for compression, communications and related information of medical images, first published in 1993. This standard is present in most of the medical imaging devices, such as CT, MRI, etc., being the most widespread healthcare standard in the world. The use of standards is particularly essential in medical imaging [30], as it assures that images can be interchangeably used and shared between the various institutions, hospitals, imaging centres, etc.

The DICOM standard is mainly a protocol for image exchange. In the context of this particular work, we are interested in its specifications for image compression. In the standard, an image data is defined as a simple Two Dimensional (2D) representation of values in a series or dataset. With the growing needs of US, CT and MRI imaging, a multi-slice concept was designed [30].

DICOM integrates common image compression standards, both reversible (lossless) and irreversible (lossy), from International Organization for Standardization (ISO) and International Telecommunication Union (ITU). The standard defines an encapsulated format archive, where the compression information is included in the bit-stream syntax [34]. Despite allowing a number of coding algorithms to be used, the standard makes no assumptions or recommendations on which encoders should be applied and in which applications [6].

This is especially valid for the irreversible encoders, as there is still an open debate on whether lossy compression should be used in the context of medical imaging, specially when images are used for diagnose purposes. Regulatory bodies in the UK, EU, USA, Canada and Australia allow the use of lossy compression for medical images. However, the decision of using irreversible compression is left for the institutions and radiologists [35]. Despite this, reversible compression is recommended by several regulatory bodies, such as the Royal Australian and New Zealand College of Radiologists, where possible [36].

The DICOM standard supported encoders are the following:

- JPEG, for lossy and lossless compression;
- RLE, for lossless compression;
- JPEG-LS, for near-lossless and lossless compression;
- JPEG2000, for lossy and lossless compression;
- MPEG-2, for image compression using the main profile and the main or high levels, for lossy compression;
- H.264, for video compression using the main and the stereo main profiles, levels 4.1 or 4.2, for lossy compression.

As can be seen, there are no video lossless encoders contemplated in the DICOM standard. In this research work, the lossless encoders allowed in the standard and state-of-the-art lossless encoders were also studied and compared. This work focus on the compression of medical sequences, such as CT and MRI, therefore the inter-slice prediction and lossless video compression was also addressed.

2.3 Lossless coding algorithms

In this section current DICOM and state-of-art lossless encoders are described. It is expected that the compression algorithms will be better understood and, therefore, allow us to better exploit their characteristics. This work is mainly focused on the Minimum Rate Predictors (MRP) encoding algorithm, thus, it is fully described.

2.3.1 A context based adaptive lossless image codec

Context based Adaptive Lossless Image Codec (CALIC) [9] is a lossless image encoder for continuous-tone images. CALIC started as a candidate algorithm to ISO/JPEG standardisation, aimed to lossless encoding of continuous-tone images [8]. Eventually, CALIC was passed over, and the LOw COmplexity LOssless COmpression for Images (LOCO-I)[1] algorithm was chosen, despite being more efficient, but presenting higher computational complexity.

The encoding and decoding process works in a raster scan order, requiring only a single pass. The prediction requires only values from the two previous lines of the image. Figure 2.5 shows a block diagram for the CALIC algorithm.

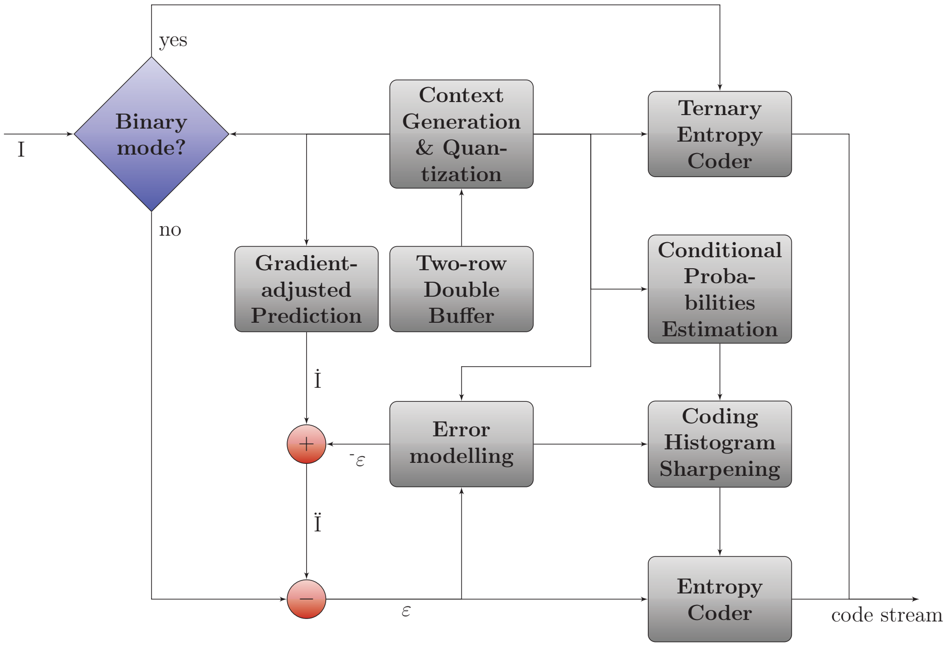

yes
I
Binary mode?
no
Context Generation & Quantization
Ternary Entropy Coder
Gradient-adjusted Prediction
Two-row Double Buffer
Conditional Probabilities Estimation
İ
+
-ε
Error modelling
Coding Histogram Sharpening
Ï
−
ε
Entropy Coder
code stream

Figure 2.5: Schematic description of the CALIC coding system [8].

CALIC uses a two step approach, prediction followed by context modelling for the residual coding. This encoder utilizes a context-based predictor in order to efficiently model the image data and characteristics. CALIC has two distinct operation modes, binary and continuous tone, the choice between these two modes is automatically made in the compression process. The binary mode is used when a given area of the image has just two distinct intensity values, and the symbols are encoded with a ternary entropy coder. In the continuous mode, a set of prediction, context modelling and entropy coding is used.

For the prediction step, a simple gradient-based non-linear prediction scheme, Gradient-Adjusted Predictor (GAP), is used. The GAP prediction adjusts its coefficients based on

[1] This algorithm and the resulting standard led to the JPEG-LS encoder. This encoder will be described in the next section.

local gradients estimation. This predictor is context sensitive and adaptable by modelling of prediction errors and feedback from the expected error, conditioned by properly chosen modelling contexts, as can be seen in Figure 2.5. The performance of the GAP predictor is improved via context modelling. Figure 2.6 shows the possible reference pixels, in black, relative to the current pixel x, in red. The local gradients are determined as in Equation 2.1, and Table 2.2 shows the prediction result, according to the gradients. The prediction errors of the continuous mode are then encoded using an adaptive m-ary arithmetic encoder, CACM++ [37].

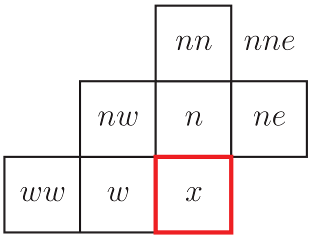


Figure 2.6: Reference pixels positions.

$$\begin{aligned} d_h &= |w - ww| + |n - nw| + |ne - n| \\ d_v &= |w - nw| + |n - nn| + |ne - nne| \end{aligned} \tag{2.1}$$

Table 2.2: Prediction value for the pixel given by GAP.

Edge type	Horizontal	Vertical
Sharp	w	n
Normal	$\frac{aux + w}{2}$	$\frac{aux + n}{2}$
Weak	$\frac{3 \times aux + w}{4}$	$\frac{3 \times aux + n}{4}$
$aux = \frac{w + n}{2} + \frac{ne - nw}{4}$		

2.3.2 JPEG-LS

Lossless and near-lossless compression of continuous-tone still images (JPEG-LS) [3,4], not to be confused with the lossless version of the JPEG encoder, is a JPEG, ISO/ITU standard for lossless and near-lossless compression of continuous-tone images. This encoder is based on the LOw COmplexity LOssless COmpression for Images (LOCO-I) algorithm. This algorithm is divided into three main stages, prediction, context modelling and residue coding, as represented in the block diagram of Figure 2.7.

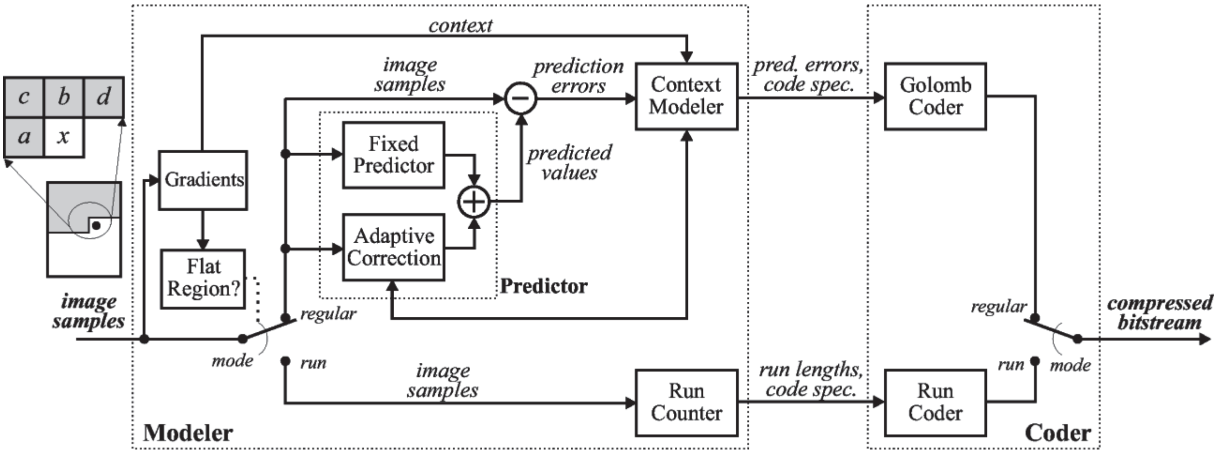


Figure 2.7: JPEG-LS block diagram [4].

The JPEG-LS prediction uses the template depicted on the left side of Figure 2.7. In this template, a, b, c and d are neighbouring samples of the current sample x. This template is on the causal area of the image and, by using only four past samples, JPEG-LS limits the image buffering requirements.

A fixed prediction scheme given by Equation 2.2 is used in the encoder. This predictor tends to pick pixel b when a vertical edge is present, pixel a when a horizontal edge is detected and $a+b-c$ if no edge is present. This predictor is called Median Edge Detector (MED).

$$\hat{x} \triangleq \begin{cases} \min(a,b), & \text{if } c \geq \max(a,b) \\ \max(a,b), & \text{if } c \leq \min(a,b) \\ a+b-c, & \text{otherwise.} \end{cases} \tag{2.2}$$

After the prediction stage, a context modelling is applied to the prediction error. In LOCO-I a Two-Sided Geometric Distribution (TSGD) model is used for the modelling, as the example showed in Figure 2.8.

The context that shapes the current prediction residual encoding is determined from the local gradient surrounding the current sample. Therefore, the level of activity can be determined, which in turn allows for the determination of the statistical behaviour of the prediction errors.

Finally, to encode the prediction residual errors, JPEG-LS uses a minimal complexity subfamily of the optimal prefix codes for TSGDs. These optimal codes are based on Golomb codes [38], which allow the calculation of the code word for any given sample without the use of code tables. This encoding method is adaptive; when a new sample is encoded the contexts and the probabilities are updated in order to further optimize the

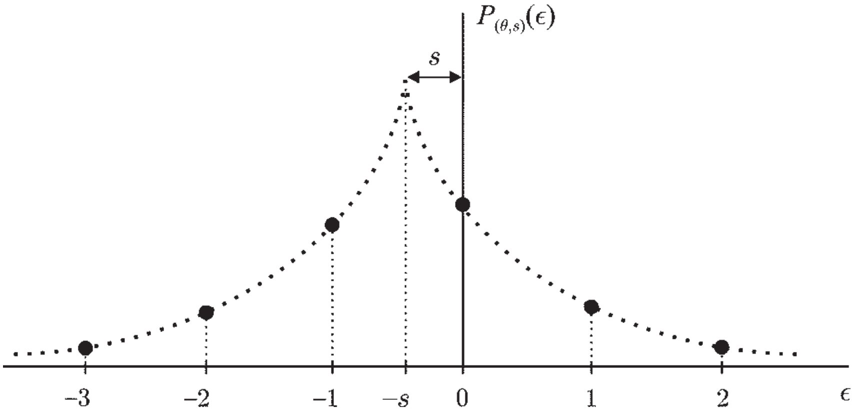


Figure 2.8: Two-sided geometric distribution (TSGD) [4].

encoding process.

2.3.3 JPEG2000

JPEG2000 [39, 40] is a standard for image compression, maintained by ISO/IEC 15444-1 and ITU recommendation T.812. JPEG2000 is a wavelet transform based encoder with applications from natural images to medical imaging and others. This standard has essentially been established to be a more efficient encoder and substitute to the JPEG encoder. The general structure of the JPEG2000 codec is shown in Figure 2.9.

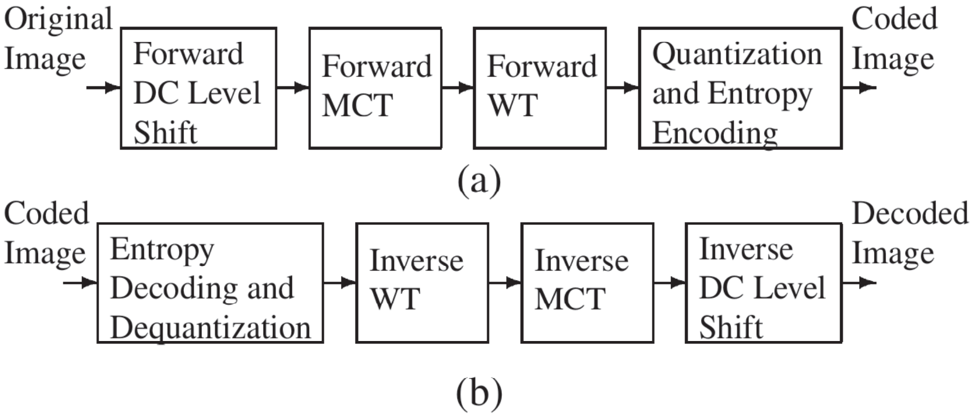


Figure 2.9: General structure of JPEG-2000 [40]. The (a) encoder and (b) decoder.

JPEG2000 applies a wavelet transform to an image which is represented by several sub-bands of frequency, as show in Figure 2.10[2]. It can be observed in this figure that the sub-bands are sampled at different spatial resolution, thus allowing the spatial scalability in JPEG2000. This characteristic is used in the decoder to build sequentially better quality versions of the encoded image, as more frequency bands are decoded.

[2] Obtained using the Wavelet Toolbox of Matlab R2014a.

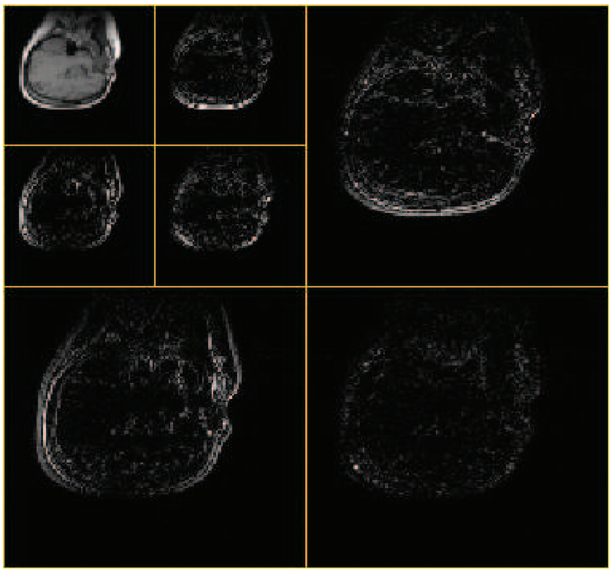


Figure 2.10: Wavelet transform decomposition for the *skull* sequence.

This standard employs different techniques and different wavelet transforms in order to encompass both lossy and lossless compression. In the case of the lossless encoding mode, an integer reversible wavelet transform is used, thus bypassing the need for quantization, unlike what is shown in Figure 2.9.

The first step in the encoding process is to adjust each image sample by an additive bias, or DC Level Shifting. This value is chosen in order to make all the sample values to be within a dynamic range centred around zero. The wavelet coefficients are encoded with an entropy coder.

Motion JPEG2000

Motion JPEG2000 [41] is a ISO/ITU standard, and part of the JPEG2000 recommendation. It was designed for video coding, although it uses only intra prediction, with every frame being independently coded by a variant of JPEG2000 encoder. Some of the expected applications are: storing of video clips, high-quality video editing, medical imaging compression, etc.

JP3D

JP3D [10–12] is an extension of the JPEG2000 for compression of volumetric images, such as medical images. This extension is backwards compatible with the JPEG2000 Part 1 and Part 2, and allows the use of image tilling. This tilling in JP3D results in Three Dimensional (3D) blocks, rather than 2D blocks, which are coded independently. The resulting tiles are encoded using a 3D Discrete Wavelet Transform (DWT) and a 3D

Embedded Block Coder with Optimized Truncation (EBCOT) [42] mechanism.

The 3D-DWT divides the image into sub-band 3D blocks where, the decomposition levels can be chosen independently in the three dimensions. The encoder partitions the wavelet coefficients into dyadically-sized cubes, for each sub-band, called code-blocks which are then individually coded with 3D-EBCOT, resulting in a partition like the one in Figure 2.11.

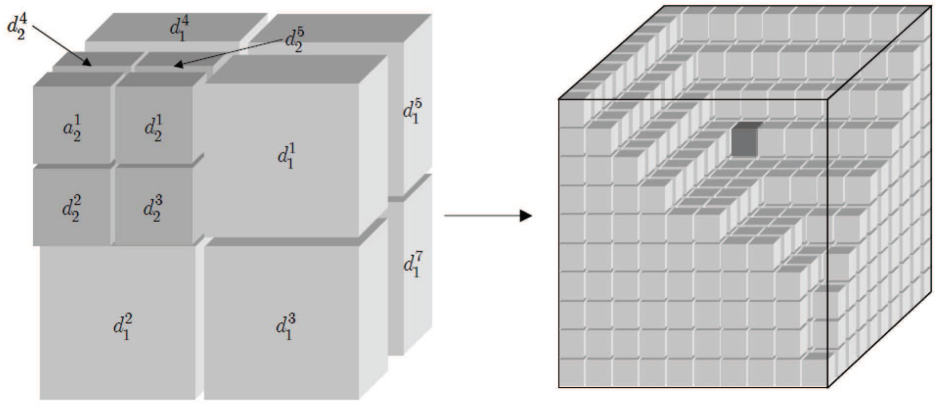


Figure 2.11: Code-block partitioning of a wavelet volume [11].

Like JPEG2000, JP3D, is scalable both in resolution, quality and spatially. Also, like in JPEG2000, the used wavelet is a 3D integer reversible wavelet.

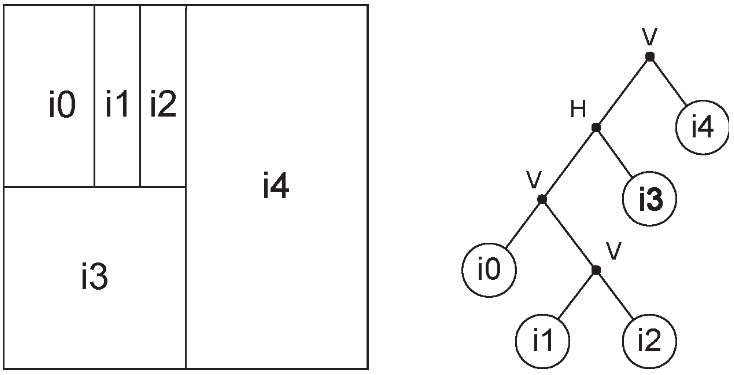
2.3.4 Multi-scale multidimensional parser

Multi-scale Multidimensional Parser (MMP) [13–15] is a dictionary based pattern matching compression algorithm. This algorithm was first derived from a Lempel-Ziv lossless scheme, although none of the most recent implementations of MMP has been adapted to lossless coding. Thus, a lossless version of this encoder has been proposed in [16].

The MMP algorithm performs a flexible block segmentation in the image to encode, with non-overlapping blocks, usually of 16 × 16 pixels. Each block can be further divided using a flexible segmentation, and are encoded in a raster scan order, as can be seen in Figure 2.12. In each block, MMP applies intra prediction, based on the H.264 modes [20].

MMP, however, does not use the traditional transform-quantize-encode paradigm, using instead a dictionary search for its residue encoding. A block-matching is performed between the residue blocks and the dictionary elements. This encoder can also use scale transformations, allowing to match different size blocks. The dictionary is updated with recent encoded residue blocks, to optimise the encoding efficiency.

In the lossless implementation, that we will refer only as MMP for simplicity, an im-



(a) (b)

Figure 2.12: Flexible block segmentation and partition tree [14]: (a) segmentation of an image block, (b) corresponding binary segmentation tree.

plicit intra prediction mode was added, based on the Least Squares Prediction (LSP) algorithm [43]. Also, the horizontal and vertical modes of H.264 were improved, based on [44], by expanding the neighbourhood to use extra pixels, instead of just the ones on the block edge. Finally, a Differential Pulse Code Modulation (DPCM) technique is also applied for the encoding of the residue for all the prediction modes.

2.3.5 3D-MMP

In [15,17], the MMP encoder was extended into a volumetric compression algorithm, called 3D-MMP. In this implementation the sequence information is treated as a volumetric signal, instead of the usual slice-by-slice approach. The flexible partition used in MMP was extended for 3D blocks, as can be seen in Figure 2.13. Thus, each block can be segmented into three directions: temporal, horizontal and vertically.

3D-MMP also uses a dictionary based approach but 3D blocks pose new challenges, driving to a new dictionary design that uses multiple scaled versions of the dictionary. Therefore, when performing a search, the algorithm only needs to perform it in the corresponding scaled dictionary. This method requires more memory, but the computational complexity is lower than having a single dictionary.

As explained before, MMP uses the intra prediction modes of H.264. In 3D-MMP these modes are expanded to a 3D block basis. Thus, as the neighbour pixels are on the edge of the blocks, the neighbourhood will also be three-dimensional, as can be seen in Figure 2.14. Additionally, the LSP algorithm, based on [45], was also adopted in 3D-MMP and expanded to have a 3D support.

For the compression of video sequences, 3D-MMP, can rearrange the slices by grouping them on the temporal axes, separating the odd and even slices, in a similar way of I-type

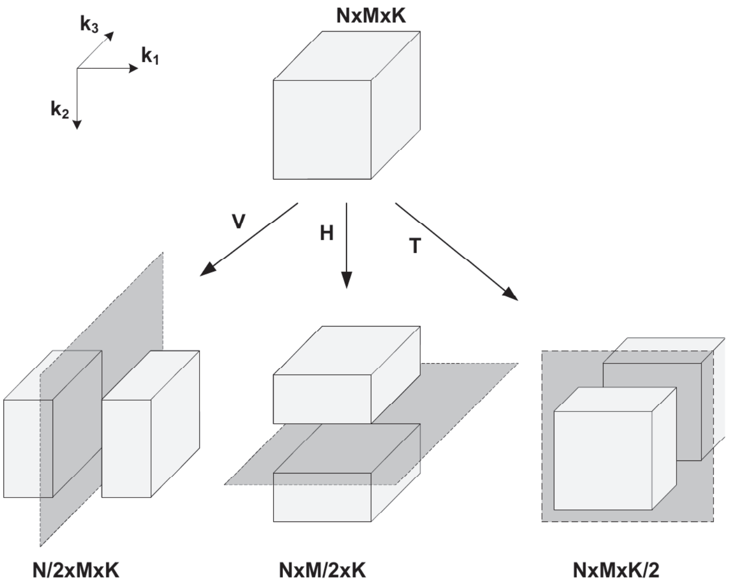


Figure 2.13: Triadic flexible partition [17].

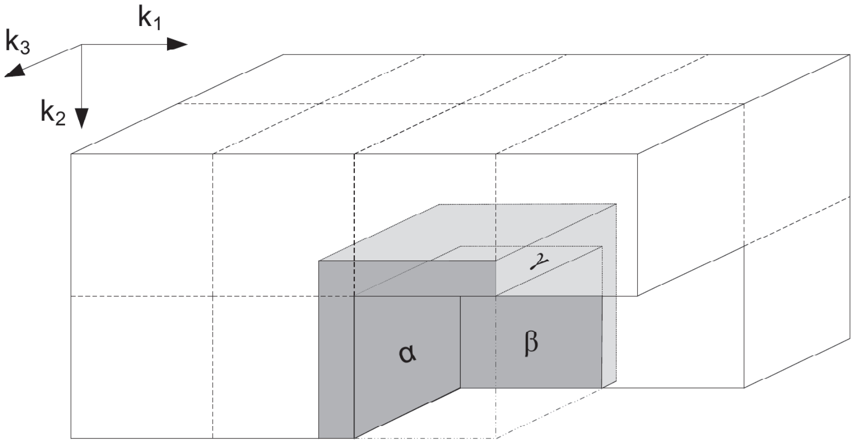


Figure 2.14: Block neighbourhood [17].

and B-type slices, as in H.264, as shown in Figure 2.15. This feature adds support for bidirectional prediction in 3D-MMP. For instance, the LSP is now able to use reference pixels in future slices. The encoding process may also be sequential without the rearrangement of slices.

This encoder also has a support for lossless compression, using the λ parameter that defines the weight given to the rate-distortion optimisation, that must be set to zero.

2.3.6 H.264 / Advanced Video Coding

The H.264 encoder is a video compression standard [18–20], from ITU and ISO. This is a hybrid encoder, whose algorithm has four main stages: prediction, both intra and inter

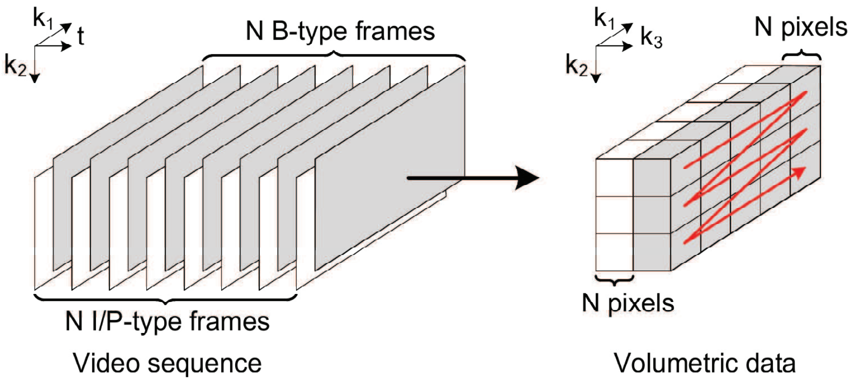


Figure 2.15: Hierarchical video compression architecture [17].

slice[3], transform, quantization and entropy coding. The structure of the H.264 encoder can be seen in Figure 2.16.

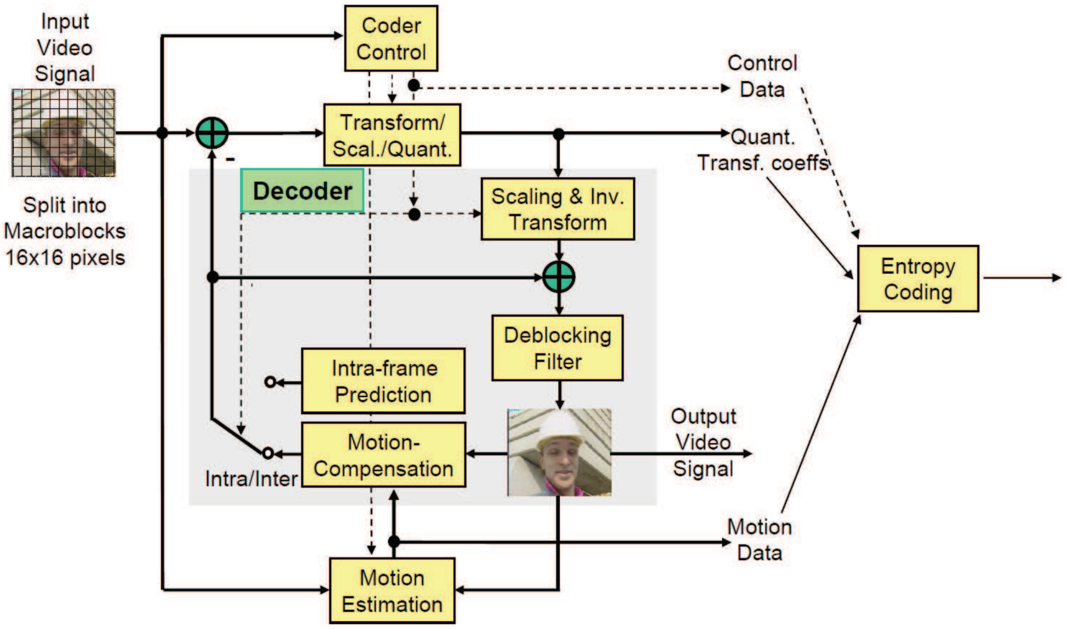


Figure 2.16: Basic coding structure of H.264 [19].

For the prediction the image is divided into macro-blocks of 16 × 16 pixels. There are three types of slice categories in H.264: I slices, whose macro-blocks are predicted only with intra prediction modes, P slices, additionally to the intra prediction modes the macro-blocks of these slices can also use inter prediction with one motion-compensated reference, and B slices, that additionally to the P slices prediction can use two inter-prediction references.

For intra prediction H.264 can divide the macro-blocks into blocks of 4 × 4 pixels. Thus, H.264 allows four intra prediction modes, namely *Vertical*, *Horizontal*, *DC* and *Plane*. The 4 × 4 block may be encoded with nine prediction modes, as can be seen in Figure 2.17.

[3] In this context, the mention to 'slice' is interchangeable to the use of 'frame'. As the subject of this work is medical imaging compression only the slice expression is used.

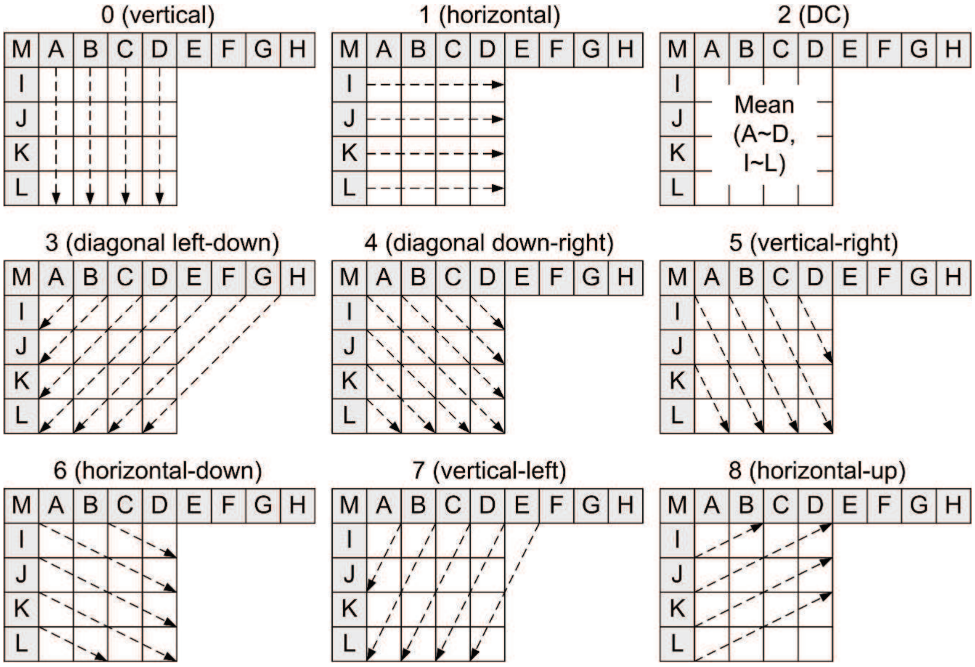


Figure 2.17: H.264 intra prediction modes for 4 × 4 blocks [46].

In the inter-slice prediction of H.264, the intra prediction modes can also be used. However, in P and B-type slices motion compensation can also be used, with past or past and future references, respectively. Macro-blocks in the motion compensation can be divided into blocks down to 4 × 4 pixels. For the prediction a precision of up to quarter-pixel can be used.

After the prediction, the Hadamard or Discrete Cosine Transform (DCT) transform is applied to the resulting residual. The transform coefficients are quantised and entropy coded. However, when quantisation is applied the coding process becomes lossy, meaning that part of the information is permanently lost.

Additionally, H.264 allows for lossless compression. In this mode, the transform and quantisation processes are bypassed. The lossless mode of H.264 reference software can only use intra encoding, which is the one used in this work.

2.3.7 High Efficiency Video Coding

The High Efficiency Video Coding (HEVC), also known as H.265, is the most recent state-of-the-art video codec standard [21, 22]. This standard was proposed in order to replace H.264, having as a main goal to improve the compression performance relative to H.264, in the order of 50% bit-rate savings for the same quality.

The H.265 standard shares many of the characteristics of H.264, the main characteristic being the hybrid coding structure. In Figure 2.18, the basic encoder structure of HEVC can be seen.

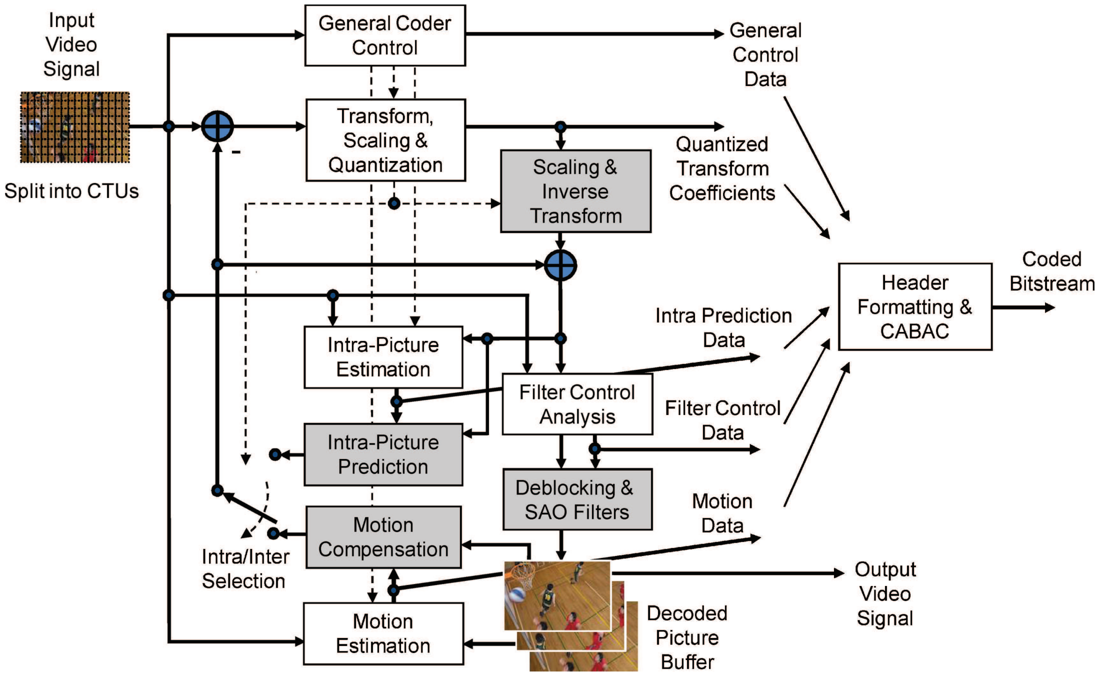


Figure 2.18: Basic coding structure of HEVC [22].

One of the differences between HEVC and H.264 is the replacement of the macro-block by the Coding Tree Unit (CTU), which can have up to 64 × 64 pixels. The CTU can then be partitioned into smaller blocks using a quadtree-like signalling. The usage of higher size blocks allows for a higher compression efficiency, specially in higher resolution images, and for the use of more prediction modes in the intra prediction. Therefore, HEVC uses 35 intra prediction modes, that can be seen in Figure 2.19. The right side of this figure shows an example of a directional prediction mode of the encoder.

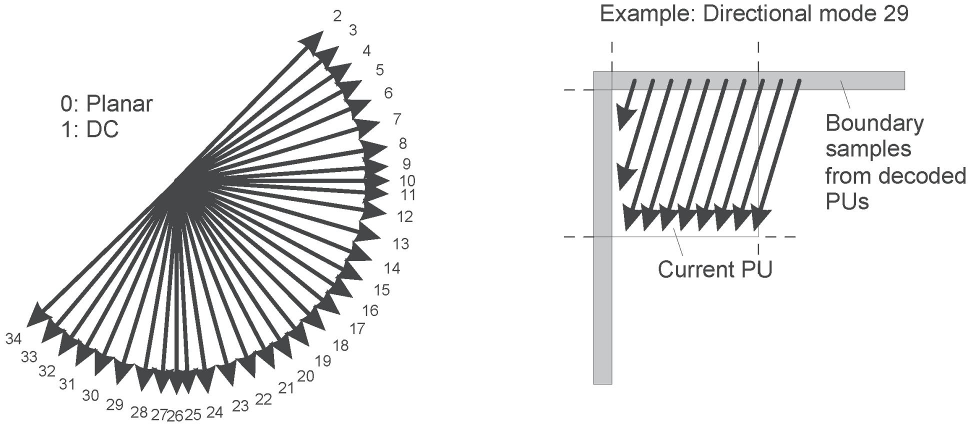


Figure 2.19: HEVC intra prediction modes [22].

As in H.264, H.265 standard relies on motion compensation for the P and B type slices. A quarter-pixel precision is, once again, used to perform the motion estimation.

The HEVC reference software in the lossless mode allows the use of inter prediction, unlike H.264, in such profiles as Random Access or Low Delay. In this mode, the transform and quantisation steps are bypassed, with the residue being encoded with an entropy coder.

Range extension

Like H.264, HEVC has several extensions to its algorithm. For the lossless coding, the Range Extension is of especial interest, HEVC Range Extension (HEVC RExt) [23]. This extension introduced the support for higher bit depth images and different chroma subsamplings, such as 4:2:2 and 4:4:4. The objective is to perform lossless and screen content compression.

The main focus of this research is the lossless coding of medical images, then, the new HEVC RExt tools for lossless compression are of interest. Some of the changes that benefit the lossless coding are: **Intra-picture block copying prediction**, similar to the motion compensation but for already coded blocks of the same sequence, **Smoothing disabling of intra-picture prediction**, disables a smoothing filter used in intra prediction, **Transform skip mode modifications**, allows the use of a DPCM vertical and horizontal modes for the the residual signals, when the transform is bypassed. Considering this, we will compare HEVC with its Range Extension for the lossless compression modes.

2.3.8 Minimum rate predictors

The Minimum Rate Predictors (MRP) codec was first proposed in 2000 [24, 25], but has been improved in [26–28]. MRP uses multiple linear predictors adapted to each image, on a Variable Block Size (VBS) basis.

Initially, the image is divided into blocks of 8×8 pixels. These blocks are then sorted in to one of M classes, each class being represented by a different linear predictor, according to the block variance. This information is used as a training set for the design of the linear predictors. All pixels of the same class are then used in Yule-Walker equations. These equations, when solved, return the optimum prediction coefficients, which are then used to calculate the prediction.

For a given pixel, p_0, the prediction value given by the *m-th* class, $m = 1, ..., M$, is given by Equation 2.3. The linear predictors are used in order to better estimate the image structures. Each predictor uses K reference pixels, distributed, as can be seen in

Figure 2.20, for $K = 30$.

$$\hat{s}(p_0) = \sum_{k=1}^{K} a_m(k) \cdot s(p_k), \tag{2.3}$$

In this equation, $a_m(k)$ is the prediction coefficient for a given pixel, of the *m-th* class, and $s(p_k)$ is the value of the p_k reference pixel.

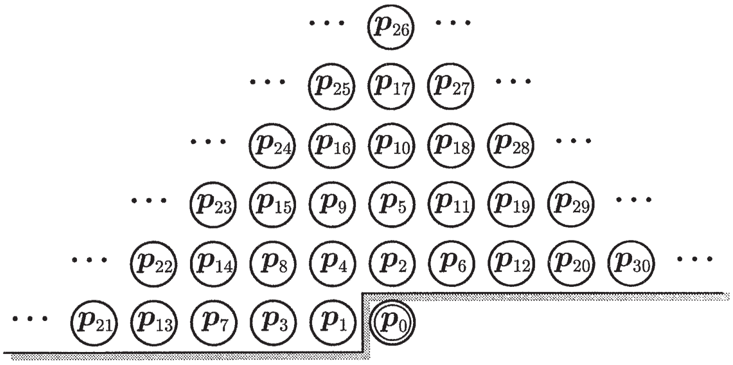

p_{26}
p_{25} p_{17} p_{27}
p_{24} p_{16} p_{10} p_{18} p_{28}
p_{23} p_{15} p_{9} p_{5} p_{11} p_{19} p_{29}
p_{22} p_{14} p_{8} p_{4} p_{2} p_{6} p_{12} p_{20} p_{30}
p_{21} p_{13} p_{7} p_{3} p_{1} p_{0}

Figure 2.20: Disposition of reference pixels [28].

After the coefficients and prediction determination, the cost of the prediction error can be calculated, taken as model the generalised Gaussian functions. The context selection for the modelling is given by Equation 2.4:

$$U = \sum_{k=1}^{12} \frac{1}{\delta_k} \cdot \left| s(p_k) - \hat{s}(p_k) \right|, \tag{2.4}$$

where δ_k is a weighting factor, indicating the Euclidean distance between p_0 and reference pixel p_k, given by Equation 2.5 and $s(p_0) - \hat{s}(p_0)$ is the prediction error.

$$\delta_k = \frac{64}{\sqrt{d_x(k)^2 + d_y(k)^2}} \tag{2.5}$$

In Equation 2.5, 64 represents the precision of the weighting factor, 6 bits, $d_x(k)$ and $d_y(k)$ represent the distance from the current pixel to the k reference pixel in the X and Y axis, respectively. Finally, $\sqrt{d_x(k)^2 + d_y(k)^2}$ is the euclidean distance between the k reference pixel and the pixel for which we wish to determine the context.

The U parameter is closely related to the variance of the prediction error. Thus, the parameter U is an estimation of the variance and it is quantised and calculated for each block. The thresholds for the quantisation are optimised in order to achieve the highest coding efficiency for each class. The blocks are then re-classified, regarding the U parameter and the context quantisation thresholds. With this new classification, the prediction coefficients will be determined again, and the process is repeated.

These optimisations are performed in order to minimise the cost function for the encoding image, the cost function is given by Equation 2.6:

$$J = \sum_{p_0} L\left(e|\hat{s}(p_0), n\right) + B_a + B_m + B_t, \tag{2.6}$$

where $L(e|\hat{s}(p_0), n)$ is the total code length of the prediction errors and B_a, B_m and B_t are the code lengths of the prediction coefficients, class selection and context modelling threshold values, respectively.

When this process is concluded, the MRP algorithm has another optimisation loop, using VBS. The block size can now change from blocks of 32×32 to 2×2 pixels, in a quadtree segmentation structure, that is optimised in order to minimise the cost function. The size of the used blocks highly depends on the local characteristics of the image.

The classes are recursively chosen for each block, regarding to the cost function J. The thresholds, the block classification and the optimum value of the shape parameter, for the probability distribution, are then optimised in a loop. If the optimisation flag is activated, two prediction coefficients are randomly chosen and a partial optimisation is performed by slightly varying their values. These optimisations are repeated several times for each class. The arithmetic encoding of all the needed parameters is performed by a range coder [47], or by a simple Huffman coder.

One of the main characteristics of MRP is the fact that it minimises a cost function, representing the amount of prediction error data, instead of minimising a sum of least-squares of the prediction errors, as done by other encoders. The prediction order, K, and the number of classes, M, can be determined from the dimensions of the encoding image. Thus, selecting appropriate values for these parameters .

2.4 Lossess codec comparison

In this section, the previously described lossless encoders will be compared, when applied to the compression of medical images. To evaluate these encoders, the test images were encoded with the parameters set to lossless compression. Publicly available software implementations of these encoders were used for the tests.

The CALIC and JPEG-LS encoders were used with their default configuration values for lossless compression. For the MRP algorithm, the extra optimisation flag was activated, with the remaining options set by the encoder.

In the MMP encoder, the fast implementation was used, and the default dictionary size was used. In 3D-MMP the λ parameter was also set to zero, the block size was of $16 \times 16 \times 4$ pixels, in the $Y \times X \times Z$ format, and the prediction level was set to 8, indicating the minimum partitioning size to use prediction. In 3D-MMP the hierarchical video compression was used, and the dictionary was set to use a maximum of 5000 elements.

For JPEG2000, the software implementation OpenJPEG v2.1 is used with the default configuration. This software also includes an implementation of JP3D, which was configured to use a 3D-DWT and 3D-EBCOT.

For H.264/AVC, the latest release of the reference software JM 18.6 was used. For lossless coding, FRExt High 4:4:4 Profile was used for Intra coding, with *QP* and *QP Offsets* set to 0.

As for HEVC, the reference software HM 16.4 was used, both for Intra Main and Random Access profiles. In the lossless coding mode, QP was set to 0, and both *TransquantBypassEnableFlag* and *CUTransquantBypassFlagForce* were set to 1. For the remaining configuration parameters the default values were used. HEVC RExt is included in the same reference software as HEVC. The same parameters, as in HEVC, were used with the same values. The HEVC RExt specific parameter *CostMode* was set to lossless.

For the lossless coding modes, only the JP3D, 3D-MMP, H.264 and HEVC encoders can have a sequence as the input. All other encoders only use intra prediction, for these each slice in a sequence is encoded independently and the result is the average of all the encoded frames. Only, JP3D, 3D-MMP and HEVC Random Access (HEVC RA) can exploit the inter-slice redundancy.

Lossless compression was chosen for this work, as it is often a requirement for the compression of medical images, as stated in Section 2.2. For instance, when reversibly compressed images are used for the diagnosis, possible compression artefacts are not an issue. The results for these encoders, in bits-per-pixel (bpp), are shown in Table 2.3, for the image encoders, in Table 2.4, for the video/volumetric encoders, and in Table 2.5 for 3D-MMP.

The 3D-MMP encoder requires that the number of slices of the input sequence to be multiple of the block size in the Z direction. Therefore, the results of this encoder are not directly comparable with the remaining encoders. In Table 2.5 the number of slices used in 3D-MMP are also shown. In Chapter 3, we show a table of encoding results for these encoders in which the number of slices is multiple of 16, see Table 3.2. Therefore we can compare the sequence *Aperts* as it uses the same number of slices in both tables, for 3D-MMP. It can be observed that, for that particular sequence, the compression efficiency

Table 2.3: Performance comparison of the image encoders (results in bpp).

Sequence	H.264	HEVC Intra	RExt Intra	MMP	JPEG 2000	JPEG-LS	CALIC	MRP
Aperts	1.193	1.289	1.136	1.178	1.261	1.058	0.998	0.775
carotid	2.062	2.198	2.001	1.977	2.019	1.778	1.684	1.374
skull	3.183	3.083	2.890	2.959	2.991	2.761	2.628	2.329
wrist	1.911	2.195	1.890	1.717	1.757	1.627	1.550	1.173
liver_t1	3.489	3.742	3.400	3.393	3.256	3.160	3.022	2.582
liver_t2e1	2.806	2.811	2.561	2.460	2.572	2.418	2.269	1.722
ped_chest	3.080	3.352	3.051	3.074	3.021	2.937	2.789	2.337
sag_head	2.635	2.732	2.594	2.808	2.905	2.582	2.519	2.279
Average	2.545	2.675	2.440	2.446	2.473	2.290	2.183	1.821

Table 2.4: Performance comparison of the video encoders (results in bpp).

Sequence	HEVC R.A.	RExt R.A.	JP3D
Aperts	0.826	0.728	0.941
carotid	1.587	1.425	1.547
skull	1.905	1.766	2.088
wrist	1.155	1.002	1.238
liver_t1	2.392	2.052	2.356
liver_t2e1	1.726	1.510	1.745
ped_chest	1.699	1.534	2.071
sag_head	1.873	1.748	2.160
Average	1.645	1.471	1.768

of 3D-MMP is similar to that of JP3D. Finally, the tests were not performed to all the sequences due to the computational complexity of the encoder. For the *Aperts* sequence, for instance, the encoding process took 4128742 seconds, which is equivalent to almost 48 days. Hence, and also due to its compression efficiency, 3D-MMP will not be used in the remainder of this work.

The remaining two tables show the results for the image and video encoders. In these tables R.A. means the Random Access profile, and RExt means the HEVC RExt encoder. As expected, it is possible to see that encoders which are able to exploit the inter-slice redundancies in a sequence, generally present better results. The best overall results, on average, are obtained for HEVC RExt with the Random Access profile. This was expected as HEVC is the state-of-the-art video encoder, and its Range Extension has several tools that improve the lossless compression of sequences. It can also be observed from Tables 2.3 and 2.4 that the Range Extension of HEVC improved the lossless compression efficiency of this encoder, with an improvement of roughly 0.2 bpp.

For the image encoders, the best result is obtained for the MRP codec. This can be

Table 2.5: Compression results of the 3D-MMP (results in bpp).

Sequence	Slices	3D-MMP
Aperts	96	0.938
carotid	72	1.622
skull		
wrist		
liver_t1	56	2.579
liver_t2e1	56	2.051
ped_chest	76	1.804
sag_head	56	2.303

explained by the encoder prediction efficiency, as the linear predictors are suitable to describe several image structures and they are optimised for the input image. MRP has results that are close to that of the video encoders, at the level of JP3D, with a difference of 0.05 bpp, and without inter-slice prediction support. For the HEVC encoders this deviation is higher, with 0.18 bpp for HEVC RA and 0.35 bpp for HEVC RExt R.A. The encoding efficiency shown by MRP was one of the reasons that this encoder was chosen as main object of our research.

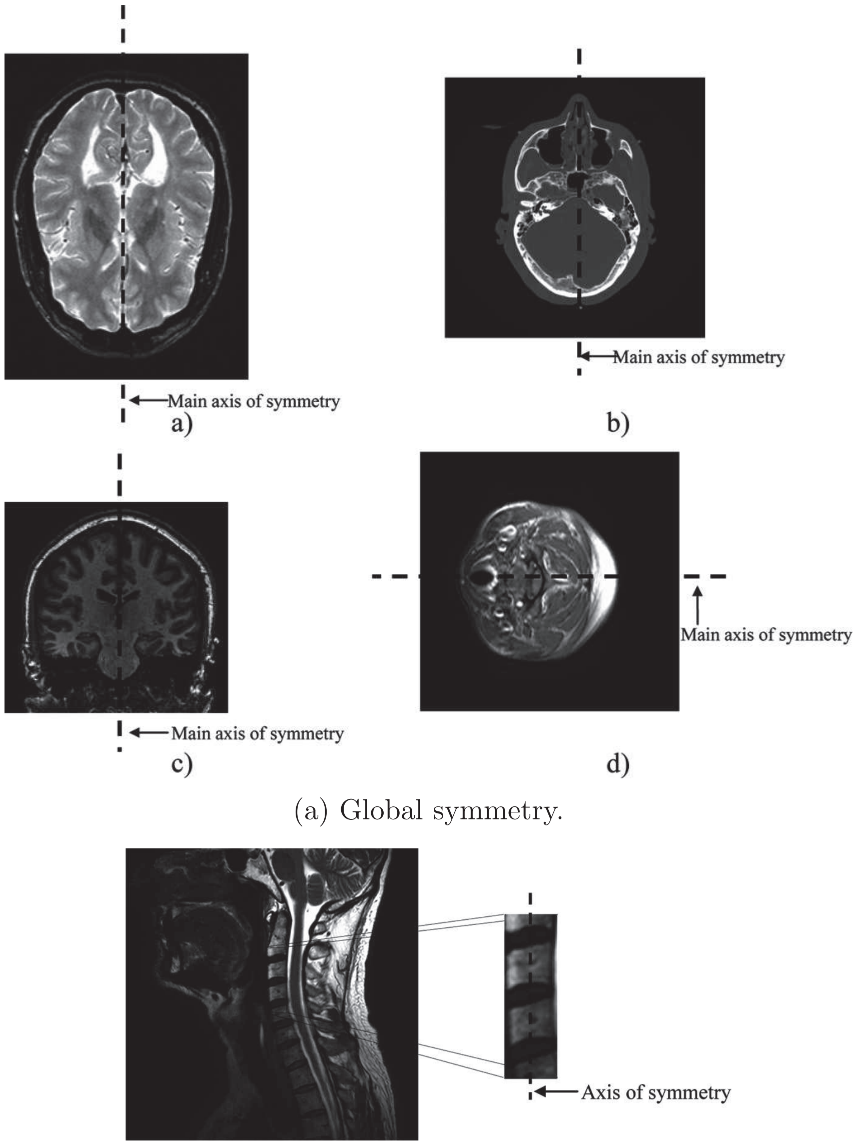
2.5 Other state-of-the-art techniques

In this section a general overview of state-of-the-art research is presented. Current research orientations are discussed, for instance, scalable compression, lossy-to-lossless compression, Region of Interest (ROI) compression, etc.

2.5.1 Scalable lossless compression based on global and local symmetries for 3D medical images

Medical images usually contain inherent symmetries that can be exploited for their compression. Victor Sanchez, et al, in [48, 49], propose to exploit these symmetries as a prediction method. The paper shows that medical images have both global and local symmetries, as seen in Figure 2.21.

The first step in the proposed procedure is to decompose the image in n frequency sub-bands using a 2D-DWT, as the symmetries remain after the transform. Then a block-based prediction is applied, followed by the entropy coding of residual data and transformation parameters. In the block-based intra-band prediction each sub-band is divided in blocks of 16×16 coefficients. Eight spatial transformations are used for the prediction.



(a) Global symmetry.

(b) Local symmetry.

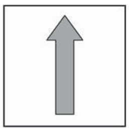
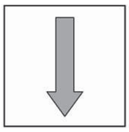
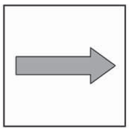
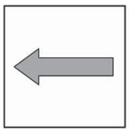
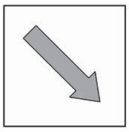
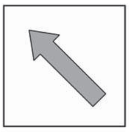
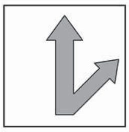
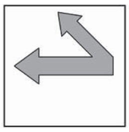
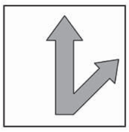
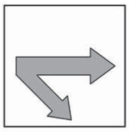
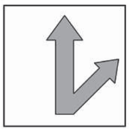
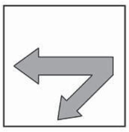
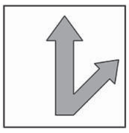
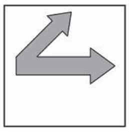
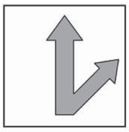
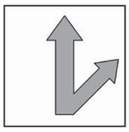
Figure 2.21: Example of the possible symmetries in medical images [48].

These transformations (see Table 2.6) are applied to each already encoded block in the image. The block and transformation pair that best approximate the current block are selected. The blocks of 16×16 coefficients can be further divided into sub-blocks of 8×8 and 4×4 coefficients, if this operation results in a compression performance improvement.

For the compression of the prediction information the authors use variable length codes. For the residue encoding a modified EBCOT is proposed. The paper shows that this method achieves an average improvement of 15% in compression ratio over JPEG2000 and H.264, for lossless compression [48]. Also, this method allows for scalable lossless compression of 3D medical image data.

In [49] some improvements are made to the previous method. Additionally to the previous spatial transformations prediction, the authors added inter-slice prediction and global

Table 2.6: Spatial transformations [48].

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		

symmetry detection. Figure 2.22 shows the block diagram of this method.

The global symmetry is detected using a Fourier-Melin transform. This symmetry is used to assess the scanning order of the blocks. The image is divided in two areas, A and B, regarding the global symmetry. After a block in area A is processed, the next block is the symmetrical positioned block in area B. This allows the prediction to exploit the global symmetry. The spatial transformations prediction of Table 2.6 are then applied to each block.

For the inter-slice prediction a DPCM mode is used. This mode is used to exploit the correlation between sub-band coefficients in adjacent slices. The DPCM mode can be used

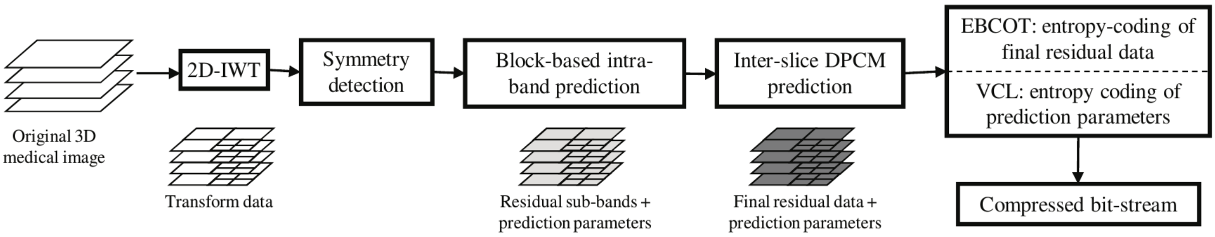

Original 3D medical image
2D-IWT
Transform data
Symmetry detection
Block-based intra-band prediction
Residual sub-bands + prediction parameters
Inter-slice DPCM prediction
Final residual data + prediction parameters
EBCOT: entropy-coding of final residual data
VCL: entropy coding of prediction parameters
Compressed bit-stream

Figure 2.22: Block diagram of the symmetry based scalable lossless compression technique [49].

in five directions, as seen in Figure 2.23. The entropy coding is once again performed using EBCOT. The results show compression efficiency gains of 1%, when compared with the previous method.

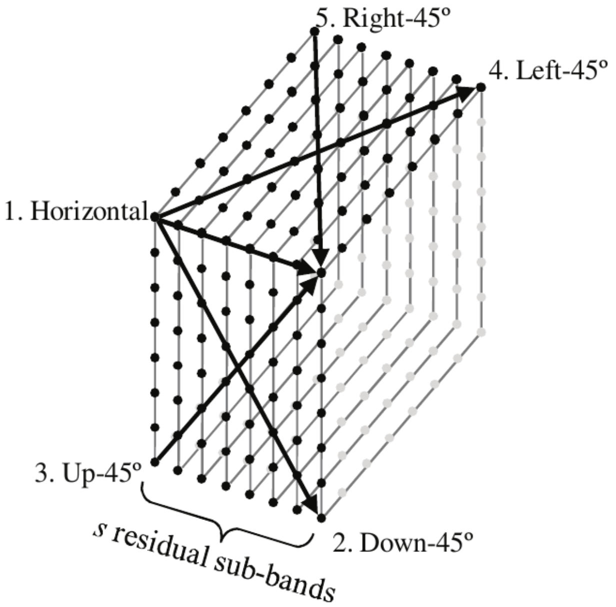


Figure 2.23: Five proposed prediction modes for inter-slice DPCM prediction [49].

2.5.2 Hierarchical oriented prediction for scalable compression of medical images

Jonathan Taquet, et al, propose in [2, 50] a Hierarchical Oriented Prediction (HOP) method for the scalable compression of medical images.

Each prediction level of the HOP algorithm is performed in two steps, as seen in Figure 2.24. In the HStep a horizontal prediction of odd indexed pixels is performed, using already known, causal, pixels. With this step, the image is horizontally sub-sampled. The VStep, performs the same operation but in the vertical direction of the sub-sampled image.

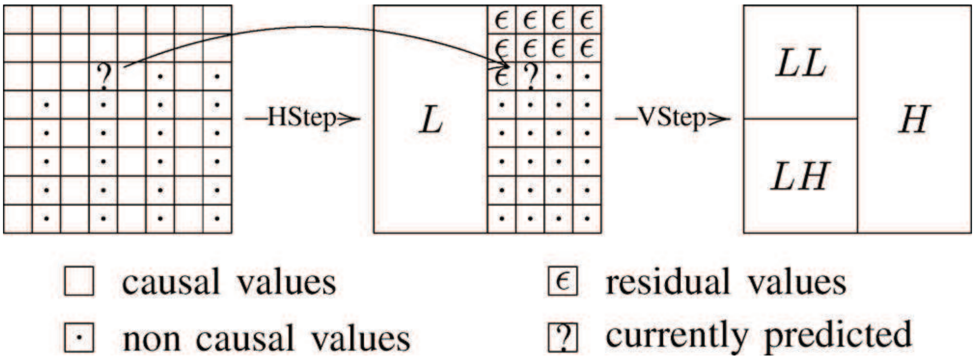


Figure 2.24: One prediction level of the HOP algorithm [2].

The HOP algorithm is designed for prediction of sharp edges in noisy images. An orientation estimation inspired by GAP is performed. This estimation allows the algorithm to choose between five predictors, to perform the prediction along edges. A predictor for homogeneous areas is also available. Additionally, a least-squares estimation can also be performed. This estimation uses an extended set of the causal pixels to perform a dynamic prediction. Therefore, it results in a better adaptation to the image characteristics. The described predictors are used in the HOP algorithm process, for the prediction of odd pixels. In order to avoid systematic errors resulting from static predictors, a prediction bias cancellation is used.

For the coding stage, a residual remapping technique is used before the entropy coder. The authors also extended the proposed algorithm for near-lossless compression. Finally, the results show that, on average, this algorithm is 6.5% more efficient than JPEG2000 and 2.1% than CALIC.

2.5.3 Compression of X-ray angiography based on automatic segmentation

Zhongwei Xu, et al, propose in [51] a diagnostically lossless compression method for X-ray angiography images. This method is based on automatic segmentation of the images, using ray-casting and α-shapes.

Medical images in general, and angiography images in particular, often have a ROI and a background region. The proposed method exploits this fact, essentially by removing the background region before feeding the resulting image to a lossless encoder. As only the background region is removed, this method is diagnostically lossless.

X-ray angiography images are essentially radially symmetric, the method exploits this characteristic to differentiate the ROI from the background region. Initially a pre-processing step is executed in which slice averaging and noise reduction are performed to adapt the sequence to the segmentation step. With this averaging process, a single slice is obtained from the angiography sequence. As all slices are correlated, the boundaries from the regions are preserved.

The segmentation has three steps: identification of the ROI boundaries, linking of the boundary pixels to generate the binary mask and background suppression. Initially, ray-casting is used to determine the pixels in the ROI boundary. With these boundary pixels a closed smooth contour is determined using α-shapes. This contour defines two regions in the binary mask, the background and the ROI.

Finally, the mask is applied to all the slices in a sequence and these resulting images are encoded with a lossless algorithm, such as JPEG-LS. The results show that the proposed automatic segmentation method has a 98.4% accuracy, with respect to the manually performed segmentation. The compression efficiency is improved by 35.3% (2 bpp) compared to JPEG-LS.

2.5.4 Progressive lossless compression

Armando J. Pinho and António J. R. Neves propose in [52, 53] a progressive lossless, i.e., quality scalable, compression scheme. This scheme is based on a hierarchical organization of the intensity levels of an image.

The hierarchical organization is obtained by building a binary tree. Each node of the binary tree represents a subset of the image intensities. Therefore, each node is represented by an intensity value corresponding to the average between the highest and lowest intensities in the subset, I^n.

This tree is built in the encoder and in the decoder by expanding each node. The nodes of the tree are expanded to form two new subsets of the intensities. This procedure is repeated until all nodes are fully expanded, i.e., the number of leaves is equal to the number of intensities.

For each node, a binary mask must be transmitted. This mask indicates the subset of each pixel, as each node is always divided in two subsets. Regions of arbitrary shapes are thus formed, which are also available at the decoder, representing the pixels to analyse in a given node.

This is a quality scalable method, as there is no need to decode all nodes. If all the nodes are not decoded, the resulting image will only have the number of intensities corresponding to the last decoded tree node. If these nodes are not the final ones, the determined I^n is used.

The binary masks are encoded with context modelling and a binary arithmetic coder. For the lossless compression of medical images the results show a compression efficiency gain of 33.1%, on average, compared to JPEG-LS.

2.5.5 Adaptive sequential prediction of multidimensional signals

Xiaolin Wu, et al, propose in [54] an adaptive sequential prediction of multidimensional signals with applications to lossless image coding.

In this work a 2D generalization of the Rissanen universal coding algorithm is proposed. This is performed by coupling it to an adaptive sequencing mechanism. An adapted sequential linear predictor, based on LSP, is implemented, whose order and support are adapted to each pixel, unlike fixed support methods found in the literature.

In order to sequence the causal pixels for the prediction support, correlation is used instead of euclidean distance. Thus, the support will have an arbitrary shape, and might even be sparse, as seen in Figure 2.25.

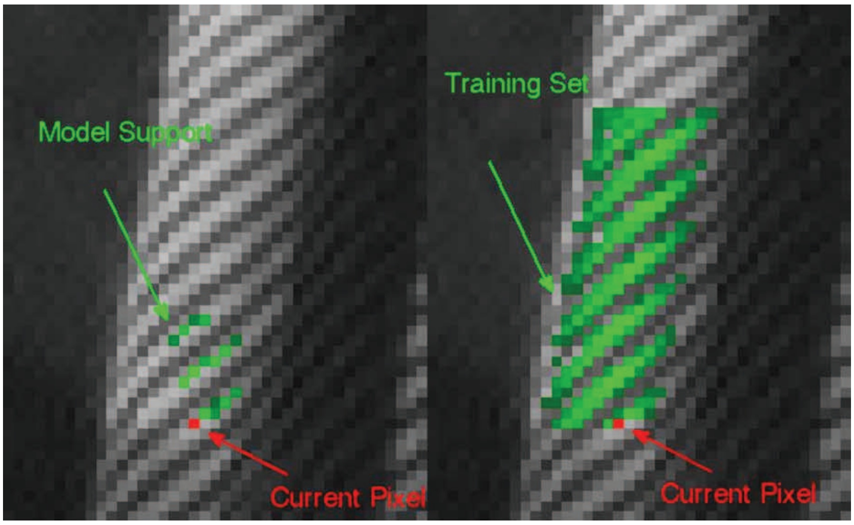


Figure 2.25: Example of the support and training set of the piecewise autoregressive model for a given pixel to predict [54].

To determine the model order, the Minimum Description Length (MDL) principle is used, which prevents model overfitting. The proposed method is called MDL-based adaptive predictor and is an implicit algorithm. To test its efficiency, this method is applied to lossless image coding. The results show a compression efficiency gain of 1.6%, when compared to MRP.

2.6 Summary

This Chapter started by providing an overview of existing medical imaging technologies, with special focus to CT and MRI. This led us to present and characterise the medical dataset used in this work. This discussion was concluded with a review of the DICOM standard, that regulates communications and archiving of medical images. It was shown that the use of lossy compression for medical images is still an open debate, and that the DICOM standard does not make recommendations in this subject.

Considering this, it was decided to focus this research on lossless compression techniques. Hence, a review of DICOM and state-of-the-art lossless encoders was made. A comparison between these encoders shows that MRP has the highest compression efficiency for still image codecs. Nevertheless, HEVC RExt, with the Random Access profile, has the highest overall compression efficiency. Due to its high efficiency, with a difference of 0.35 bpp to HEVC RExt, MRP was the encoder chosen as the base of this work.

Finally, this Chapter is concluded with a review of other state-of-the-art medical image compression techniques. These techniques are mainly comprised in the following areas: scalable, lossy-to-lossless and ROI oriented compression. This review showed that the medical image compression field is a relevant topic in the image processing research community.

Chapter 3

Processing techniques for the compression of medical images

In this chapter several processing techniques are applied to the medical images, prior to being compressed. These techniques are used to better exploit some of the Minimum Rate Predictors (MRP) encoder characteristics and increase its compression efficiency. Each technique is described and the corresponding results analysed. To finalise the chapter a comparison of the processing techniques is performed and the achieved results are discussed.

Some of the used encoders, namely Multi-scale Multidimensional Parser (MMP) and MRP, require input images with dimensions that are multiples of 16, for MMP, and 8, for MRP. Some of the proposed processing techniques will change the geometry of the sequences, therefore it is required to insure that the number of slices used in each sequence is multiple of 16, in order to compare the achieved results for each technique, by each encoder. Table 3.1 shows the number of slices used for each sequence, and Table 3.2 replicates the results of Tables 2.3 and 2.4, for the number of slices given in Table 3.1.

Table 3.1: Number of slices used by each processing techniques.

Sequence	Slices
Aperts	96
carotid	64
skull	192
wrist	176
liver_t1	48
liver_t2e1	48
ped_chest	64
sag_head	48

Table 3.2: Coding performance evaluation for lossless encoders using the number of slices given in Table 3.1 (results in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	1.200	1.142	0.734	1.184	1.267	0.945	1.064	1.004	0.776
carotid	2.017	1.955	1.383	1.936	1.981	1.507	1.739	1.647	1.347
skull	3.284	2.981	1.822	3.058	3.079	2.143	2.847	2.713	2.411
wrist	1.939	1.919	1.024	1.743	1.782	1.261	1.653	1.574	1.190
liver_t1	3.493	3.399	2.118	3.383	3.257	2.409	3.158	3.019	2.566
liver_t2e1	2.754	2.512	1.552	2.416	2.530	1.771	2.369	2.223	1.688
ped_chest	3.066	3.037	1.515	3.061	3.015	2.055	2.928	2.780	2.327
sag_head	2.639	2.599	1.771	2.810	2.908	2.190	2.585	2.522	2.281
Average	2.549	2.443	1.490	2.449	2.477	1.785	2.293	2.185	1.823

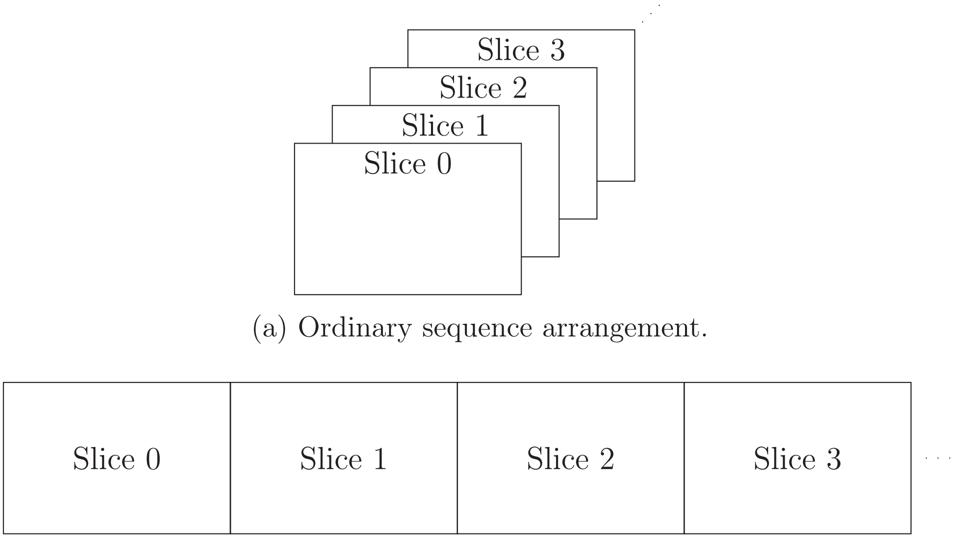
In Table 3.2, the third and fourth columns represent, respectively, HEVC RExt Intra and Random Access profiles, for convenience this nomenclature will be used in the rest of the chapter. This table is shown here to be used as a comparison reference to the results that are shown in this chapter. These results are slightly different from the ones shown in Tables 2.3 and 2.4 due to the difference on the number of used slices, as slices generally have different characteristics.

In the following sections each processing technique will be presented, namely concatenation of slices, directional approaches, "pre-processing" inter-slice prediction and histogram packing.

3.1 Concatenation of slices

Some encoding algorithms, such as MRP and MMP, optimise the encoding process for the input image, instead of using more generic predictors, such as JPEG-LS or CALIC. Considering this, a method can be developed in order to take advantage of such optimisation, which may be extended to encode several images.

These characteristics can be exploited by concatenating all the slices in a sequence, prior to the encoding process. An example of a possible slice concatenation can be seen in Figure 3.1, in which slices are concatenated side-by-side. In our experimental set-up this type of concatenation was used, because then all the sequence is going to be treated by the encoder as a single image. Thus, instead of needing to optimise and transmit the information regarding to the encoding process for each image, this optimisation is only performed once and, therefore, the amount of information to transmit is much lower. This method has been applied for various medical image sequences, which were then compressed using an encoder algorithm.



(a) Ordinary sequence arrangement.

(b) Concatenated slices of a sequence.

Figure 3.1: Example of slice concatenation.

The results for this approach are shown on Table 3.3. Due to the dimensions of the resulting images, namely sequences *skull* and *wrist*, we were not able to encode this concatenated sequences with H.264 and JPEG-LS as their dimensions exceed encoders requirements. Therefore it was decided to use the previous results for these sequences, that is why in Table 3.4 a 0% difference appears for them, thus allowing to compare their average result. Table 3.4 shows the compression efficiency gains achieved by encoding the concatenated slices. In the following tables, a positive percentage indicates an improvement in the compression efficiency.

Table 3.3: Compression results of the concatenated slices (in bpp).

Sequence	H.264	Intra	MMP	JPEG 2000	JPEG-LS	CALIC	MRP
Aperts	1.177	1.136	1.017	1.246	1.011	0.894	0.736
carotid	2.003	1.949	1.757	1.955	1.698	1.548	1.276
skull	3.284	2.973	2.882	3.051	2.847	2.639	2.355
wrist	1.939	1.911	1.605	1.757	1.653	1.492	1.139
liver_t1	3.472	3.393	3.146	3.233	3.106	2.903	2.496
liver_t2e1	2.742	2.504	2.197	2.506	2.316	2.107	1.571
ped_chest	3.040	3.029	2.781	3.000	2.873	2.663	2.264
sag_head	2.618	2.592	2.546	2.883	2.534	2.434	2.243
Average	2.534	2.436	2.241	2.454	2.255	2.085	1.760

As expected, MMP, CALIC and MRP encoders show a higher compression efficiency than

Table 3.4: Percentage of improvement on coding performance when using concatenated slices (in %).

Sequence	H.264	Intra	MMP	JPEG 2000	JPEG-LS	CALIC	MRP
Aperts	1.9	0.6	14.1	1.7	5.0	10.9	5.2
carotid	0.7	0.3	9.2	1.3	2.3	6.0	5.3
skull	0.0	0.3	5.8	0.9	0.0	2.7	2.3
wrist	0.0	0.4	7.9	1.4	0.0	5.2	4.3
liver_t1	0.6	0.2	7.0	0.8	1.7	3.9	2.7
liver_t2e1	0.4	0.3	9.0	1.0	2.2	5.2	7.0
ped_chest	0.9	0.3	9.1	0.5	1.9	4.2	2.7
sag_head	0.8	0.3	9.4	0.8	2.0	3.5	1.6
Average	0.6	0.3	8.5	1.0	1.7	4.6	3.5

other encoders when slices are concatenated. MMP shows an average gain of, approximately, 8.5%, CALIC shows a gain of 4.6%, and MRP has a gain of 3.5%. The encoders with the lower compression efficiency are HEVC RExt Intra and H.264. These results were expected as the intra compression tools of these encoders are not adapted for each input image.

In Chapter 2, it was shown that MMP is a dictionary based encoder. This type of encoders build a dictionary of symbols, extracted from the causal data of the image, which are used to encode an image, or its residue. When we concatenate all slices of a sequence, in the encoding process, we are building a dictionary that will be used for all slices. Therefore, there is the advantage of not having to design a dictionary for each slice, which takes time to be adapted to the source, but, instead, an unique dictionary already adapted from the first slice for the remaining sequence is used.

In the case of MRP, as described in Chapter 2, it optimises a set of classes, each one describing a predictor. In this algorithm, the input is divided into blocks, and each block is assigned to a class, depending on the performed optimisations. These classes are optimised for the input image, thus, when the concatenated images are used, the classes, and the predictors, are optimised for all slices. Thus, the compression efficiency may be improved, as there is no need to optimise, and transmit, extra classes.

As can be observed from Table 3.4, all the tested encoders show some compression gains. This can be, in part, explained by the cost of transmitting the header of each slice, which are not used in this case due to the concatenation. Despite the presented compression efficiency improvements, these results are still bellow those achieved by HEVC RExt Random Access as shown in Table 3.2.

3.2 Directional approaches

A medical exam, such as a CT or an MRI is, in its essence, a representation of a three-dimensional object, which is usually the human body or part of it. The direction of the image scan may be dependent on the body morphology or the diagnostics specialist preference. Nevertheless, for data compression purposes the best direction may be different from that used by the scanner equipment, and a different direction may result in a more efficient compression, as each direction results in slices with different characteristics.

3.2.1 Slice formation on different axes

On a three-dimensional dataset, a group of slices can be aligned in one of three axes, X, Y or Z, that we call a coordinate system. If we consider that the usual direction in which the slices are in medical exams is the Z axis, with the slices formed in the XY plane, a representation of the volumetric dataset looks like the one in Figure 3.2.

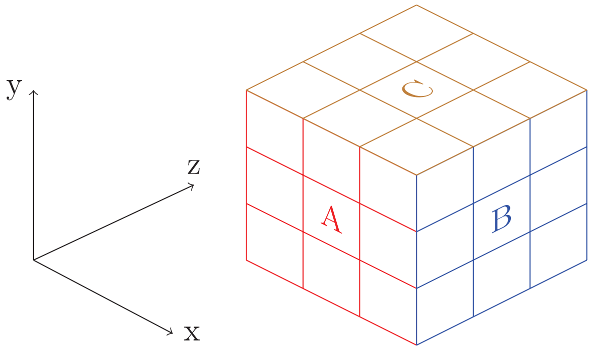
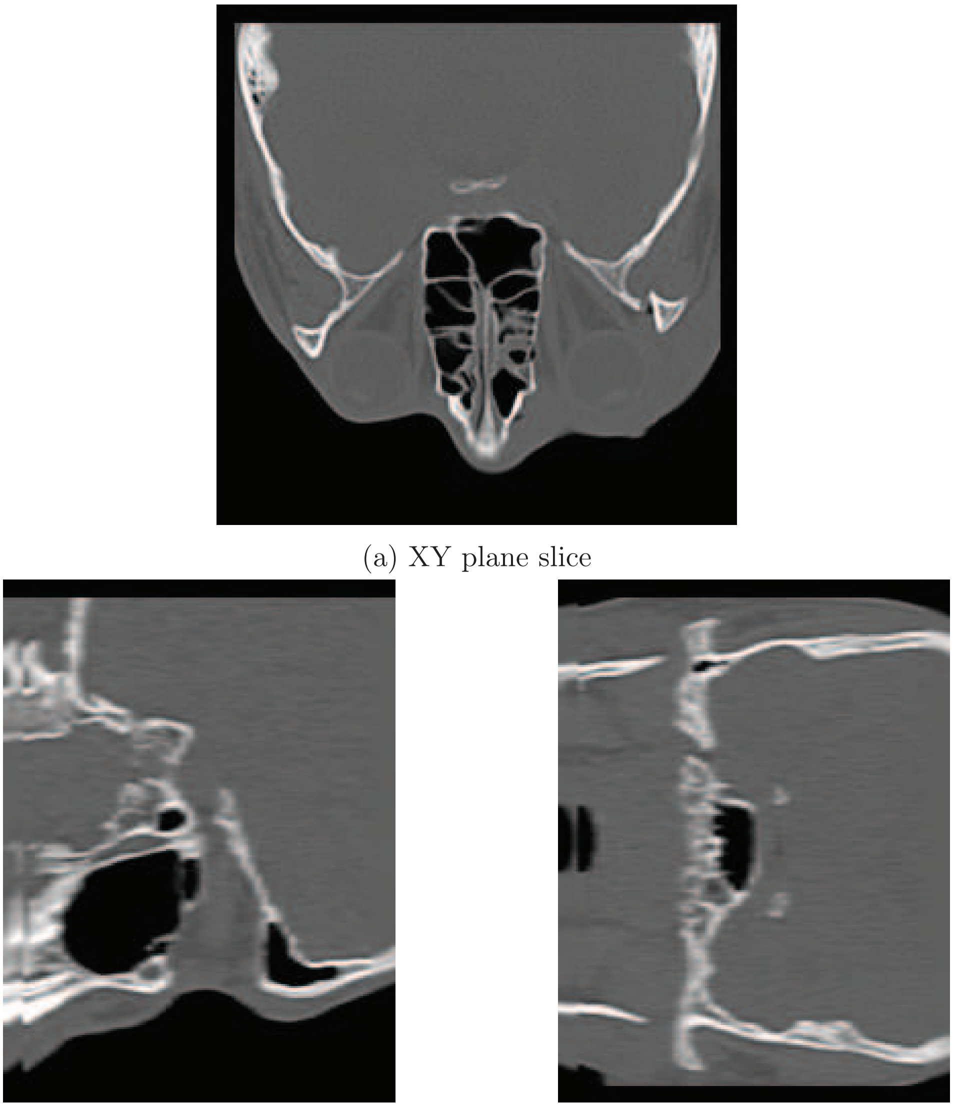


Figure 3.2: Slice orientation on a medical dataset.

As we can see from this figure, slices can be extracted in different directions. 'A' type slices are parallel to the XY plane following in the Z axis direction, 'B' slices are parallel to the YZ plane and, finally, 'C' slices are parallel to the XZ plane. In Figures 3.3 and 3.4 a representation of a slice in each of these alignments is shown, for *skull* and *ped_ chest* sequences, respectively.

Each of these slices, in the these planes, has different characteristics. It is expected that when we change the plane of the slices some of the inter-slice redundancy may be exploited within a single slice in a different plane. Thus, the 'new' slices might have a higher spatial redundancy. If this is the case intra encoders, such as MRP, can exploit some of the Z axis inter-slice redundancy.



(a) XY plane slice

(b) YZ plane slice

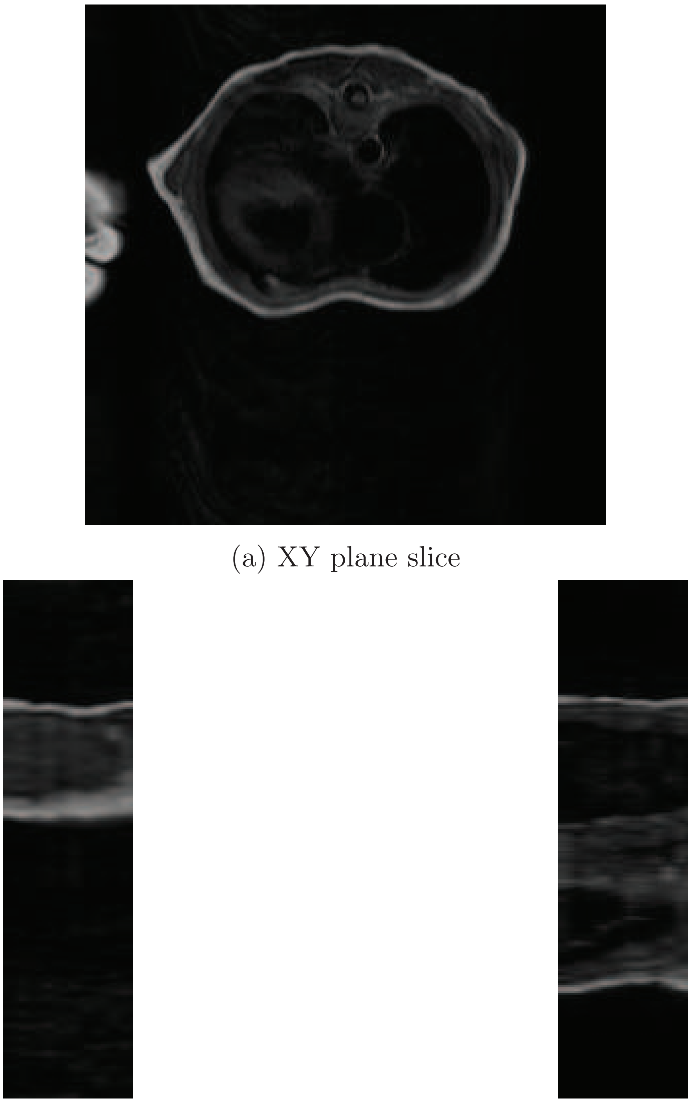
(c) XZ plane slice

Figure 3.3: Slice 101 of *skull* sequence for each plane.

3.2.2 Experimental tests

In order to evaluate the most efficient coding direction, each was tested to align the sequences. The results of this approach are shown in Table 3.5, for slices formed in YZ plane, and in Table 3.7, for slices formed in XZ plane. Tables 3.6 and 3.8 show the compression efficiency improvements for YZ plane and XZ plane, respectively.

These results show that the inter-slice redundancy is lower, for slices formed in planes different of the XY, as HEVC RExt Random Access presents a lower compression efficiency. This difference in HEVC RExt Random Access is of 11.7% and 10.2% when using slices formed in the YZ and in the XZ planes, respectively. The same thing happens for JP3D, that shows differences of 21.0% and 29% for slices in the YZ and in the XZ planes, respectively.



(a) XY plane slice

(b) YZ plane slice

(c) XZ plane slice

Figure 3.4: Slice 51 of *ped_ chest* sequence for each plane.

Table 3.5: Compression results for slices aligned with the YZ plane (in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	1.074	1.008	0.771	1.125	1.134	1.630	0.937	0.888	0.676
carotid	1.777	1.701	1.424	1.735	1.723	1.910	1.459	1.458	1.150
skull	2.493	2.314	2.038	2.352	2.227	2.850	2.127	1.970	1.599
wrist	1.651	1.602	1.136	1.460	1.483	1.734	1.357	1.289	0.962
liver_t1	3.261	3.106	2.317	3.128	2.717	3.161	2.557	2.583	2.013
liver_t2e1	2.486	2.289	1.889	2.328	2.059	2.675	1.945	1.944	1.393
ped_chest	2.140	2.003	1.713	2.071	1.966	2.275	1.747	1.712	1.322
sag_head	2.417	2.325	2.033	2.498	2.358	2.580	2.059	2.081	1.687
Average	2.162	2.043	1.665	2.087	1.958	2.352	1.773	1.741	1.350

Table 3.6: Percentage of compression gain when changing the plane direction from XY to YZ.

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	10.5	11.8	-5.1	5.0	10.5	-72.4	11.9	11.5	12.9
carotid	11.9	13.0	-3.0	10.4	13.0	-26.8	16.1	11.5	14.6
skull	24.1	22.4	-11.9	23.1	27.7	-33.0	25.3	27.4	33.7
wrist	14.9	16.5	-10.9	16.2	16.7	-37.6	17.9	18.1	19.1
liver_t1	6.6	8.6	-9.4	7.6	16.6	-31.2	19.0	14.4	21.5
liver_t2e1	9.7	8.9	-21.7	3.6	18.6	-51.0	17.9	12.5	17.5
ped_chest	30.2	34.0	-13.1	32.3	34.8	-10.7	40.4	38.4	43.2
sag_head	8.4	10.6	-14.7	11.1	18.9	-17.8	20.4	17.5	26.0
Average	15.2	16.4	-11.7	14.8	21.0	-31.8	22.7	20.3	25.9

Table 3.7: Compression results for slices aligned with the XZ plane (in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	1.090	1.037	0.757	1.113	1.157	1.572	0.943	0.873	0.691
carotid	1.792	1.720	1.391	1.842	1.763	1.864	1.530	1.518	1.231
skull	2.517	2.339	2.006	2.411	2.285	2.776	2.198	2.004	1.639
wrist	1.513	1.483	1.209	1.276	1.259	1.912	1.154	1.112	0.821
liver_t1	3.243	3.088	2.335	3.084	2.705	3.143	2.510	2.535	2.052
liver_t2e1	2.568	2.319	1.737	2.430	2.144	2.393	2.053	2.005	1.642
ped_chest	2.198	2.030	1.670	2.198	2.103	2.175	1.844	1.848	1.460
sag_head	2.422	2.335	2.036	2.522	2.366	2.590	2.068	2.085	1.682
Average	2.168	2.044	1.643	2.110	1.972	2.303	1.788	1.747	1.402

Table 3.8: Percentage of compression gain when changing the plane direction from XY to XZ.

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	9.1	9.2	-3.2	6.0	8.7	-66.3	11.3	13.0	11.0
carotid	11.2	12.0	-0.6	4.9	11.0	-23.7	12.0	7.9	8.6
skull	23.4	21.5	-10.1	21.1	25.8	-29.6	22.8	26.1	32.0
wrist	22.0	22.7	-18.0	26.8	29.4	-51.7	30.2	29.4	31.0
liver_t1	7.2	9.2	-10.2	8.8	17.0	-30.5	20.5	16.0	20.0
liver_t2e1	6.7	7.7	-11.9	-0.6	15.3	-35.1	13.3	9.8	2.7
ped_chest	28.3	33.1	-10.2	28.2	30.3	-5.8	37.0	33.5	37.2
sag_head	8.2	10.2	-14.9	10.2	18.6	-18.3	20.0	17.3	26.2
Average	15.0	16.3	-10.2	13.9	20.4	-29.0	22.0	20.0	23.1

As expected, intra encoders showed the highest gains in compression efficiency, namely MRP, with up to 25.9% and 23.1% of improvement, when using slices formed in the YZ and in the XZ planes, respectively. In Table 3.2, HEVC RExt, with the Random Access profile, presented the best compression efficiency, of the studied encoders, with 1.490 bpp. Using this technique to align slices in different planes, rather than the usual XY, MRP arises as the encoder with the highest compression efficiency, 1.35 bpp, for slices aligned with the YZ plane.

The loss in efficiency of the inter encoder HEVC RExt Random Access can be explained by the decrease on similarity between adjacent slices aligned with directions different than Z axis. Thus, the motion estimation prediction of the encoder will use less accurate references for the inter prediction, therefore, resulting in a lower compression efficiency.

For the intra encoders, their higher performance can be explained by the increase on the spatial correlation of the newly formed slices. With a higher spatial correlation, the compression efficiency of the intra encoders is, inevitably, improved.

The achieved results for the tested dataset show that the most efficient plane direction on average to perform the encoding is YZ, closely followed by XZ plane. It was clearly shown that changing the direction in which the slices are formed improves the compression efficiency of the intra encoders. This was an expected result due to the scanned image characteristics, not relying specifically on the image alignment.

3.2.3 Optimal compression plane algorithm

Anmin Liu, et al, describe in [55] and [56], an algorithm to automatically choose the best plane direction to align the sequence slices for compression. In these papers a preprocessing technique is proposed, the Optimal Compression Plane (OCP), which determines the plane alignment automatically, prior to the compression stage.

The algorithm uses a Correlation Coefficient (CC), which is calculated in the three possible slice planes, XY, XZ and YZ. In the proposed framework, the OCP is calculated and the resulting frames are encoded with a standard coding method. The OCP algorithm can be used to determine the plane for both intra-only and inter prediction schemes through Equation 3.1 and Equation 3.2, for intra only and inter prediction schemes, respectively.

$$OCP_{intra} = \begin{cases} XY, \text{if } \min\{C_Z, C_X, C_Y\} = C_Z \\ XZ, \text{if } \min\{C_Z, C_X, C_Y\} = C_Y \\ YZ, \text{if } \min\{C_Z, C_X, C_Y\} = C_X. \end{cases} \tag{3.1}$$

$$OCP_{inter} = \begin{cases} XZ, \text{if } C_X \geqslant C_Y \wedge C_T - \max C_X, C_Y > T_C \\ YZ, \text{if } C_X < C_Y \wedge C_T - \max C_X, C_Y > T_C \\ XY, \text{otherwise.} \end{cases} \tag{3.2}$$

C_A represents the CC along axis A, and T_C is a threshold, that accounts for the efficiency of motion estimation prediction on the XY plane direction, the Z axis.

The compression performance is improved by using the OCP, both for lossless and lossy compression. In the lossless case, for JPEG-LS, a saving of 22% bpp is reported. For the lossy case, i.e. H.264, a BD-Bitrate of -11.01% and a BD-PSNR of 0.89 dB is reported. Also, it is shown that in most cases, using this technique prior to the encoding process results in similar or better results.

Experimental results using medical images

The achieved results when using the OCP algorithm for medical images are shown in Table 3.9. In this table, the CC results are shown for each axis, in columns five to seven, and also the OCP choice for intra and inter type predictions are determined based on Equations 3.1 and 3.2, respectively, in columns three and four. Finally, in column one, results for the best plane are shown from the results of Section 3.2.2, only for comparison.

Table 3.9: Correlation coefficients for all directions and choice of the best OCP.

Sequence	Section 3.2.2	Intra	Inter	Cy	Cx	Ct
Aperts	YZ	XZ	XY	0.900	0.981	0.984
carotid	YZ	XZ	XZ	0.771	0.909	0.981
skull	YZ	YZ	YZ	0.902	0.901	0.990
wrist	XZ	XZ	XY	0.749	0.978	0.992
liver_t1	YZ	XZ	XY	0.945	0.981	0.975
liver_t2e1	YZ	XZ	XY	0.760	0.975	0.975
ped_chest	YZ	YZ	YZ	0.936	0.814	0.981
sag_head	XZ	XZ	XZ	0.752	0.796	0.967

From this table, it is possible to observe that only for four sequences, in case of inter prediction, the optimal plane is not the usual XY plane. These four sequences are *carotid*, *skull*, *ped_chest* and *sag_head*. In the case of intra prediction the optimal plane is different than the usual one for all sequences. When comparing these results with the best plane described in Section 3.2.2, we can see that only in half the cases, four out of eight, the OCP algorithm chooses the best plane for compression, according to the best compression option for MRP, which is the main focus of this work.

As MRP is an intra encoder, from now the focus is the intra case. For this porpuse, we show in Table 3.10 the results obtained by using the OCP algorithm with the lossless encoders under research. Additionally in Table 3.11 the changes in efficiency due to the use of the OCP algorithm are shown, when compared to the encoding of slices aligned to YZ plane, which provided the best results in Section 3.2.2.

Table 3.10: Intra compression results using OCP algorithm (in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	1.090	1.037	0.757	1.113	1.157	1.572	0.943	0.873	0.691
carotid	1.792	1.720	1.391	1.842	1.763	1.864	1.530	1.518	1.231
skull	2.493	2.314	2.038	2.352	2.227	2.850	2.127	1.970	1.599
wrist	1.513	1.483	1.209	1.276	1.259	1.912	1.154	1.112	0.821
liver_t1	3.243	3.088	2.335	3.084	2.705	3.143	2.510	2.535	2.052
liver_t2e1	2.568	2.319	1.737	2.430	2.144	2.393	2.053	2.005	1.642
ped_chest	2.140	2.003	1.713	2.071	1.966	2.275	1.747	1.712	1.322
sag_head	2.422	2.335	2.036	2.522	2.366	2.590	2.068	2.085	1.682
Average	2.168	2.037	1.652	2.086	1.948	2.325	1.767	1.726	1.380

Table 3.11: Percentage of compression gain when using OCP algorithm, compared to the compression in YZ plane aligned slices.

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	-1.5	-2.9	1.8	1.1	-2.0	3.5	-0.7	1.7	-2.2
carotid	-0.8	-1.1	2.3	-6.1	-2.3	2.4	-4.9	-4.1	-7.0
skull	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
wrist	8.4	7.4	-6.4	12.6	15.1	-10.2	14.9	13.7	14.6
liver_t1	0.6	0.6	-0.8	1.4	0.5	0.6	1.8	1.9	-1.9
liver_t2e1	-3.3	-1.3	8.0	-4.4	-4.1	10.5	-5.5	-3.1	-17.9
ped_chest	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
sag_head	-0.2	-0.4	-0.2	-1.0	-0.3	-0.4	-0.5	-0.2	0.3
Average	-0.3	0.3	0.8	0.0	0.5	1.2	0.4	0.8	-2.2

These tables show that for most encoders there is an increase on the compression efficiency, with the exception of MRP, which has a decrease in compression efficiency of 2.2%, and H.264, that has a loss of efficiency of 0.3%, when compared to the compression in the YZ plane. The MRP encoder losses can be explained by the fact that for intra coding the OCP algorithm only correctly chooses the best plane for half of the sequences, as can be observed from Table 3.9. The remainder of the encoders have a maximum increase in efficiency of 1.2%, for JP3D, compared to the results of Table 3.5. It is interesting to notice that two encoders, that benefit the most from the OCP algorithm, exploit inter-slice redundancy, HEVC RExt and JP3D, when the algorithm was adjusted for intra encoding, as in this case.

Despite the losses in compression efficiency, the overall results are still better than those related to the YZ plane, and much better than those in the XY plane. This way, this is an automatic plane selection method that overall produces good results, but it could be optimised to work with MRP.

3.3 Inter-slice prediction technique

In the previous section it was argued that, when slices are aligned in planes different that the usual XY plane, the usual inter-slice redundancy is exploited by intra encoders. In this section an alternative method is presented, in order to explore such redundancy outside of the intra encoders.

This alternative approach consists on using an inter-slice predictor prior to the encoding process. Thus, after applying the predictor a residual sequence still remains, that will then be encoded by the lossless encoders, and, therefore, the inter-slice redundancy is exploited, in a different way. In the following sections the used inter-slice predictors and their implementation results will be described and discussed.

3.3.1 Pixel-wise difference predictor

A simple pixel-wise difference predictor, between co-located pixels in adjacent slices, was implemented, keeping only the first slice as reference, as shown in the example of Figure 3.5. In this figure, $x(k)$, $y(k)$ and $z(k)$ represent the pixels in a slice. The result of the predictor is given by $z(1) = y(1) - x(1)$, ... , $z(k) = y(k) - x(k)$.

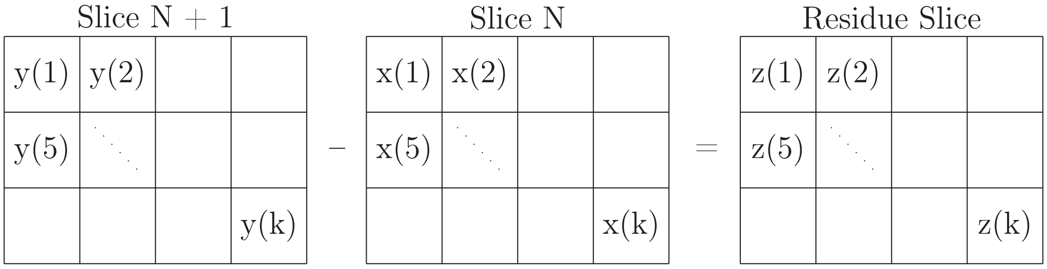


Figure 3.5: Example of the pixel-wise difference predictor.

Usually, the residual sequence samples would be represented by 9 bits. However the residual for these sequences only have 8 bits-per-pixel. This is a relevant fact, as most used encoders only allow 8 bits-per-pixel input images. The results of encoding the residue generated by the pixel-wise difference predictor are shown in Tables 3.12 and 3.13.

As expected, these tables show that intra encoders have an increase on the compression

Table 3.12: Results of the encoding of the pixel-wise difference residue (in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	0.791	0.776	0.679	0.885	0.933	0.890	0.790	0.771	0.630
carotid	1.459	1.442	1.243	1.500	1.546	1.380	1.354	1.318	1.115
skull	2.071	2.045	1.488	2.141	2.173	1.924	2.011	1.935	1.696
wrist	1.115	1.117	0.879	1.145	1.223	1.171	1.049	1.064	0.862
liver_t1	2.216	2.211	1.897	2.256	2.287	2.243	2.098	2.049	1.790
liver_t2e1	1.781	1.774	1.245	1.714	1.815	1.632	1.681	1.597	1.307
ped_chest	1.576	1.580	1.292	1.696	1.761	1.785	1.577	1.536	1.324
sag_head	1.983	1.979	1.538	2.216	2.177	2.046	1.995	1.975	1.789
Average	1.624	1.615	1.282	1.694	1.739	1.634	1.569	1.531	1.314

Table 3.13: Percentage of compression gain when using the pixel-wise difference predictor.

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	34.1	32.1	7.5	25.2	26.4	5.8	25.8	23.2	18.9
carotid	27.7	26.2	10.1	22.5	22.0	8.4	22.1	20.0	17.2
skull	37.0	31.4	18.3	30.0	29.4	10.2	29.3	28.7	29.7
wrist	42.5	41.8	14.2	34.3	31.4	7.1	36.6	32.4	27.6
liver_t1	36.6	35.0	10.5	33.3	29.8	6.9	33.6	32.1	30.2
liver_t2e1	35.3	29.4	19.8	29.0	28.3	7.8	29.0	28.1	22.6
ped_chest	48.6	48.0	14.7	44.6	41.6	13.2	46.1	44.7	43.1
sag_head	24.9	23.9	13.2	21.1	25.2	6.6	22.8	21.7	21.6
Average	36.3	33.9	13.9	30.8	29.8	8.5	31.6	30.0	27.9

efficiency up to 36.3% for H264, resulting from a difference of 0.93 bpp. The MRP encoder has an increase in efficiency of 27.9%, resulting from a difference of 0.51 bpp. Notwithstanding, the inter encoders also show an increase in the compression efficiency, of up to 13.9% for HEVC RExt Random Access, resulting from a 0.21 bpp difference.

The encoder that evidences the highest compression efficiency gain is the H.264, with 36.3%, followed by HEVC RExt Intra, with 33.9%. It is worth to note that the results for HEVC RExt Intra, 1.62 bpp, are close to the result of HEVC RExt Random Access from Table 3.2, 1.49 bpp. Previously there was a difference in both profiles performance of almost 1.0 bpp, but now the difference is only 0.13 bpp.

The encoder that produces the most efficient result is, nonetheless, the HEVC RExt Random Access, with 1.28 bpp, followed by MRP, with 1.31 bpp. These results show a slightly more efficient compression than the one given in Section 3.2, for MRP, although the results show that HEVC RExt Random Access is now slightly more efficient. This can be explained with the higher efficiency that the pixel-wise difference predictor has for exploiting the inter-slice similarities, when compared with the changing of the slices alignment.

As stated in Chapter 2, medical sequences such as Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are volumetric images, consisting of multiple adjacent slices of the human body or part of it. These slices are anatomically or physiologically correlated [30], i.e., there are many similarities between them. Therefore, when we use the pixel-wise difference predictor we are essentially cancelling equal co-located pixels in adjacent slices. This means that these areas of equal pixels will be smooth on the residue image, as can be seen in Figure 3.6a. Consequently, the residue image will be comprised by smooth areas and sharp edges, related to the internal structures of the body. This is true because there is essentially little to no movement from slice to slice, apart from expansions or contractions of the organs, for instance.

Thus, intra encoders will have a higher compression efficiency due to the spatial redundancy of the residue slices. As for inter encoders, due to the relative smoothness of the residue slices, they will also be able to perform an efficient prediction in the residue slices, for both intra and inter prediction.

3.3.2 HEVC RExt Random Access prediction

An alternative method to perform an inter slice prediction outside of the encoders is to use the residue generated by the temporal prediction of a video encoder. State-of-the-art video encoders, such as HEVC, make use of several highly complex temporal prediction tools, and therefore can better exploit the inter-slice similarities. HEVC is the most recently approved video encoder standard and, together with its Range Extension, which has additional tools for lossless compression, as explained in Chapter 2, is the encoder with the highest efficiency for lossless compression, see Table 3.2. Considering this, HEVC RExt was selected to generate the prediction residue, using the Random Access profile for the lossless encoding mode.

However, unlike in Section 3.3, where the pixel-wise difference predictor returns a 8 bits-per-pixel residual, in this case that is not true. Thus, Table 3.14 shows the number of pixels, for each sequence, where 8 bits are not enough to represent the residue. In this approach, each overflow pixel, will cost eight bits to transmit the extra information, which is negligible, as we can see in the third column of this table. Both this cost and the cost to generate the residual image, represented in the fourth column, are taken into account in the results. In Table 3.15 the results for the encoding of the HEVC RExt prediction residue are shown, while Table 3.16 shows the difference in compression gain from the results presented in Section 3.3.1.

Table 3.14: Number of lossy pixels of the HEVC RExt residual.

Sequence	Lossy pixels	Lossy Pixels Cost (in bpp)	Prediction Cost (in bpp)
Aperts	48	6.1×10^{-5}	0.043
carotid	63	1.2×10^{-4}	0.049
skull	1902	1.2×10^{-3}	0.041
wrist	217	1.5×10^{-4}	0.025
liver_t1	1	2.5×10^{-6}	0.053
liver_t2e1	197	5.0×10^{-4}	0.036
ped_chest	14	2.7×10^{-5}	0.037
sag_head	28	7.1×10^{-5}	0.057

Table 3.15: Results of the encoding of the HEVC RExt residue (in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	0.762	0.762	0.737	0.874	0.912	1.016	0.797	0.785	0.679
carotid	1.416	1.425	1.391	1.509	1.527	1.646	1.377	1.356	1.207
skull	1.884	1.853	1.776	1.926	2.001	2.386	1.857	1.768	1.619
wrist	1.043	1.046	1.012	1.071	1.176	1.478	1.045	1.019	0.881
liver_t1	2.137	2.140	2.082	2.209	2.267	2.755	2.100	2.014	1.856
liver_t2e1	1.590	1.575	1.492	1.587	1.675	2.038	1.545	1.467	1.302
ped_chest	1.537	1.543	1.507	1.622	1.745	2.228	1.621	1.525	1.376
sag_head	1.807	1.804	1.748	1.965	2.028	2.381	1.845	1.808	1.666
Average	1.522	1.519	1.468	1.595	1.666	1.991	1.523	1.468	1.323

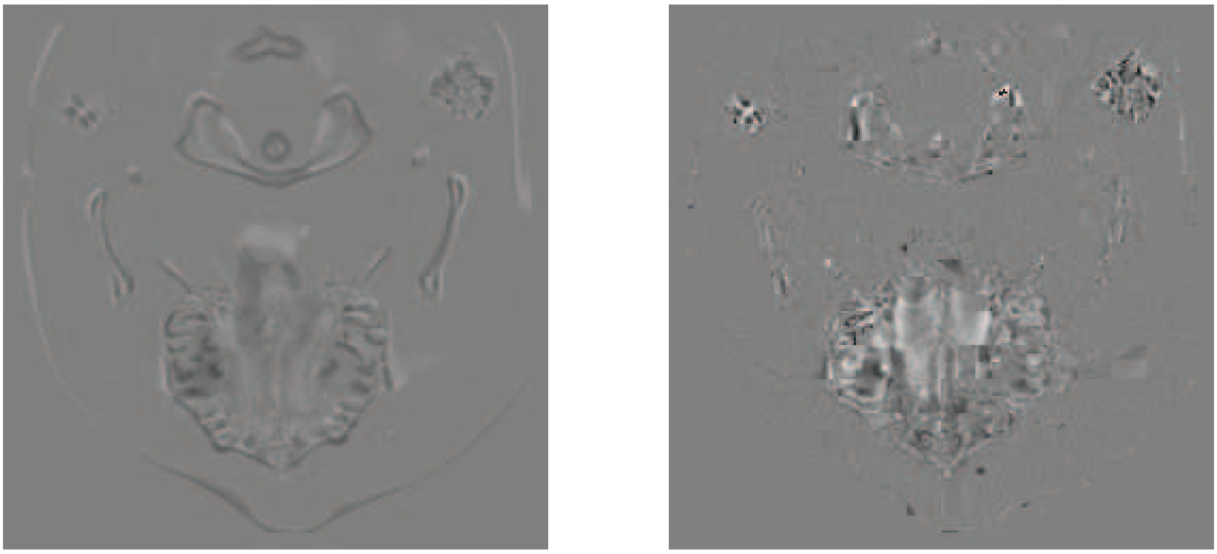
Table 3.16: Percentage of compression gain when using the HEVC RExt residue, compared to the pixel-wise difference predictor.

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	3.601	1.769	-8.579	1.286	2.231	-14.190	-0.922	-1.874	-7.869
carotid	2.905	1.184	-11.940	-0.592	1.223	-19.241	-1.718	-2.921	-8.234
skull	9.037	9.399	-19.359	10.056	7.924	-24.045	7.682	8.670	4.513
wrist	6.458	6.333	-15.185	6.473	3.804	-26.189	0.391	4.192	-2.255
liver_t1	3.572	3.213	-9.793	2.082	0.854	-22.846	-0.071	1.676	-3.701
liver_t2e1	10.731	11.223	-19.875	7.391	7.716	-24.896	8.103	8.154	0.404
ped_chest	2.459	2.311	-16.683	4.377	0.942	-24.843	-2.752	0.726	-3.859
sag_head	8.856	8.808	-13.659	11.348	6.844	-16.354	7.506	8.481	6.899
Average	6.271	5.995	-14.492	5.838	4.194	-21.864	2.940	4.105	-0.694

From Table 3.16 it is possible to infer that all intra encoders, with the exception of MRP, show improvements on their compression efficiency, on average, which can be up to 6.3% for H.264, compared to the results of Table 3.12. However, some intra encoders, for some sequences, have losses in compression efficiency. Nonetheless, despite the prediction of

HEVC RExt being more efficient, there is an additional side information that needs to be transmitted, in order to reconstruct the prediction in the decoder. This can explain, in part, the loss in compression efficiency, on average, shown by MRP, as the side information is roughly 0.04 bpp, on average.

It is possible to observe that, the inter encoders show losses in compression efficiency, which can also be partially explained by the prediction overhead. But the loss in compression efficiency for these encoders, 14.5% for HEVC RExt and 21.9% for JP3D on average, is too high to be only explained by the extra prediction information. As can be seen in Figure 3.6, the HEVC RExt generated residue has more energy than the one originated by the pixel-wise difference predictor. The figure on the left has an entropy of 3.26 bits and the figure on the right has an entropy of 4.17 bits, which may justify the more efficient compression results for the images with lower entropy. Also, it can be observed from the figure on the left side that the HEVC RExt residue has a notorious block effect which can have an impact in the compression process. Analysing the results on Table 3.16, we can argue that this block effect has a negative effect for the inter-slice prediction, explaining the difference in results between intra and inter encoders.



(a) Pixel-wise difference predictor residue.

(b) HEVC RExt Random Access residue.

Figure 3.6: Example of the residue obtained for the skull sequence for slice 29.

3.4 Prediction on different slice planes

In Section 3.2, we argued that the optimal plane in which to form the slices for the compression could be different that the usual XY one. It was shown that both YZ and XZ planes present higher compression efficiency than sequences aligned with the XY plane. Also, in Section 3.3 it was demonstrated that applying a prior inter-slice predictor results in higher compression efficiency for the MRP encoder. Therefore, in this section both processing techniques are combined, and the results of their compression efficiency is assessed.

3.4.1 Pixel-wise difference calculated in different planes

In this first approach we will first align sequences in planes YZ and XZ, as in Section 3.2, and then apply to these sequences the pixel-wise difference predictor, as in Section 3.3. However, like in Section 3.3.2, the resulting image from the pixel-wise difference predictor has pixels that can not be represented by 8 bits. Therefore, Table 3.17 shows the number of overflow pixels, for each sequence and plane. Again, each overflow pixel will cost us eight bits to transmit the extra information, which may be considered negligible, which are taken into account in the following results. The results for this approach are shown in Tables 3.18 and 3.19, for the YZ and XZ planes, respectively.

Table 3.17: Number of lossy pixels when the pixel-wise difference is calculated in different planes.

Sequence	YZ Lossy pixels	XZ Lossy pixels
Aperts	0	0
carotid	0	1
skull	141	105
wrist	5	6
liver_t1	0	0
liver_t2e1	41	0
ped_chest	0	0
sag_head	0	0

Table 3.18: Coding performance for the pixel-wise difference applied on the YZ plane (results in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	0.914	0.887	0.823	1.060	1.053	1.497	0.869	0.872	0.713
carotid	1.608	1.580	1.506	1.678	1.637	1.935	1.399	1.417	1.194
skull	2.330	2.294	2.205	2.353	2.218	2.909	2.157	2.013	1.729
wrist	1.171	1.160	1.116	1.278	1.276	1.601	1.094	1.125	0.907
liver_t1	2.631	2.522	2.409	2.832	2.543	3.041	2.323	2.361	2.031
liver_t2e1	2.128	2.047	1.989	2.135	1.957	2.610	1.834	1.811	1.411
ped_chest	2.053	1.924	1.901	2.236	2.131	2.550	1.867	1.893	1.590
sag_head	2.407	2.327	2.288	2.706	2.540	2.757	2.260	2.288	1.946
Average	1.905	1.843	1.780	2.035	1.920	2.363	1.725	1.722	1.440

Table 3.19: Coding performance for the pixel-wise difference applied on the XZ plane (results in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	0.908	0.888	0.826	1.021	1.059	1.419	0.868	0.845	0.714
carotid	1.541	1.513	1.424	1.677	1.622	1.858	1.395	1.401	1.194
skull	2.252	2.215	2.127	2.323	2.201	2.794	2.127	1.976	1.715
wrist	1.325	1.321	1.268	1.299	1.267	1.879	1.129	1.124	0.921
liver_t1	2.641	2.537	2.395	2.826	2.563	3.015	2.321	2.355	2.064
liver_t2e1	1.973	1.903	1.695	2.107	1.919	2.213	1.795	1.752	1.527
ped_chest	1.930	1.796	1.748	2.185	2.122	2.348	1.820	1.872	1.591
sag_head	2.407	2.324	2.288	2.693	2.540	2.757	2.265	2.286	1.939
Average	1.872	1.812	1.721	2.016	1.912	2.285	1.715	1.701	1.458

Comparing Table 3.18 and Table 3.19, allows to infer that there is no clear preferred direction, as there are encoders that benefit more from the YZ plane and others that benefit more from the XZ plane. However, for MRP the best plane is the YZ one, as in Section 3.2.

Overall, when using this approach the encoder with the highest compression efficiency is MRP, with 1.44 bpp. From Tables 3.5 and 3.12, is possible to observe that these results are close to the ones showed in previous approaches, namely, a reduction of 0.11 bpp for the compression in YZ plane, and 0.13 bpp, for the pixel-wise difference predictor results.

3.4.2 Pixel-wise difference compression in different planes

In the second approach the pixel-wise difference predictor is applied to the sequences, as in Section 3.3, and then the resulting sequences are aligned and encoded along in the YZ and XZ planes, as in Section 3.2. The results for this approach are show in Tables 3.20 and 3.21, for the YZ and XZ planes, respectively.

Once again, it is possible to observe from these tables that there is not a global preferred direction, as different encoders have a better compression efficiency for different planes. This approach results in a higher compression efficiency than the previous one, as MRP have the best result with 1.36 bpp. In the previous approach, the best result was 1.44 bpp, also for MRP. Unlike in Section 3.2, the best plane for MRP is now the XZ one.

Analysing Tables 3.5 and 3.12, it is possible to observe that these results present a lower compression efficiency than the ones showed in previous approaches, namely in Section 3.3. Nevertheless, these results still have a higher compression efficiency than those of last section. Namely, these results have a reduction of 0.01 bpp and 0.05 bpp, when comparing

Table 3.20: Coding performance for pixel-wise difference in the YZ plane (results in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	0.901	0.878	0.706	1.077	1.127	1.689	0.910	0.948	0.742
carotid	1.523	1.504	1.300	1.702	1.606	1.678	1.429	1.476	1.223
skull	1.958	1.967	1.784	2.074	2.112	2.567	1.784	1.853	1.659
wrist	1.220	1.233	0.987	1.282	1.394	1.687	1.152	1.203	0.994
liver_t1	2.458	2.474	2.055	2.810	2.660	3.058	2.266	2.406	2.090
liver_t2e1	1.793	1.803	1.575	2.019	2.027	2.681	1.618	1.703	1.421
ped_chest	1.603	1.609	1.481	1.781	1.921	1.993	1.571	1.597	1.374
sag_head	2.063	2.074	1.860	2.357	2.349	2.793	1.878	2.094	1.816
Average	1.690	1.693	1.469	1.888	1.900	2.268	1.576	1.660	1.415

Table 3.21: Coding performance for the pixel-wise difference in the XZ plane (results in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	0.918	0.910	0.702	1.072	1.169	1.631	0.923	0.938	0.765
carotid	1.568	1.568	1.301	1.783	1.631	1.642	1.494	1.523	1.295
skull	1.978	1.986	1.772	2.118	2.156	2.524	1.814	1.870	1.704
wrist	1.094	1.106	0.990	1.145	1.232	1.781	1.052	1.009	0.860
liver_t1	2.443	2.459	2.048	2.801	2.647	3.017	2.248	2.358	1.670
liver_t2e1	1.850	1.863	1.503	2.091	2.089	2.373	1.688	1.785	1.295
ped_chest	1.673	1.666	1.475	1.876	2.006	1.957	1.646	1.726	1.489
sag_head	2.077	2.083	1.861	2.360	2.352	2.789	1.884	2.096	1.814
Average	1.700	1.705	1.457	1.906	1.910	2.214	1.594	1.663	1.362

with the results from YZ plane and for the pixel-wise difference predictor encoding, respectively.

3.5 Histogram packing

In [57], Armando Pinho shows that prediction and transform based encoders have a poor performance when compressing images with sparse histograms. Essentially, it is shown that when a histogram of an image is sparse, that is, it does not use all the possible values in its range, encoders have a poor performance. Medical images, especially with higher dynamic ranges, above 8 bits-per-pixel, are known to have sparse histograms, therefore this characteristic will be exploited in order to improve the compression efficiency of MRP. Several authors have proposed simple and more complex methods, applied both off-line and on-line, to perform the histogram packing, see for instance in [58–60].

In the histogram packing method implemented in this research a simple off-line method was used. The number of different values are determined in an image in order to establish the dynamic range of the new sequence. Then it is a simple matter of mapping the real values, by the same order, to a lookup table. For the transmission of the table, the number of different values are sent, as well as their respective intensity. This values are represented by b bits per value, where b represents the number of bits-per-pixel of the original image. This cost will be taken into account in the following results, presented in Tables 3.22 and Table 3.23.

Table 3.22: Results of the use of histogram packing before the encoding (in bpp).

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	1.224	1.166	0.758	1.208	1.289	0.966	1.088	1.028	0.803
carotid	2.049	1.986	1.414	1.968	2.010	1.538	1.770	1.679	1.374
skull	3.316	3.012	1.855	3.088	3.108	2.168	2.878	2.744	2.446
wrist	1.968	1.948	1.055	1.771	1.811	1.293	1.682	1.603	1.221
liver_t1	3.513	3.419	2.138	3.403	3.276	2.428	3.178	3.039	2.598
liver_t2e1	2.778	2.537	1.577	2.440	2.552	1.794	2.394	2.247	1.715
ped_chest	3.086	3.057	1.536	3.081	3.034	2.073	2.949	2.801	2.348
sag_head	2.657	2.617	1.789	2.828	2.926	2.208	2.603	2.540	2.301
Average	2.574	2.468	1.515	2.473	2.501	1.808	2.318	2.210	1.851

Table 3.23: Percentage of compression gain when using the histogram packing.

Sequence	H.264	Intra	R. A.	MMP	JPEG 2000	JP3D	JPEG-LS	CALIC	MRP
Aperts	-2.0	-2.1	-3.3	-2.0	-1.7	-2.3	-2.2	-2.4	-3.5
carotid	-1.6	-1.6	-2.3	-1.6	-1.4	-2.1	-1.8	-1.9	-2.0
skull	-1.0	-1.0	-1.8	-1.0	-0.9	-1.2	-1.1	-1.2	-1.4
wrist	-1.5	-1.5	-3.0	-1.6	-1.6	-2.5	-1.8	-1.9	-2.6
liver_t1	-0.6	-0.6	-1.0	-0.6	-0.6	-0.8	-0.6	-0.7	-1.2
liver_t2e1	-0.9	-1.0	-1.6	-1.0	-0.9	-1.3	-1.0	-1.1	-1.6
ped_chest	-0.7	-0.7	-1.4	-0.7	-0.6	-0.8	-0.7	-0.7	-0.9
sag_head	-0.7	-0.7	-1.0	-0.6	-0.6	-0.8	-0.7	-0.7	-0.9
Average	-1.0	-1.0	-1.7	-1.0	-0.9	-1.3	-1.1	-1.1	-1.5

As we can see from these tables, the histogram packing does not produce more efficient compression results for these sequences. The losses range from 1.7%, for HEVC RExt Random Access, to 0.9%, for JPEG2000. The MRP encoder has a loss in compression efficiency of 1.5%. With a better lookup table compression method, as shown in [60] for instance, some of these results could be improved and, at least, have lower compression efficiency losses.

These losses in efficiency can be explained by the fact that the used sequences do not show a significant difference between the expected dynamic range and the number of actual different values. In Table 3.24 the actual number of different values for the these sequences is shown.

Table 3.24: Average number of actual values present in the medical sequences.

Sequence	Number of values
Aperts	196
carotid	256
skull	255
wrist	241
liver_t1	164
liver_t2e1	200
ped_chest	166
sag_head	148

Table 3.25 shows the average of the number of different values present for medical images, with bit-depths ranging from 8 to 16 bits-per-pixel, available in [61]. A more detailed description of these images can be seen in Table B.1. In this table the types of images are as follow: Computed Radiography (CR), Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and US.

Table 3.25: Average number of values actually present in the medical images from [61].

Type	Average bit-depth	Percentage of used values
CR	12.5	57.4
CT	14.7	17.2
MRI	16	1.7
US	8	91.7

The values expressed in these tables show us that higher bit-depth images usually use a lower percentage of the values of the images original dynamic ranges. A lower percentage result in higher compression efficiency gains, as seen in [60]. Thus, the histogram packing will be applied to these images. However, the only used encoder that supports images up to 16 bits-per-pixel is JPEG-LS. Therefore, in Table 3.26 average results for the compression of the images available in [61] are shown using JPEG-LS, with and without the histogram packing. A more detailed version of these results can be seen in Table B.2.

From Table 3.26 we can see that, unlike for the sequences we have been using so far, for these medical images the compression efficiency of JPEG-LS is improved when histogram packing is used, by 33.4%, or 2.25 bpp, on average. Once again, images with lower differences between the expected range and the actual range present smaller gains or losses in compression efficiency, this can be seen, for instance, in MRI and US image

Table 3.26: Average compression results for images from [61], with and without histogram packing, using JPEG-LS (results in bpp).

Type	Original	Histogram Compression	Difference (in %)
CR	6.343	5.464	13.9
CT	7.838	4.696	40.1
MRI	10.009	5.078	49.3
US	2.748	2.714	1.2
Average	6.734	4.488	33.4

types. The MRI is the image type with the highest compression efficiency gain, 49.3%, and with the lowest dynamic range percentage, 1.7%. On the other hand, US is the image type with the lowest compression efficiency gain, 1.2%, and with the highest dynamic range percentage, 91.7%. The highest efficiency gain occurs for *ct_3030*, with 54.8% (6.3 bpp), as seen in Table B.2. For the US images, that have 8 bits-per-pixel, we can see that, as expected, the results vary between small losses and small gains in compression efficiency (in Table B.2). This is consistent with the results of Table 3.23.

An explanation for this improvement in efficiency is given in [62], where it is shown that the histogram packing lowers the total variation of an image. This means that the approximation error is cut down, thus providing higher compression efficiencies.

This is an interesting and simple approach, however due to the characteristics of MRP, that only is capable of encoding 8 bits-per-pixel images, we are not currently able of making use of it. A part of our intended future work will lead us to adding support to encode higher bit depth images to MRP, then we will try to use this technique.

3.6 Techniques comparison

In this section we will compare the efficiency gains given by the previously described techniques. We will now focus mainly on MRP and on HEVC RExt, with the Random Access profile, as these are the encoders with the highest compression efficiencies. Tables 3.27 and 3.28 show these comparisons, for HEVC RExt Random Access and for MRP, respectively.

As can be seen from both tables, the technique that shows the best compression efficiency is the pixel-wise difference predictor. As mentioned in Section 3.3, this fact is due to the anatomically or physiologically correlation between slices [30].

If we focus on Table 3.27, we see that, apart from the pixel-wise difference predictor, all the other techniques result in losses in compression efficiency. For slices aligned in different

Table 3.27: Comparison of the techniques applied to HEVC RExt Random Access (results in bpp).

Sequence	Original	YZ Slices	OCP Algorithm	Pixel-wise Difference	Histogram Packing
Aperts	0.734	0.771	0.757	0.679	0.758
carotid	1.383	1.424	1.391	1.243	1.414
skull	1.822	2.038	2.038	1.488	1.855
wrist	1.024	1.136	1.209	0.879	1.055
liver_t1	2.118	2.317	2.335	1.897	2.138
liver_t2e1	1.552	1.889	1.737	1.245	1.577
ped_chest	1.515	1.713	1.713	1.292	1.536
sag_head	1.771	2.033	2.036	1.538	1.789
Average	1.490	1.665	1.652	1.282	1.515

Table 3.28: Comparison of the techniques applied to MRP (results in bpp).

Sequence	Original	Concatenation	YZ Slices	OCP Algorithm	Pixel-wise Difference	Histogram Packing
Aperts	0.776	0.736	0.676	0.691	0.630	0.803
carotid	1.347	1.276	1.150	1.231	1.115	1.374
skull	2.411	2.355	1.599	1.599	1.696	2.446
wrist	1.190	1.139	0.962	0.821	0.862	1.221
liver_t1	2.566	2.496	2.013	2.052	1.790	2.598
liver_t2e1	1.688	1.571	1.393	1.642	1.307	1.715
ped_chest	2.327	2.264	1.322	1.322	1.324	2.348
sag_head	2.281	2.243	1.687	1.682	1.789	2.301
Average	1.823	1.760	1.350	1.380	1.314	1.851

planes, columns three and four, this is explained by the fact that we are essentially moving some of the inter-slice redundancy to intra-slice redundancy. As HEVC RExt relies its gains on inter-slice prediction, this means that this prediction will be less efficient.

If we focus on Table 3.28, we see that, apart from the histogram packing, all the methods results in an increase in compression efficiency for MRP, which is also true for the remaining intra encoders. Considering the histogram packing, as we argued in Section 3.5, these results could still be improved if the lookup table transmission method were improved.

Comparing both tables, we can see that the highest compression efficiency is still obtained for HEVC RExt, with the Random Access profile, with 1.28 bpp, on average. Despite this, MRP now have a compression efficiency closer to that of HEVC RExt, with 1.31 bpp, corresponding to a difference in just 0.03 bpp, or 2.3%, on average.

3.7 Summary

The presented techniques were able to achieve the goal of improving the compression efficiency of MRP. Initially, MRP was encoding the sequences with an average of 1.82 bpp (see Table 3.2). The different proposed techniques exploit different characteristics of this encoder. For instance, the slice concatenation exploited the fact that MRP optimises its predictors to the input image. Another example is the formation of slices in different planes, that exploits the highly efficient intra prediction of MRP by taking advantage of the inter-slice redundancy on the usual XY plane, on the individual slices, on different planes.

Nevertheless, the most efficient technique was the pixel-wise difference predictor. Using this method we were able to achieve an average compression of 1.31 bpp, which is equivalent to a compression efficiency gain of 27.9% (see Tables 3.12 and 3.13).

The best performance, however, was still obtained by HEVC RExt, using the Random Access profile, when encoding the pixel-wise difference residue, with 1.28 bpp, which is equivalent to a gain in compression efficiency of 13.9% (see Tables 3.12 and 3.13).

Several techniques were studied, but it was possible to conclude that the use of inter-slice prediction was the technique with the higher compression efficiency gains. If we consider the use of this technique on MRP, it becomes quite clear that MRP is a very efficient encoder, that with the appropriate techniques can match or even outperform HEVC RExt. With these conclusions in mind, the next chapter of this work will focus on improving the MRP encoder. Namely we will add the ability to perform inter-prediction to MRP, keeping in mind the improvements added by some of the other techniques.

Chapter 4

Proposed Methods in Minimum Rate Predictors

In this chapter the proposed contributions to improve the MRP algorithm will be described and their compression efficiency analysed. These contributions are mainly related to the inter-slice prediction support in MRP. The chapter will be concluded with a summary of the proposed improvements and a discussion of the experimental results.

4.1 Context calculation

The original MRP algorithm uses 12 pixels to determine the context for the arithmetic coding of the residue, which is calculated for each pixel using Equation 2.4.

However, as we intend to add inter-slice prediction support to MRP some problems might arise with the original context calculation. One of these issues is the distance between pixels in different slices, as we will use another slice as reference for the inter-slice prediction. Thus, the context calculation must be different. In this sense, it was decided to include all the reference pixels used in the prediction, as shown in Equation 4.1. For the case of the prediction using one reference slice. Nevertheless, this can easily be extended to an arbitrary number of slices.

$$U = \sum_{k=1}^{K_1} \frac{1}{\delta_{1_k}} \left| s(k) - \hat{s}(k) \right| + \sum_{k=1}^{K_2} \frac{1}{\delta_{2_k}} \left| s_p(k) - \hat{s}_p(k) \right|, \tag{4.1}$$

In this equation, K_1 is the number of reference pixels in the current slice and K_2 is the number of reference pixels in the reference slice. The δ_k factor was changed accordingly, as seen in Equation 4.2, for the reference pixels in the current pixel slice, and in Equation 4.3 for the reference pixels in the reference slice.

$$\delta_{1_k} = \frac{64}{\sqrt{d_x(k)^2 + d_y(k)^2}} \tag{4.2}$$

$$\delta_{2_k} = \frac{64}{\sqrt{d_x(k)^2 + d_y(k)^2 + D^2}} \tag{4.3}$$

The distance between slices is represented by D. However, this variable will be further discussed and its influence will be analysed in the following sections.

4.1.1 Experimental results

In order to assess the impact of these changes, the new context calculation was implemented on the MRP algorithm. The results of the influence of the new context calculation are shown in Table 4.1.

Table 4.1: Comparison between the original MRP algorithm context and the proposed context calculation (results in bpp).

Sequence	Original context	New Context	Difference (result in %)
Aperts	0.775	0.777	-0.26
carotid	1.374	1.376	-0.15
skull	2.329	2.331	-0.09
wrist	1.173	1.174	-0.09
liver_t1	2.582	2.581	0.05
liver_t2e1	1.722	1.727	-0.29
ped_chest	2.337	2.339	-0.09
sag_head	2.279	2.282	-0.13
Average	1.821	1.823	-0.13

From the results of this table, it is possible to infer that the effect of the new context calculation on the compression efficiency is minimal. Actually, there is a slight loss in compression efficiency, 0.13% on average, which can be considered negligible. In the following sections the proposed context will be combined with all proposed contributions to the MRP algorithm.

4.2 Pixel-wise difference predictor

This contribution is related to the tests performed in Section 3.3, namely the pixel-wise difference predictor. This predictor was implemented in MRP codec, allowing the use of this prediction mode, before the compression stage on MRP.

As discussed before, the difference between pixels in a sequence can take values in the range $\left]-2^b : 2^b\right[$, where b represents the original image bit depth. This means the residue can have up to $b+1$ bits-per-pixel. In an 8 bit-per-pixel sequence, as the ones used in this research, the residue samples are in the range $]-256 : 256[$.

The MRP codec however only allows the encoding of 8 bits-per-pixel images. If the residue exceeds the range of values allowed in the 8 bits-per-pixel range, extra information must be used. It was decided to transmit this extra information, because this situation rarely occurs, not compressed.

The pixel-wise difference is calculated by subtracting two co-located pixels in adjacent slices, as shown in Figure 3.5. If this difference is represented by 9 bits, the result is truncated to 8 bits, and then it is normalized to be in the range $[0 : 255]$. The extra information is calculated and transmitted without compression, as an 8 bit number.

Transmitting extra information without compression does not have much influence in the coding efficiency, because the amount of pixels represented by 9 bits is negligible, when compared to the total number of pixels. Table 4.2 presents the cost of the extra information for several sequences.

Table 4.2: Number of lossy pixels in a sequence when using the pixel-wise difference predictor and its transmission cost.

Sequence	Width	Height	Frames	Lossy pixels	Cost (bpp)
skull	256	256	203	0	0.00
sag_head	256	256	58	0	0.00
carphone	352	288	382	320	6.60×10^{-5}
tempete	352	288	260	239	7.30×10^{-5}
MOBILE	352	288	300	299	7.90×10^{-5}
PeopleOnStreet	2560	1600	150	149	2.00×10^{-6}
Tennis	1920	1080	240	232	4.00×10^{-6}
ChinaSpeed	1024	768	500	499	1.00×10^{-5}

Performance evaluation of this prediction mode in combination with the succeeding contributions made to MRP will be described in the following sections.

4.3 Inter slice prediction

A sequence of frames, slices in our application, such as a medical image sequence, certainly presents high levels of inter-slice redundancy. Since the early days of video compression, with H.261 [63], video encoders have been exploiting inter-slice redundancy, which is one of the main sources of their coding efficiency. State-of-the-art encoders, such as H.264 [19] and HEVC, still heavily rely on this characteristic [22].

This way the use of inter-slice redundancy was an obvious characteristic to be exploited by MRP. Inter-slice prediction support was developed for MRP algorithm, while maintaining its coding structure. Inter-slice redundancy can be implemented in mainly, two ways:

- Uni-directional inter slice prediction;
- Bi-directional inter slice prediction, such as the ones used by H.264 and HEVC.

The main difference between these two modes of inter-slice prediction is that the bi-directional prediction can handle, in a more efficient way, the uncovered regions of an image. Both of these approaches will be implemented in MRP algorithm and described in the following sections.

4.3.1 Uni-directional prediction

Initially the ability of using the previous slice, relative to the slice to encode, as reference was added to MRP. In order to achieve this goal, the original prediction scheme was extended to include K_2 pixels from the previous slice, when available. Thus, Equations 2.3 and 2.6 remain approximately the same.

Figure 4.1 shows the disposition of the reference pixels on the previous slice. Pixel p_0 in Figure 2.20 is co-located with the pixel p_1 in Figure 4.1, in adjacent slices.

If we consider K_1 to be the number of references used for the intra slice prediction and K_2 the number of references used for the inter slice prediction, the result of the prediction from the *m-th* predictor is given bt Equation 4.4,

$$\hat{s}(0) = \sum_{k=1}^{K_1} a_{1m}(k) \cdot s(k) + \sum_{k=1}^{K_2} a_{2m}(k) \cdot s_p(k), \tag{4.4}$$

where $a_{1m}(k)$ is the prediction coefficient for current slice reference pixel, $a_{2m}(k)$ is the prediction coefficient for the previous slice reference pixel, and $s_p(k)$ represents the k

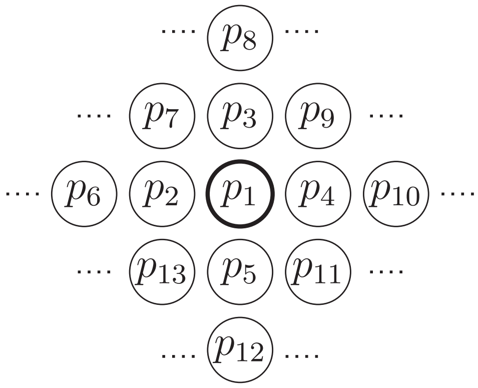


Figure 4.1: Reference pixels in the previous slice.

reference pixel in the reference slice. The context for the residue encoding is calculated using Equation 4.1, as previously described.

4.3.2 Bi-directional prediction

Bi-directional inter-slice prediction is used, nowadays, by state-of-the-art encoders [20,22]. In this type of prediction both a previous and a following slice, not necessarily an adjacent one, according to the viewing order, can be used for the inter-slice prediction.

This, however, requires that the coding order of the sequence must be different than the display order, because it requires the prior decoding of reference frames to be decoded. Typically, these types of slices require less bits to be encoded than I-type and P-type slices.

Older video coding standards, such as MPEG-2 Video, also known as H.262, from the Motion Picture Experts Group (MPEG) group [64], do not use B slices as references for the prediction. Even so, in the recent encoding standards, such as H.264 and HEVC, B-type slices can be used as references. Both of these cases will be discussed and tested, we will refer to them as MPEG-2 B-type slices and HEVC B-types slices.

MPEG-2 B-type slices

MPEG-2 uses three types of slices, Intra (I), forward prediction slices (P) and bi-directional prediction slices (B). The bidirectional prediction uses fixed references, meaning that B-type slices cannot be used as references. This results on the prediction scheme showed in Figure 4.2, for the case of two MPEG-2 B-type slices between references.

It can be seen from this figure that, when we use a higher number of MPEG-2 B slices between the reference slices, some of the B-type slices will be further away from their

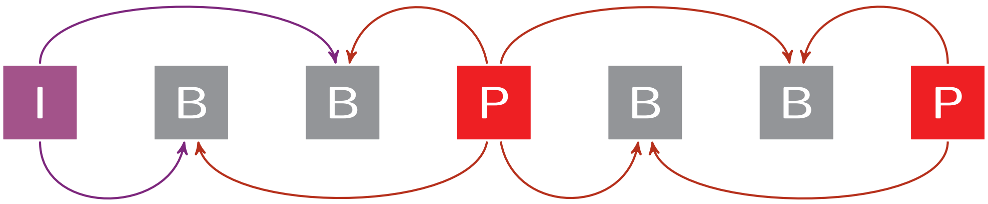


Figure 4.2: Usual bidirectional prediction dependencies

references. This can result in lower quality references for the inter-slice prediction.

HEVC B-type slices

The state-of-art HEVC standard uses a different kind of bidirectional prediction than MPEG-2. HEVC B-type slices can be used as a reference for the prediction of other slices of the same type. A schematic example of this type of prediction is shown in Figure 4.3.

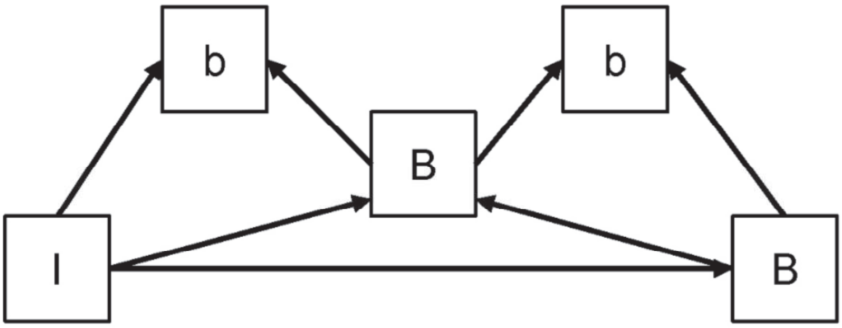


Figure 4.3: HEVC bidirectional prediction scheme [22].

It is possible to see that in this case, references for HEVC B-type slices can be closer, for the same number of B slices between references, than in the MPEG-2 case, which may result in a better prediction efficiency.

As mentioned before, both of these approaches have been tested on MRP. Thus, it was possible to assess if the current state-of-the-art bi-directional prediction, the one used by H.264 and HEVC, also produces a more efficient compression results for MRP, than the bi-directional prediction used by MPEG-2.

The bidirectional prediction scheme of MRP algorithm will be extended as described in Section 4.3.1 for the two reference slices case. In both used reference slices, the reference will be positioned around the co-located pixel, as in Figure 4.1.

In this case we have three slices to consider for the prediction. Hence, we have K_3 reference pixels used for the intra prediction, K_4 reference pixels used for one of the reference slices and K_5 for the other (in our experiments K_4 and K_5 are the same). Considering this, the result of the prediction from the *m-th* predictor in a HEVC type B slice is given by Equation 4.5.

$$\hat{s}(0) = \sum_{k=1}^{K_3} a_{3m}(k) \cdot s(k) + \sum_{k=1}^{K_4} a_{4m}(k) \cdot s_{b_1}(k) + \sum_{k=1}^{K_5} a_{5m}(k) \cdot s_{b_2}(k), \tag{4.5}$$

where $a_{3m}(k)$ is the prediction coefficient for a given same slice reference pixel, $a_{4m}(k)$ and $a_{5m}(k)$ are the prediction coefficients for the reference pixels in each of the reference slices, and $s_{b_1}(k)$ and $s_{b_2}(k)$ are the k reference pixel in each of the reference slices. The context calculation is also extended accordingly, as seen in Equation 4.6,

$$U = \sum_{k=1}^{K_3} \frac{1}{\delta_{3_k}} \left| s(k) - \hat{s}(k) \right| + \sum_{k=1}^{K_4} \frac{1}{\delta_{4_k}} \left| s_{b_1}(k) - \hat{s}_{b_1}(k) \right| + \sum_{k=1}^{K_5} \frac{1}{\delta_{5_k}} \left| s_{b_2}(k) - \hat{s}_{b_2}(k) \right|, \tag{4.6}$$

where $s(k)$ is the real value of the reference pixel and $\hat{s}(k)$ is the prediction obtained for the same reference pixel. δ_{3_k} is calculated in a similar way as in Equation 4.2, while δ_{4_k} and δ_{5_k} are calculated as in Equation 4.3. As stated in Section 4.1, U is used to determine the context modelling for the arithmetic encoder.

4.3.3 Experimental Results

We will now proceed with an analysis of the compression efficiency added by the introduction of the inter-slice prediction support to MRP. For easier consultation the number of reference pixels are as follow:

- K_0 : number of intra reference pixels used in the I-type slices;
- K_1 : number of intra reference pixels used in the P-type slices;
- K_2 : number of inter reference pixels used in the P-type slices;
- K_3 : number of intra reference pixels used in the B-type slices;
- K_4, K_5 : number of inter reference pixels, for the two reference slices, used in the B-type slices.

The actual values of the references used will be determined in the following sections. In the following experiments the distance between slices, D, is assumed to be 1, later on this value will be further discussed, and K_0 is 30, as calculated by the original MRP algorithm, using the spatial dimensions of the input sequences.

Uni-directional prediction results

The first step taken in the experimental procedure, to analyse the uni-directional prediction influence, was to determine the optimal combination of the parameters K_1 and K_2. An exhaustive set of tests were performed to determine the combination that resulted in the most efficient compression rate for each sequence. Table 4.3 shows the results of this optimisation for each sequence.

Table 4.3: Optimisation of the K_1 and K_2 parameters for each sequence.

Sequence	K_1	K_2	Best Result (in bpp)
Aperts	12	13	0.539
carotid	20	25	1.049
skull	12	25	1.529
wrist	12	25	0.728
liver_t1	12	25	1.657
liver_t2e1	20	25	1.212
ped_chest	12	13	1.227
sag_head	6	13	1.576

The optimal set of parameters K_1 and K_2 obtained for each sequence were then applied to the other sequences. This way we intend to determine the optimal K_1 and K_2 pair, on average, for all the sequences. These results are shown in Table 4.4.

Table 4.4: Compression results of the optimised K_1 and K_2 parameters (in bpp).

Sequence	$K_1 = 6$ $K_2 = 13$	$K_1 = 12$ $K_2 = 13$	$K_1 = 12$ $K_2 = 25$	$K_1 = 20$ $K_2 = 25$
Aperts	0.539	0.539	0.539	0.544
carotid	1.087	1.059	1.055	1.049
skull	1.536	1.534	1.529	1.534
wrist	0.746	0.736	0.728	0.729
liver_t1	1.672	1.664	1.657	1.657
liver_t2e1	1.288	1.239	1.221	1.212
ped_chest	1.231	1.227	1.230	1.237
sag_head	1.576	1.577	1.582	1.586
Average	1.209	1.197	1.193	1.194

As can be inferred from this table, the optimal pair of K_1 and K_2 for all the sequences is $K_1 = 12$ and $K_2 = 25$, with an average result of 1.193 bpp. The same experimental procedure was repeated using the pixel-wise difference predictor, as can be seen in Tables 4.5 and 4.6.

Table 4.5: Optimisation of the K_1 and K_2 parameters for each sequence, using pixel-wise difference predictor.

Sequence	K_1	K_2	Best Result (in bpp)
Aperts	12	13	0.530
carotid	20	13	1.038
skull	12	13	1.251
wrist	12	25	0.641
liver_t1	20	25	1.479
liver_t2e1	30	25	0.926
ped_chest	12	13	1.095
sag_head	12	13	1.362

Table 4.6: Compression results of the optimised K_1 and K_2 parameters, using pixel-wise difference predictor (in bpp).

Sequence	$K_1 = 12$	$K_1 = 12$	$K_1 = 20$	$K_1 = 20$	$K_1 = 30$
	$K_2 = 13$	$K_2 = 25$	$K_2 = 13$	$K_2 = 25$	$K_2 = 25$
Aperts	0.530	0.532	0.532	0.535	0.538
carotid	1.043	1.044	1.038	1.040	1.042
skull	1.251	1.253	1.253	1.256	1.262
wrist	0.645	0.641	0.645	0.642	0.645
liver_t1	1.480	1.479	1.480	1.479	1.484
liver_t2e1	0.931	0.930	0.929	0.928	0.926
ped_chest	1.095	1.099	1.097	1.102	1.109
sag_head	1.362	1.366	1.365	1.369	1.375
Average	1.042	1.043	1.042	1.044	1.048

Table 4.6 shows that there are two pairs of K_1 and K_2 that, on average, have the best compression efficiency, $K_1 = 12$, $K_2 = 13$ and $K_1 = 20$, $K_2 = 13$, with an average result of 1.042 bpp.

It can be noticed that the results in each column of the tables are usually separated by less that 0.01 bpp. Thus, if we slightly change the prediction order, the overall compression efficiency will remain mostly unaffected. This way it was decided to keep the same parameters, with or without the pixel-wise difference predictor, hence $K_1 = 12$ and $K_2 = 25$ were chosen to be used from here on, for both cases.

Bi-directional prediction results

The same optimisations, as before, were performed for the bi-directional prediction. In the case of the bi-directional prediction, however, the number of parameters to optimise is higher, namely K_1, K_2, K_3, K_4 and B, the number of B slices to use. This would result in a very high number of possible combinations, so we will consider that K_1 and K_2 are not correlated with K_3 and K_4. This is true if we choose a pair of K_3 and K_4 that does not affect much the compression efficiency, meaning that B-type slices compression should not have a poor performance, that is, when we are using higher values of B.

This way, in an initial test we will optimise the values of K_1 , K_2 and B, using the values obtained for the type P frames in uni-directional prediction for K_3 and K_4, $K_3 = 12$ and $K_4 = K_5 = 25$, for both types of B type slices. The results of these optimisations are shown in Tables 4.7 and 4.8.

Table 4.7: Optimisation of the K_1, K_2 and B parameters for each sequence, using MPEG-2 B-type slices.

Sequence	K_1	K_2	B	Best result (in bpp)
Aperts	20	13	1	0.508
carotid	30	13	1	1.005
skull	12	25	1	1.366
wrist	30	25	2	0.644
liver_t1	20	25	1	1.526
liver_t2e1	20	13	1	1.046
ped_chest	20	13	2	1.099
sag_head	12	13	1	1.391

Table 4.8: Optimisation of the K_1, K_2 and B parameters for each sequence, using HEVC B-type slices.

Sequence	K_1	K_2	B	Best result (in bpp)
Aperts	20	5	5	0.500
carotid	30	13	4	1.006
skull	20	5	6	1.308
wrist	30	5	5	0.594
liver_t1	20	5	7	1.464
liver_t2e1	20	25	7	1.002
ped_chest	20	5	8	1.009
sag_head	12	13	9	1.339

We can clearly notice from these tables that, when the HEVC type bi-directional prediction is used there are more B slices between references. This is expected as B-type slices can be used as references and, this way, the reference slices will be closer to any given B slice.

Then the best values for each sequence were applied to other sequences, as before, to determine the best possible combination. These results are shown in Tables 4.9 and 4.10.

Table 4.9: Compression results of the optimization of the K_1, K_2 and B parameters for the MPEG-2 B-type slices (in bpp).

Sequence	$K_1 = 12$ $K_2 = 13$ $B = 1$	$K_1 = 12$ $K_2 = 25$ $B = 1$	$K_1 = 20$ $K_2 = 13$ $B = 1$	$K_1 = 20$ $K_2 = 13$ $B = 2$	$K_1 = 20$ $K_2 = 25$ $B = 1$	$K_1 = 30$ $K_2 = 13$ $B = 1$	$K_1 = 30$ $K_2 = 25$ $B = 2$
Aperts	0.508	0.509	0.508	0.530	0.510	0.511	0.535
carotid	1.024	1.022	1.009	1.072	1.010	1.005	1.071
skull	1.372	1.366	1.371	1.418	1.366	1.371	1.417
wrist	0.656	0.649	0.649	0.647	0.646	0.650	0.644
liver_t1	1.541	1.530	1.527	1.584	1.526	1.527	1.580
liver_t2e1	1.079	1.063	1.046	1.093	1.055	1.053	1.094
ped_chest	1.121	1.119	1.121	1.099	1.122	1.124	1.104
sag_head	1.391	1.393	1.392	1.695	1.392	1.393	1.697
Average	1.086	1.081	1.078	1.142	1.079	1.079	1.143

Table 4.10: Compression results of the optimization of the K_1, K_2 and B parameters for HEVC B-type slices (in bpp).

Sequence	$K_1 = 12$ $K_2 = 13$ $B = 9$	$K_1 = 20$ $K_2 = 5$ $B = 5$	$K_1 = 20$ $K_2 = 5$ $B = 6$	$K_1 = 20$ $K_2 = 5$ $B = 7$	$K_1 = 20$ $K_2 = 5$ $B = 8$	$K_1 = 20$ $K_2 = 25$ $B = 7$	$K_1 = 30$ $K_2 = 5$ $B = 5$	$K_1 = 30$ $K_2 = 13$ $B = 4$
Aperts	0.509	0.500	0.504	0.504	0.502	0.504	0.502	0.508
carotid	1.019	1.016	1.010	1.019	1.016	1.019	1.014	1.006
skull	1.357	1.320	1.308	1.320	1.320	1.321	1.318	1.353
wrist	0.607	0.598	0.599	0.599	0.599	0.600	0.594	0.602
liver_t1	1.512	1.496	1.505	1.464	1.495	1.467	1.492	1.498
liver_t2e1	1.043	1.023	1.036	1.004	1.039	1.002	1.028	1.018
ped_chest	1.018	1.010	1.022	1.020	1.009	1.020	1.010	1.037
sag_head	1.339	1.501	1.360	1.345	1.408	1.345	1.502	1.365
Average	1.050	1.058	1.043	1.034	1.049	1.035	1.058	1.048

From these tables we can conclude that the optimal combination, in terms of compression efficiency, using the MPEG-2 B-type slices, is $K_1 = 20$, $K_2 = 13$ and $B = 1$, which gives an average of 1.078 bpp. The optimal combination for the compression efficiency for the HEVC type B slices is $K_1 = 20$, $K_2 = 5$ and $B = 7$, with an average of 1.034 bpp for the compression.

Comparing both results, as expected for a more recent technique, we can see that the bidirectional prediction using HEVC B-type slices results in a more efficient compression.

The two different types of B slices have a difference in compression efficiency of roughly 0.05 bpp on average. We can also see that K_2 is much lower for the case of HEVC B-type slices, which is a consequence of the distance between P-type slices that is higher for the HEVC type bi-directional prediction, meaning that the prediction will rely more on its intra reference pixels.

The optimisation of the remaining parameters, K_3 and K_4, is shown in Tables 4.11 and 4.12, for MPEG-2 B-type slices and HEVC B-type slices, respectively. Like before, the best values for each sequence were then applied to other sequences, to determine the best average combination. These results are shown in Tables 4.13 and 4.14.

Table 4.11: Optimisation of the K_3 and K_4 parameters for each sequence, using MPEG-2 B-type slices.

Sequence	K_3	K_4	Best result (in bpp)
Aperts	6	13	0.498
carotid	12	13	1.000
skull	6	5	1.348
wrist	6	13	0.641
liver_t1	6	13	1.511
liver_t2e1	12	13	1.035
ped_chest	2	5	1.093
sag_head	6	13	1.382

Table 4.12: Optimisation of the K_3 and K_4 parameters for each sequence, using HEVC B-type slices.

Sequence	K_3	K_4	Best result (in bpp)
Aperts	12	13	0.491
carotid	20	13	1.000
skull	12	13	1.298
wrist	12	13	0.587
liver_t1	12	13	1.443
liver_t2e1	20	13	0.977
ped_chest	6	5	0.990
sag_head	6	13	1.331

Tables 4.13 and 4.14 show that the best results are obtained with $K_0 = 30$, $K_1 = 20$, $K_2 = 13$, $K_3 = 6$, $K_4 = K_5 = 13$ and $B = 1$ for MPEG-2 type bi-directional prediction, and with $K_0 = 30$, $K_1 = 20$, $K_2 = 5$, $K_3 = 12$, $K_4 = K_5 = 13$ and $B = 7$ for HEVC type bi-directional prediction. As seen before, small changes in the prediction order, i.e. the number of reference pixels, do not result in a high loss of compression efficiency, the results in each column of the tables differ at most 0.02 bpp, for each table. The difference

Table 4.13: Compression results of the optimization of the K_3 and K_4 parameters for the MPEG-2 B-type slices (in bpp).

Sequence	$K_3 = 2$ $K_4 = 5$	$K_3 = 6$ $K_4 = 5$	$K_3 = 6$ $K_4 = 13$	$K_3 = 12$ $K_4 = 13$
Aperts	0.501	0.500	0.498	0.500
carotid	1.014	1.012	1.002	1.000
skull	1.349	1.348	1.348	1.351
wrist	0.650	0.647	0.641	0.641
liver_t1	1.515	1.515	1.511	1.514
liver_t2e1	1.045	1.043	1.040	1.035
ped_chest	1.093	1.094	1.100	1.104
sag_head	1.383	1.384	1.382	1.383
Average	1.069	1.068	1.065	1.066

Table 4.14: Compression results of the optimization of the K_3 and K_4 parameters for HEVC B-type slices (in bpp).

Sequence	$K_3 = 6$ $K_4 = 5$	$K_3 = 6$ $K_4 = 13$	$K_3 = 12$ $K_4 = 13$	$K_3 = 20$ $K_4 = 13$
Aperts	0.495	0.492	0.491	0.494
carotid	1.050	1.026	1.008	1.000
skull	1.314	1.298	1.298	1.301
wrist	0.616	0.593	0.587	0.589
liver_t1	1.467	1.449	1.443	1.446
liver_t2e1	1.045	1.019	0.984	0.977
ped_chest	0.990	0.992	0.994	0.999
sag_head	1.335	1.331	1.332	1.334
Average	1.039	1.025	1.017	1.018

in the compression efficiency between both types of bi-directional prediction is also small, roughly 0.05 bpp.

In Section 4.3.1 we have seen that the compression efficiency using pixel-wise difference predictor is not much affected by changing the number of reference pixels. Therefore, for the bi-directional prediction, the order of the prediction will remain the same, with and without the use of pixel-wise difference predictor.

4.3.4 Distance between slices

To finalise the inter-slice prediction experimental evaluation it is still needed to assess the influence of the inter-slice distance parameter, D. The distance between slices, D, will be tested for both uni-directional and bi-directional prediction, using the optimal parameters previously obtained.

Distance between slices in uni-directional prediction

The results of this optimisation are shown in Tables 4.15 and 4.16, without and with the use of the pixel-wise difference predictor, respectively. The parameters used for the encoding process were $K_1 = 12$ and $K_2 = 25$, as stated in Section 4.3.3.

Table 4.15: Optimization of the parameter D for uni-directional prediction (results in bpp).

Sequence	D = 0.1	D = 0.5	D = 1	D = 2	D = 5	D = 10
Aperts	0.539	0.539	0.539	0.540	0.541	0.542
carotid	1.055	1.055	1.055	1.057	1.057	1.058
skull	1.529	1.529	1.529	1.531	1.533	1.534
wrist	0.728	0.728	0.728	0.729	0.730	0.731
liver_t1	1.657	1.657	1.657	1.658	1.662	1.664
liver_t2e1	1.221	1.221	1.221	1.223	1.226	1.224
ped_chest	1.230	1.230	1.230	1.231	1.231	1.233
sag_head	1.582	1.582	1.582	1.585	1.588	1.590
Average	1.193	1.193	1.193	1.194	1.196	1.197

Table 4.16: Optimization of the parameter D for uni-directional prediction, using pixel-wise difference predictor (results in bpp).

Sequence	D = 0.1	D = 0.5	D = 1	D = 2	D = 5	D = 10
Aperts	0.532	0.532	0.532	0.533	0.533	0.534
carotid	1.044	1.044	1.044	1.043	1.043	1.043
skull	1.253	1.253	1.253	1.253	1.254	1.254
wrist	0.641	0.641	0.641	0.641	0.642	0.642
liver_t1	1.479	1.479	1.479	1.479	1.480	1.480
liver_t2e1	0.930	0.930	0.930	0.931	0.930	0.930
ped_chest	1.099	1.099	1.099	1.099	1.099	1.099
sag_head	1.366	1.366	1.366	1.367	1.368	1.369
Average	1.043	1.043	1.043	1.043	1.044	1.044

Different values of D were tested, namely 0.1, 0.5, 1, 2, 5, 10, as can be seen in the tables. Thus we can see that, the higher the value of D the less efficient the compression becomes, in both tables. It also can be noticed that for values of D equal or less than 1, the results remain the same, in both tables. Thus, the inter-slice distance parameter for the uni-directional prediction was set to $D = 1$. Once again, it can be observed that changes in this parameter results in variations of the compression efficiency lower than 0.005 bpp, so there is no need to rigidly set this parameter.

Distance between slices in bi-directional prediction

In a similar way to that described in the previous section, the inter-slice parameter was tested for the bi-directional prediction. However, there is a slight difference in this case, due to the fact that, if the distance between reference slices is not the same for all slices, a problem arises. In fact, although for P-type slices the distance to their reference slices is the same, except the last one that can have a different distance. This notwithstanding is not the case for the B-type slices. Each of the B-type slices, between P-type slices, will have different distances to their reference slices.

In preliminary tests we used the same distance for all types of slices, as before. The results of these tests are shown in Tables 4.17 and 4.18. In these tables the number of reference pixels used are the ones determined in Section 4.3.3 for each case.

Table 4.17: Optimization of the parameter *D* for MPEG-2 bi-directional prediction (results in bpp).

Sequence	D = 0.1	D = 0.5	D = 1	D = 2	D = 5	D = 10
Aperts	0.498	0.498	0.498	0.499	0.500	0.500
carotid	1.002	1.002	1.002	1.002	1.001	1.002
skull	1.348	1.348	1.348	1.347	1.349	1.350
wrist	0.641	0.641	0.641	0.642	0.643	0.643
liver_t1	1.511	1.511	1.511	1.513	1.514	1.516
liver_t2e1	1.040	1.040	1.040	1.038	1.039	1.038
ped_chest	1.100	1.100	1.104	1.100	1.102	1.103
sag_head	1.382	1.382	1.382	1.383	1.384	1.385
Average	1.065	1.065	1.066	1.065	1.066	1.067

Table 4.18: Optimization of the parameter *D* for HEVC bi-directional prediction (results in bpp).

Sequence	D = 0.1	D = 0.5	D = 1	D = 2	D = 5	D = 10
Aperts	0.491	0.491	0.491	0.491	0.490	0.491
carotid	1.008	1.008	1.008	1.008	1.006	1.008
skull	1.298	1.298	1.298	1.297	1.296	1.295
wrist	0.587	0.587	0.587	0.587	0.588	0.588
liver_t1	1.443	1.443	1.443	1.444	1.444	1.445
liver_t2e1	0.984	0.984	0.984	0.984	0.985	0.983
ped_chest	0.994	0.994	0.994	0.995	0.994	0.998
sag_head	1.332	1.332	1.332	1.331	1.333	1.333
Average	1.017	1.017	1.017	1.017	1.017	1.018

The preliminary results show, once again, that the distance parameter, *D*, does not have much influence in the compression efficiency, with the results in the tables columns changing less than 0.002 bpp, on average for each table. It is also possible to conclude that in the bi-directional prediction, in opposition of what happens in uni-directional prediction, higher values of D present the most efficient compression. This is more noticeable in the HEVC type bi-directional prediction, due to the greater distance between B-type slices and their references.

As the results show that the distance parameter does not have much influence in the compression efficiency the value $D = 1$ was set. As stated before, the same distance, $D = 1$ will be used with the pixel-wise difference predictor.

4.4 Motion compensation

In most video encoders, such as H.264 [20] and HEVC, [22], inter-slice prediction mainly relies on motion estimation and compensation. There are several techniques to compensate the motion between slices. However, the most common is the motion estimation that uses vectors to represent the motion of blocks. Figure 4.4, represents two slices of a sequence where the right slice is to be encoded and left slice is the reference. The best match for the blocks on the right side slice are searched in the left slice. In order to perform the inter-slice prediction the motion vectors of each block are transmitted.

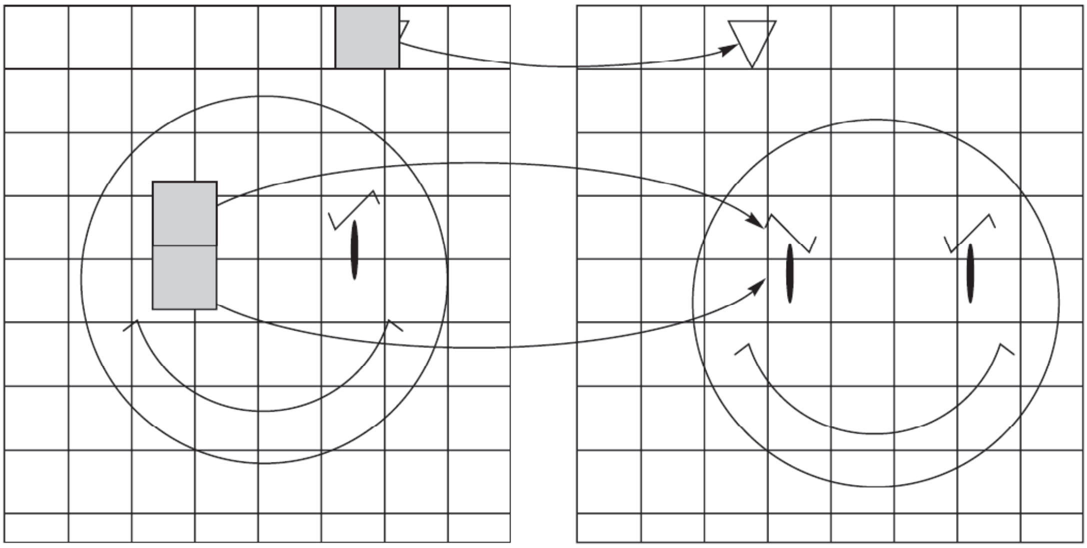


Figure 4.4: Motion compensation example [65].

Medical sequences are not known to have much movement, as the internal organs are not expected to shift during an examination. However, there might be some differences between slices that can be corrected by using motion compensation.

Therefore, it is expected that if we use motion compensation to place the reference pixels, see Figure 4.1, in a more appropriate way on the reference slices the compression efficiency might be improved. In this approach, the slices are divided into fixed sized blocks and a block-matching motion compensation is performed. At this stage the motion compensation can be done in two ways: prior to the encoding process or in the encoding process loop, in order to optimise the placement; both of these methods were tested in MRP.

During our most recent review of the state-of-the-art, two papers describing a similar method for MRP were found, [66, 67], namely using inter-slice prediction with motion compensation. There are, however, some differences in the implementations, for instance in the context calculation, which will be explained in the following sections.

4.4.1 Experimental results

In order to assess the efficiency added by the motion compensation, common natural test sequences were used for the initial experiments, as this type of sequences exhibit more movement than medical sequences and we have a baseline reference in [67]. For the experimental tests, blocks of size 4-by-4, 8-by-8, 16-by-16 and 32-by-32 pixels were used, and the cost of the encoding the motion vectors were not taken into account in this preliminary results. Bi-directional prediction in MRP was used with the following parameters: $K_0 = 30$, $K_1 = 20$, $K_2 = 25$, $K_3 = 20$, $K_4 = 13$ and $K_5 = 13$; the same as used in [67], with three B type slices between references following the HEVC type scheme. In [67], however, Variable Block Size (VBS) is used for the motion compensation. The results for both approaches are shown in Table 4.19, for the motion compensation outside of the encoding loop, and Table 4.20 for the in-loop motion compensation. In these tables the two first columns represent the results of [67] and of HEVC type bi-directional prediction from Section 4.3. The remaining columns represent the HEVC type bi-directional prediction with motion compensation with the represented block-size. Tables 4.21 and 4.22 show the percentage difference to the results of [67].

Table 4.19: Motion compensation compression results for natural sequences (in bpp).

Sequence	[67]	No MC	4-by-4	8-by-8	16-by-16	32-by-32
Carphone	2.669	3.053	2.751	3.075	3.127	3.136
Container	2.257	2.281	2.333	2.336	2.311	2.291
Foreman	2.757	2.820	2.769	2.944	2.968	3.011
Mobile	3.507	3.678	3.984	3.817	3.759	3.768
News	1.329	1.337	1.302	1.348	1.366	1.370
Tempete	3.292	3.335	3.571	3.523	3.480	3.468
Average	2.635	2.751	2.785	2.840	2.835	2.840

First only the HEVC type bi-directional prediction is compared, with and without motion compensation. From these tables, it is possible to infer that, on average, only the in-loop

Table 4.20: In-loop motion compensation compression results for natural sequences (in bpp).

Sequence	[67]	No MC	4-by-4	8-by-8	16-by-16	32-by-32
Carphone	2.669	3.053	2.749	3.040	3.058	3.049
Container	2.257	2.281	2.265	2.312	2.298	2.285
Foreman	2.757	2.820	2.555	2.807	2.822	2.822
Mobile	3.507	3.678	3.781	3.703	3.671	3.663
News	1.329	1.337	1.353	1.328	1.332	1.336
Tempete	3.292	3.335	3.369	3.405	3.357	3.331
Average	2.635	2.751	2.679	2.766	2.756	2.748

Table 4.21: Percentage of compression efficiency difference between the proposed motion compensation and the results of [67].

Sequence	No MC	4-by-4	8-by-8	16-by-16	32-by-32
Carphone	-14.4	-3.1	-15.2	-17.2	-17.5
Container	-1.1	-3.4	-3.5	-2.4	-1.5
Foreman	-2.3	-0.4	-6.8	-7.7	-9.2
Mobile	-4.9	-13.6	-8.8	-7.2	-7.4
News	-0.6	2.0	-1.4	-2.8	-3.1
Tempete	-1.3	-8.5	-7.0	-5.7	-5.3
Average	-4.4	-5.7	-7.8	-7.6	-7.8

Table 4.22: Percentage of compression efficiency difference between the proposed in-loop motion compensation and the results of [67].

Sequence	No MC	4-by-4	8-by-8	16-by-16	32-by-32
Carphone	-14.4	-3.0	-13.9	-14.6	-14.2
Container	-1.1	-0.4	-2.4	-1.8	-1.2
Foreman	-2.3	7.3	-1.8	-2.4	-2.4
Mobile	-4.9	-7.8	-5.6	-4.7	-4.4
News	-0.6	-1.8	0.1	-0.2	-0.5
Tempete	-1.3	-2.3	-3.4	-2.0	-1.2
Average	-4.4	-1.7	-5.0	-4.6	-4.3

motion compensation with blocks of 4-by-4 and 32-by-32 pixels show improvements in the compression efficiency. Nevertheless, as explained before the cost of encoding the motion vectors was not taken into account, so we have a small margin of 0.072 bpp and of 0.003 bpp, for blocks of 4-by-4 and 32-by-32 pixels, respectively.

We will now compare the results from [67] and the results from the HEVC bi-directional prediction without motion compensation. From these tables we can observe a loss in compression efficiency for all the sequences, with the results for the Carphone sequence showing the highest loss, of 14.4% or 0.38 bpp. This higher difference for the Carphone sequence can be explained by the higher compression efficiency gain this sequence has, resulting from the motion compensation when comparing, for instance, columns 'No MC' and '4-by-4', with a gain of 0.30 bpp[1]. Overall, when the results of 'No MC' are compared with the results of [67] there is a loss of 4.4%, or 0.12 bpp, in compression efficiency. It is worth noticing the similarity of the results in both cases, when we consider that our method method does not have the added computational complexity of the motion estimation.

Given the low margin of the gains in efficiency added by the motion compensation versus the cost in computational complexity this method was not adopted, as it is expected that the margins, to exist, would be even lower for the medical sequences.

4.5 Optimal compression plane in MRP Video

In Section 3.2.3 we described an algorithm by Anmin Liu, et al [55, 56], to automatically choose the best plane to align the slices for the encoding process. The Optimal Compression Plane (OCP) algorithm was already tested in the original version of MRP. As several improvements to the MRP algorithm have been performed, it is of interest to evaluate the impact of the OCP algorithm for MRP with inter-slice prediction.

In Table 3.9, the third column shows that the OCP algorithm only chooses a different compression plane for four of the medical sequences, regarding to inter-slice prediction, thus, only those four sequences will be evaluated. Table 4.23 shows the compression results using MRP with and without the use of OCP algorithm, for the uni-directional and bidirectional prediction, with the parameters set in Section 4.3. In this table, the results for the video encoder with the highest compression efficiency, HEVC RExt Random Access, are also shown. In Chapter 3, we did not use all the slices in the sequences, in order to adapt the pre-processing techniques to the encoders. One of these techniques was the OCP algorithm, then the number of used slices is the same as described in Chapter 3.

From this table, it can be observed a slight loss in compression efficiency when only the uni-directional prediction in MRP is considered, roughly 0.01 bpp. The same happens for the bi-directional prediction, although showing a higher efficiency loss of almost 0.2 bpp. This result is also supported by the results obtained for HEVC RExt, which shows a loss in compression efficiency of 0.17 bpp. In papers [55, 56], the authors also did not consider the bi-directional prediction when using H.264, which leads us to conclude that the algorithm is not efficient in those cases. Given that the algorithm resulted in a lower

[1] Once again the motion vectors cost is not taken into account.

compression efficiency, the use of OCP was not included in the final implementation of MRP.

Table 4.23: Compression results for inter-slice prediction using OCP (in bpp).

		MRP			MRP + OCP	
Sequences	HEVC RExt	Uni	Bi	HEVC RExt + OCP	Uni	Bi
carotid	1.383	1.027	0.979	1.391	1.081	1.068
skull	1.822	1.587	1.344	2.038	1.489	1.486
ped_chest	1.515	1.214	0.986	1.713	1.269	1.269
sag_head	1.771	1.602	1.348	2.036	1.607	1.607
Average	1.623	1.357	1.164	1.794	1.362	1.358

4.6 Contributions comparison

In this research, several improvements were performed to the original MRP algorithm, in order to add inter-slice prediction support, and thus improving the compression efficiency. Two types of inter-slice prediction were added to MRP, uni-directional prediction, using only one slice as reference, and bi-directional prediction, using two slices as references. In the case of the bi-directional prediction two modes are used, MPEG-2 type bi-directional prediction and HEVC type bi-directional prediction, where B-type slices were also be used as inter-slice prediction references.

In Table 4.24 the optimal parameters for MRP, determined for the various types of inter-slice prediction, are shown. These are the values that will be used when referring to a given type of inter-slice prediction.

Table 4.24: Optimal parameters used in the various types of inter-slice prediction.

Prediction type	K_0	K_1	K_2	K_3	K_4	K_5	B	D
Uni-directional		12	25	-	-	-	0	
MPEG-2	30	20	13	6	13	13	1	1
HEVC		20	5	12	13	13	7	

Table 4.25 shows the various contributions improvements, in terms of compression efficiency. The results of the use of pixel-wise difference predictor are also shown. In Table 4.26 the percentage gain of each improvement is shown, when compared to the original MRP algorithm (in Table 2.3).

These tables show that all the contributions result on the increase of the compression efficiency, with percentage gains ranging from 34.5% to 46.1%, on average, when compared to the original MRP algorithm.

Table 4.25: Comparison of the improvements made to MRP (results in bpp).

Sequence	MRP Uni-pred	MRP Uni-pred Diff	MRP MPEG-2 Bi-pred	MRP MPEG-2 Bi-pred Diff	MRP HEVC Bi-pred	MRP HEVC Bi-pred Diff
Aperts	0.539	0.532	0.498	0.521	0.491	0.517
carotid	1.055	1.044	1.002	0.960	1.008	0.959
skull	1.529	1.253	1.348	1.205	1.298	1.185
wrist	0.728	0.641	0.641	0.616	0.587	0.604
liver_t1	1.657	1.479	1.511	1.432	1.443	1.406
liver_t2e1	1.221	0.930	1.040	0.895	0.984	0.870
ped_chest	1.230	1.099	1.100	1.052	0.994	1.044
sag_head	1.582	1.366	1.382	1.299	1.332	1.271
Average	1.193	1.043	1.065	0.997	1.017	0.982

Table 4.26: Percentage of compression efficiency gains of the MRP proposed improvements.

Sequence	MRP Uni-pred	MRP Uni-pred Diff	MRP MPEG-2 Bi-pred	MRP MPEG-2 Bi-pred Diff	MRP HEVC Bi-pred	MRP HEVC Bi-pred Diff
Aperts	30.4	31.3	35.7	32.7	36.6	33.3
carotid	23.2	24.0	27.1	30.2	26.6	30.2
skull	34.4	46.2	42.1	48.3	44.3	49.1
wrist	37.9	45.3	45.4	47.5	49.9	48.5
liver_t1	35.8	42.7	41.5	44.5	44.1	45.5
liver_t2e1	29.1	46.0	39.6	48.0	42.8	49.5
ped_chest	47.4	53.0	52.9	55.0	57.5	55.3
sag_head	30.6	40.1	39.4	43.0	41.6	44.2
Average	34.5	42.7	41.5	45.2	44.1	46.1

It is also possible to observe, as expected, that the best results are obtained when the HEVC with bi-directional prediction and the pixel-wise difference predictor is used, with 0.98 bpp, on average, corresponding to a 46.1% gain in compression efficiency, or approximately 0.8 bpp. Therefore, on the following comparisons only the HEVC bi-directional prediction will be considered, with and without the pixel-wise difference predictor.

4.6.1 Comparison with state-of-the-art lossless encoders

In this section we will compare the contributions results with state-of-the-art lossless encoders. The compression results of the pixel-wise difference will also be shown. These results can be seen in Table 4.27, where the original MRP algorithm and HEVC RExt are used, given that in the previous chapters they were the ones with the better compression efficiency.

Table 4.27: Comparison of the proposed alterations to MRP with the original encoder and HEVC RExt (results in bpp).

Sequence	HEVC RExt	HEVC RExt Diff	MRP	MRP Diff	MRP HEVC Bi-pred	MRP HEVC Bi-pred Diff
Aperts	0.728	0.673	0.775	0.624	0.491	0.517
carotid	1.424	1.272	1.374	1.144	1.008	0.959
skull	1.766	1.445	2.329	1.640	1.298	1.185
wrist	1.002	0.861	1.173	0.844	0.587	0.604
liver_t1	2.052	1.852	2.582	1.756	1.443	1.406
liver_t2e1	1.509	1.228	1.722	1.305	0.984	0.870
ped_chest	1.536	1.302	2.337	1.344	0.994	1.044
sag_head	1.748	1.510	2.279	1.785	1.332	1.271
Average	1.471	1.268	1.821	1.305	1.017	0.982

At the end of Chapter 2, we have seen that the lossless encoder with the higher compression efficiency was HEVC RExt. In Chapter 3, several pre-processing techniques, that improved the compression efficiency of all encoders, were studied, and we were able to improve the compression efficiency of MRP, but the compression efficiency of HEVC RExt was also improved and it remained, by a small margin, the most efficient lossless encoder, when encoding the pixel-wise difference.

The current results in Table 4.27 show that we were able to surpass the compression efficiency of HEVC RExt, with and without the use of the pixel-wise difference predictor. Regarding to HEVC RExt using pixel-wise difference we have a gain in compression efficiency of 22.6%, on average, due to a difference of 0.29 bpp. If we consider the original HEVC RExt in comparison to our best result, we have a compression efficiency gain of 32.2%, on average, due to a difference of 0.49 bpp.

4.7 Summary

In this chapter we have proposed several contributions to the MRP original algorithm. We started by adjusting the calculation of the context to the contributions we were planning, namely for the inter-slice prediction. Then we proceed to add the inter-slice prediction support to MRP, using two types of prediction: uni-directional and bi-directional prediction. One of the advantages of bi-directional prediction is that we might perform an easier prediction of objects that were not present previously. Inside the bi-directional prediction we have two modes, MPEG-2 type B slices and HEVC type B slices. The main difference between these modes is that the HEVC type B slices can be used as reference slices to other slices of the same type. Our experimental results showed, as expected, that the best results happen when using the HEVC type bi-directional prediction.

Other contributions were tested in MRP, namely motion compensation, to better place the reference pixels on a reference slice, and the use of OCP algorithm, to better choose the optimal compression plane for the video encoding. However, both of these contributions did not improve the overall compression efficiency of the MRP algorithm and were, therefore, not included in the final version.

Chapter 5

Conclusions

Medical imaging technologies have known a huge development in recent decades. This led to a more extensive use of medical images in medicine in general, mainly with the objective of having more informed diagnosis. As stated in Chapter 1, the resulting exams need to be kept for several years. These facts have led to an extra pressure to the medical images archiving databases, with the consequently increase of storage costs. Thus, the need for compression of medical images is an active topic in image processing research.

DICOM standard recommended lossless encoders do not include inter-slice prediction in their algorithms. This opens several paths for research in medical imaging compression. As one of the more efficient lossless compression algorithms, the MRP encoder was chosen has a starting point for this work.

Initially, in Chapter 3, we proposed different processing techniques to better adapt the input images to the characteristics of the MRP algorithm. Two of the techniques that had a higher impact on the compression efficiency were the changing of the slices orientation for the compression process and the pixel-wise difference predictor. In the first, we simply change the slices alignment to other than the usual XY one. This changes the images characteristics and some of the usual XY inter-slice redundancy can then be spatially exploited in the encoder. It was shown that the best plane in which to align slices was the YZ plane. Changing the orientation of the slices led to an increase of the compression efficiency of 25.9%, or 0.47 bpp, for MRP.

However, the most efficient processing technique was the pixel-wise difference predictor. This is a simple inter-slice predictor that performs a difference between co-located pixels in adjacent slices (see Figure 3.5). The resulting residual image, that for the used images has eight bits-per-pixel, is provided to the encoders. By using this predictor we expect to exploit the inter-slice redundancy of medical images, even when using intra encoders, like MRP. Using this predictor led to an increase of compression efficiency in MRP of 27.9%, or 0.51 bpp, on average.

In Chapter 2 it was shown that the encoder with the highest compression efficiency for the used medical images was HEVC RExt with its Random Access Profile, with 1.47 bpp on average. At this point the result obtained for MRP with the pixel-wise difference predictor was 1.31 bpp. However, considering the application of this predictor prior to the encoding in HEVC RExt, the obtained average result was 1.28 bpp. Therefore, HEVC RExt was still the encoder with the highest compression efficiency, when considering the use of the proposed techniques.

Another technique worth mention is the histogram packing. It was shown that this technique did not work for the used images. However, with medical images of higher resolution and bit-depth, in the case of JPEG-LS, it was shown that it can increase the compression efficiency of the encoder by up to 54.8%. Thus, this is an interesting technique to take into account in future MRP implementations.

Considering the efficient compression obtained by MRP when using the proposed procesing techniques, when compared for instance with HEVC RExt, the next aim of this work was to contribute to this algorithm. Thus, in Chapter 4 the first objective was to add inter-slice prediction support to this encoder, as it was shown that it could enhance the compression efficiency.

For inter-slice prediction, the algorithms of the encoder were basically extended to add support to reference pixels in a previous slice, using the support shown in Figure 4.1. This support was placed around the co-located pixel in the reference slice. Optimisations were performed resulting in the optimal values for the prediction parameters. Using the optimal parameters, the compression efficiency of MRP was increased by 34.5%, surpassing the results obtained with the processing techniques.

State-of-the-art video encoders employ even more slice references for the inter-slice prediction. Hence, bi-directional prediction support was also added to MRP, regarding two different cases: MPEG-2 and HEVC B-type slices, using the same reference pixels support as in the unidirectional prediction case. Once again, optimisations were made that provided the optimal values for the prediction parameters. The results show that the compression efficiency was increased, on average, by 41.5%, for MPEG-2 B slices, and by 44.1%, for HEVC B slices, when compared with MRP original algorithm. As expected, given that it is a more recent standard, the HEVC B-type slices result in a higher compression efficiency than the MPEG-2 B-type slices. This is explained by the higher proximity between reference slices in the HEVC B slices case.

Finally, the pixel-wise difference predictor of Chapter 3 was also implemented in MRP. In this implementation, it was taken into account the possibility of the resulting residuals having nine bits-per-pixel and it was shown that usually the number of pixels that cannot be represented by 8 bits is very low. The compression performance was improved with this predictor by 12.6%, for the uni-directional prediction, and by 3.4%, for the HEVC type bi-directional prediction. The best result was obtained by using the HEVC B-type slices and pixel-wise difference predictor in MRP , with 0.98 bpp. Comparing this result with DICOM and state-of-the-art encoders we have a compression efficiency gain of 57.1%, regarding to JPEG-LS, and of 33.2%, regarding to HEVC RExt (23.4% if we consider HEVC RExt using the pixel-wise difference predictor), on average.

The research presented in this dissertation showed that the currently used lossless encoders in the DICOM standard cannot compete with more recent state-of-the-art encoders. Namely, HEVC RExt with its Random Access profile has a compression efficiency 35.8% better than that of JPEG-LS. Also the contributions made in this work were able to improve the compression efficiency of MRP and surpass that of HEVC RExt. Thus having the best compression efficiency of the encoders under study.

In future work the extension of the MRP to encode images with bit depths up to 16 bits-per-pixel will be performed. In connection with this, histogram packing will also be further studied and applied in the MRP algorithm. Additionally, different template formats for the reference pixels positions will be analysed.

Bibliography

[1] L. W. Goldman, “Principles of CT: multislice CT.” *Journal of Nuclear Medicine Technology*, vol. 36, no. 2, pp. 57–68; quiz 75–6, 2008. [Online]. Available: http://www.biomedsearch.com/nih/Principles-CT-multislice/18483143.html

[2] J. Taquet and C. Labit, “Hierarchical oriented predictions for resolution scalable lossless and near-lossless compression of ct and mri biomedical images,” *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2641–2652, May 2012.

[3] ITU, “ITU-T Recommendation T.87: Information technology – lossless and near-lossless compression of continuoues-tone still images - Baseline,” International Telecommunication Union, Geneva, Switzerland, ITU, 1998. [Online]. Available: https://jpeg.org/jpegls/

[4] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS,” *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug 2000. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=855427

[5] DICOM - Digital Imaging and Communications in Medicine. (2015, December) Strategic document. [Online]. Available: http://dicom.nema.org/dicom/geninfo/Strategy.pdf

[6] DICOM, “DICOM Standard 2015 PS3.5 2015c - Data Structures and Encoding,” Digital Imaging and Communications in Medicine, Arlington, VA, USA, DICOM, 2015. [Online]. Available: http://dicom.nema.org/medical/dicom/current/output/pdf/part05.pdf

[7] ISO, “ISO 12052:2006: Health informatics – digital imaging and communication in medicine (dicom) including workflow and data management,” ISO, Tech. Rep., 2006. [Online]. Available: http://www.iso.org/iso/catalogue_detail?csnumber=43218

[8] X. Wu, N. Memon, and K. Sayood, “A context-based, adaptive, lossless/nearly-lossless coding scheme for continuous-tone images,” in *ISO/IEC JTC 1/SC 29/WC*

1 document No, 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.5882

[9] X. Wu and N. Memon, "CALIC - A context based adaptive lossless image codec," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings, 1996 IEEE International Conference on*, vol. 4, May 1996, pp. 1890–1893 vol. 4.

[10] ITU, "ITU-T Recommendation T.809: Information technology – JPEG 2000 image coding system: Extensions for three-dimensional data," International Telecommunication Union, Geneva, Switzerland, ITU, 2011. [Online]. Available: https://jpeg.org/jpeg2000/index.html

[11] P. Schelkens, A. Munteanu, A. Tzannes, and C. Brislawn, "JPEG2000. part 10. volumetric data encoding," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, May 2006, pp. 4 pp.–3877. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1693474

[12] T. Bruylants, "Volumetric image compression with JPEG2000," *SPIE Newsroom*, 2007. [Online]. Available: http://dx.doi.org/10.1117/2.1200706.0779

[13] M. B. de Carvalho, E. A. da Silva, and W. A. Finamore, "Multidimensional signal compression using multiscale recurrent patterns," *Elsevier Signal Processing*, vol. 82, no. 11, pp. 1559–1580, Nov. 2002. [Online]. Available: http://dx.doi.org/10.1016/S0165-1684(02)00302-X

[14] N. Francisco, N. Rodrigues, E. da Silva, M. de Carvalho, and S. de Faria, "Efficient recurrent pattern matching video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 8, pp. 1161–1173, Aug. 2012. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6193166

[15] N. C. Francisco, N. M. M. Rodrigues, and E. A. B. da Silva, "Contribuições à codificação eficiente de imagem e vídeo utilizando recorrência de padrões multiescala," Ph.D. dissertation, Universidade Federal do Rio de Janeiro, Nov. 2012. [Online]. Available: http://www02.smt.ufrj.br/~eduardo/teses/Tese_doutoramento_nelson_francisco.pdf

[16] D. Graziosi, N. Rodrigues, E. da Silva, M. de Carvalho, and S. Maciel de Faria, "Lossy and lossless image encoding using multi-scale recurrent pattern matching," *IET Image Processing*, vol. 7, no. 6, pp. 556–566, August 2013. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6616264

[17] N. Francisco, N. Rodrigues, E. da Silva, M. de Carvalho, and S. de Faria, "Video compression using 3D multiscale recurrent patterns," in *IEEE International*

Symposium on Circuits and Systems (ISCAS), May 2013, pp. 1412–1415. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6572120

[18] ITU, "ITU Recomendation H.264: Advanced video coding for generic audiovisual services," ITU-T, Tech. Rep., 2014. [Online]. Available: http://www.itu.int/rec/T-REC-H.264-201402-S/en

[19] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1218189

[20] G. J. Sullivan, P. N. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions," in *Applications of Digital Image Processing XXVII*, vol. 5558 of Proceedings Of SPIE. SPIE, November 2004, pp. 454–474. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=850444

[21] ITU, "ITU Recomendation H.265: High efficiency video," ITU-T, Tech. Rep., 2015. [Online]. Available: https://www.itu.int/rec/T-REC-H.265-201504-I/en

[22] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, December 2012. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6316136

[23] G. J. Sullivan, J. M. Boyce, Y. Chen, J.-R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (HEVC)," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1001–1016, December 2013. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6630053

[24] I. Matsuda, H. Mori, and S. Itoh, "Design of a minimum-rate predictor and its application to lossless image coding," in *Signal Processing Conference, 2000 10th European*, Sep. 2000, pp. 1–4.

[25] I. Matsuda, H. Mori, and S. Itoh, "Lossless coding of still images using minimum-rate predictors," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 1, 2000, pp. 132–135 vol.1. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=900912

[26] I. Matsuda, H. Mori, J. Maeda, and S. Itoh, "Design and evaluation of minimum-rate predictors for lossless image coding," *Systems and Computers in Japan*, vol. 38, no. 5, pp. 90–98, 2007. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/scj.10318/abstract

[27] I. Matsuda, N. Ozaki, Y. Umezu, and S. Itoh, "Lossless coding using variable block-size adaptive prediction optimized for each image," in *Signal Processing Conference, 2005 13th European*, Sep. 2005, pp. 1–4.

[28] I. Matsuda, Y. Umezu, N. Ozaki, J. Maeda, and S. Itoh, "A lossless coding scheme using adaptive predictors and arithmetic code optimized for each image," *Systems and Computers in Japan*, vol. 38, no. 4, pp. 1–11, 2007. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/scj.20630/abstract

[29] C. Cavaro-Menard, L. Zhang, and P. L. Callet, "Diagnostic quality assessment of medical images: Challenges and trends," in *Visual Information Processing (EUVIP), 2010 2nd European Workshop on*, Jul. 2010, pp. 277–284. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5699147

[30] A. Naït-Ali and C. Cavaro-Ménard, *Compression of Biomedical Images and Signals*, ser. ISTE. Wiley, 2008. [Online]. Available: https://books.google.pt/books?id=WsxQAAAAYAAJ

[31] Siemens Healthcare. (2016, March) Computed tomography. Medical Radiation. [Online]. Available: http://www.medicalradiation.com/types-of-medical-imaging/imaging-using-x-rays/computed-tomography-ct/

[32] Siemens Healthcare. (2016, March) Magnetic resonance imaging. Medical Radiation. [Online]. Available: http://www.medicalradiation.com/types-of-medical-imaging/other-types-of-medical-imaging/magnetic-resonance-imaging/

[33] Center for Image Processing Research. (2016, Mar.) Medical sequences. [Online]. Available: http://www.cipr.rpi.edu/resource/sequences/sequence01.html

[34] I. Bankman, *Handbook of Medical Imaging: Processing and Analysis Management*, ser. Biomedical Engineering. Elsevier Science, 2000. [Online]. Available: http://www.sciencedirect.com/science/book/9780120777907

[35] Royal College of Radiologists, "The adoption of lossy image data compression for the purpose of clinical interpretation," version 1.1, Apr. 2008, accessed: April 2016. [Online]. Available: https://www.rcr.ac.uk/it-guidance-adoption-lossy-image-data-compression-purpose-clinical-interpretation

[36] Royal Australian and New Zealand College of Radiologists, "A guideline for the use of image compression in diagnostic imaging," Nov. 2010, accessed: April 2016. [Online]. Available: http://www.ranzcr.edu.au/documents-download/document-library-2/document-library-3/574-a-guideline-for-the-use-of-image-compression-in-diagnostic-imaging

[37] A. Moffat, R. Neal, and I. H. Witten, “Arithmetic coding revisited,” in *Data Compression Conference , 1995. DCC ’95. Proceedings*, Mar. 1995, pp. 202–211. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=515510

[38] S. Golomb, “Run-length encodings (corresp.),” *IEEE Trans. Inf. Theory*, vol. 12, no. 3, pp. 399–401, 1966. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1053907

[39] ITU, “ITU-T Recommendation T.812: Information technology – JPEG 2000 image coding system: An entry level JPEG 2000 encoder,” International Telecommunication Union, Geneva, Switzerland, ITU, 2007. [Online]. Available: https://www.itu.int/rec/T-REC-T.812/en

[40] M. Adams and R. Ward, “Wavelet transforms in the jpeg-2000 standard,” in *Communications, Computers and signal Processing, 2001. PACRIM. 2001 IEEE Pacific Rim Conference on*, vol. 1, 2001, pp. 160–163. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=953547

[41] ITU, “ITU-T Recommendation T.802: Information technology – JPEG 2000 image coding system: Motion JPEG 2000,” International Telecommunication Union, Geneva, Switzerland, ITU, 2005. [Online]. Available: https://jpeg.org/jpeg2000/index.html

[42] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=847830

[43] X. Li and M. T. Orchard, “Edge-directed prediction for lossless compression of natural images,” *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 813–817, Jun. 2001. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=923277

[44] Y.-L. Lee, K.-H. Han, and G. J. Sullivan, “Improved lossless intra coding for H.264/MPEG-4 AVC,” *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2610–2615, Sep. 2006. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1673442

[45] X. Li, “Least-square Prediction for Backward Adaptive Video Coding,” *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–13, Jan. 2006. [Online]. Available: http://asp.eurasipjournals.springeropen.com/articles/10.1155/ASP/2006/90542

[46] T. Kim, K. Park, and Y. Hong, “Video watermarking technique for h.264/avc,” *Optical Engineering*, vol. 51, no. 4, pp. 047 402–1–047 402–12, 2012. [Online]. Available: http://dx.doi.org/10.1117/1.OE.51.4.047402

[47] G. N. N. Martin, “Range encoding: an algorithm for removing redundancy from a digitized message,” in *Proceedings of the Video and Data Recording Conference*, Southampton, UK, Mar. 1979.

[48] 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, July 2009.

[49] V. Sanchez, R. Abugharbieh, and P. Nasiopoulos, “3D scalable lossless compression of medical images based on global and local symmetries,” in *IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 2525–2528.

[50] J. Taquet and C. Labit, “Near-lossless and scalable compression for medical imaging using a new adaptive hierarchical oriented prediction,” in *IEEE International Conference on Image Processing (ICIP)*, Sept 2010, pp. 481–484.

[51] Z. Xu, J. Bartrina-Rapesta, V. Sanchez, J. Serra-Sagrista, and J. Munoz-Gomez, “Diagnostically lossless compression of x-ray angiography images based on automatic segmentation using ray-casting and α-shapes,” in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, Sept 2013, pp. 738–742. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6738152

[52] A. Pinho and A. Neves, “L-Infinity Progressive Image Compression,” in *PCS 2007 - 26th Picture Coding Symposium*, 2007.

[53] A. J. Pinho and A. J. R. Neves, “Progressive lossless compression of medical images,” in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, Apr. 2009, pp. 409–412.

[54] X. Wu, G. Zhai, X. Yang, and W. Zhang, “Adaptive sequential prediction of multidimensional signals with applications to lossless image coding,” *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 36–42, Jan 2011.

[55] A. Liu, W. Lin, and F. Zhang, “Lossless video compression with optimal compression plane determination,” in *IEEE International Conference on Multimedia and Expo, 2009. ICME 2009*, June 2009, pp. 173–176. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5202464

[56] A. Liu, W. S. Lin, M. Paul, F. Zhang, and C. Deng, “Optimal compression plane for efficient video coding,” *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2788–2799, 2011. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5746533

[57] A. J. Pinho, “On the impact of histogram sparseness on some lossless image compression techniques,” in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 3, 2001, pp. 442–445 vol.3. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=958146

[58] R. Starosolski, “Compressing high bit depth images of sparse histograms,” *AIP Conference Proceedings*, vol. 1060, no. 1, pp. 269–272, 2008. [Online]. Available: http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.3037069

[59] A. J. Pinho, “A comparison of methods for improving the lossless compression of images with sparse histograms,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 2, 2002, pp. II–673–II–676 vol.2. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1040040

[60] R. Starosolski, “Compressing images of sparse histograms,” in *SPIE Proceedings . 5959*, 2005, pp. 209–217. [Online]. Available: http://spie.org/Publications/Proceedings/Paper/10.1117/12.624489

[61] R. Starosolski. (2016, Mar.) Higher bit depth medical images. [Online]. Available: http://sun.aei.polsl.pl/~rstaros/mednat/index.html

[62] P. J. S. G. Ferreira and A. J. Pinho, “Why does histogram packing improve lossless compression rates?” *IEEE Signal Process. Lett.*, vol. 9, no. 8, pp. 259–261, Aug. 2002. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1034993

[63] ITU, “Video codec for audiovisual services at p × 64 kbit/s, ITU-T Recommendation H.261,” ITU-T, Tech. Rep., 1993. [Online]. Available: http://www.itu.int/rec/T-REC-H.261-199303-I/en

[64] ITU, “H.262 : Information technology - generic coding of moving pictures and associated audio information: Video,” ITU-T, Tech. Rep., 2012. [Online]. Available: http://www.itu.int/rec/T-REC-H.262-201202-I/en

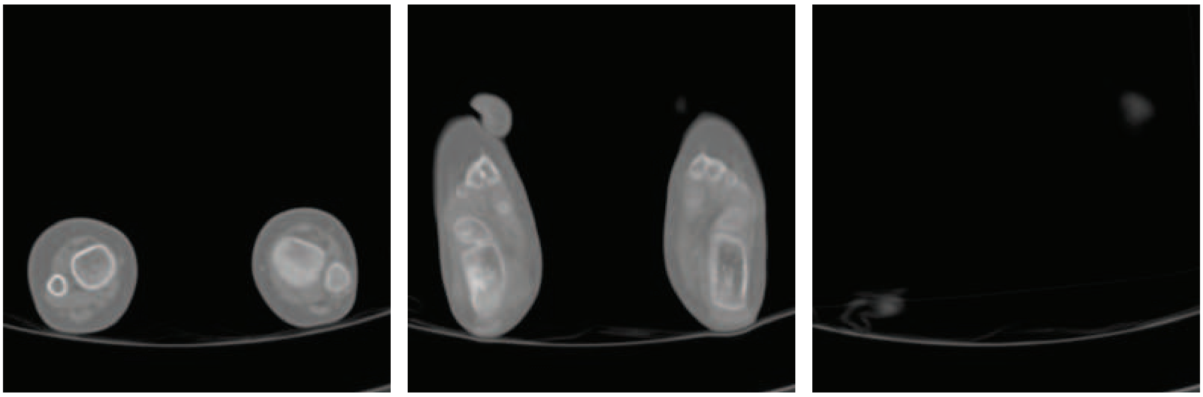
[65] K. Sayood, *Introduction to Data Compression, Third Edition (Morgan Kaufmann Series in Multimedia Information and Systems).* Morgan Kaufmann, 2005. [Online]. Available: http://www.sciencedirect.com/science/book/9780126208627

[66] I. Matsuda, T. Shiodera, and S. Itoh, “Lossless video coding using variable block-size MC and 3D prediction optimized for each frame,” in *Signal Processing Conference, 2004 12th European*, Sep. 2004, pp. 1967–1970. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7079782

[67] I. Matsuda, T. Shiodera, H. Maeda, and S. Itoh, "Lossless video coding using bi-directional 3D prediction optimized for each frame," in *Circuit Theory and Design, 2005. Proceedings of the 2005 European Conference on*, vol. 2, Aug. 2005, pp. II/71–II/74 vol. 2. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1522995

Appendix A

Medical Sequences

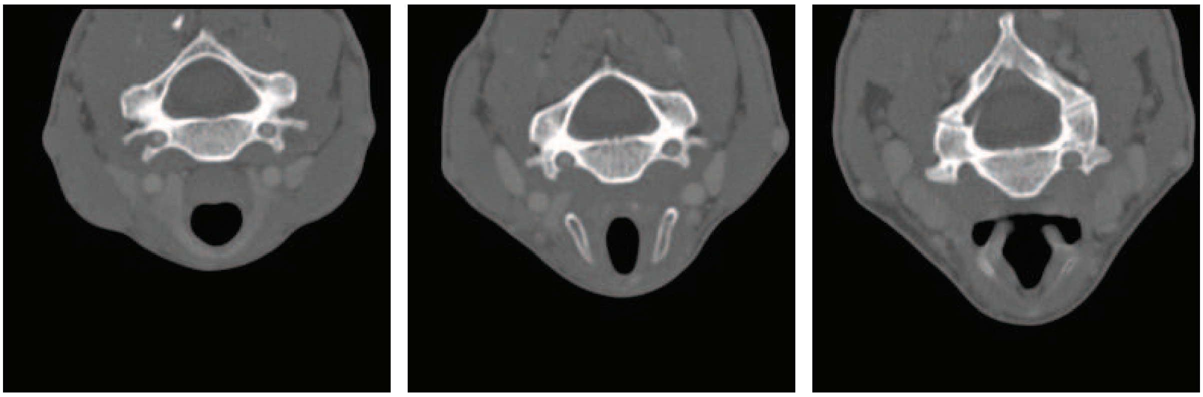


(a) Slice 0.

(b) Slice 48

(c) Slice 96

Figure A.1: Slices of the CT Aperts sequence [33].

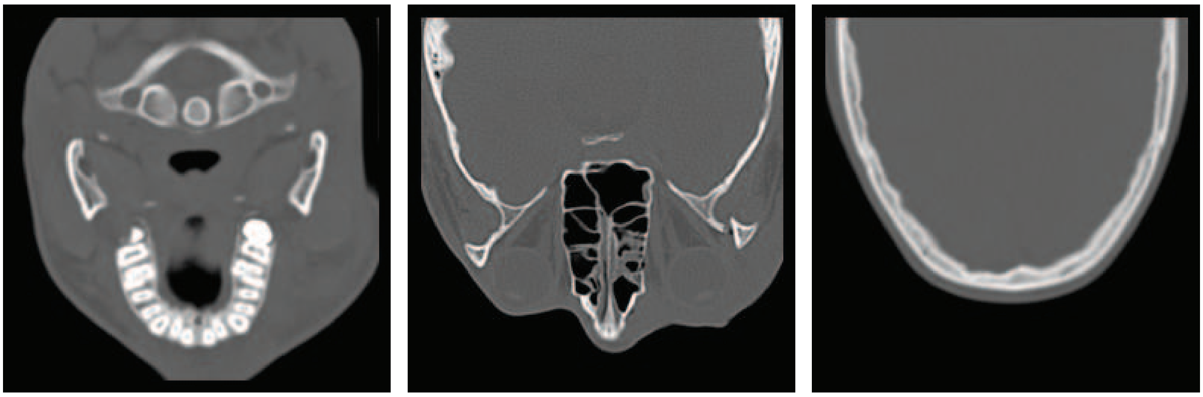


(a) Slice 0.

(b) Slice 37

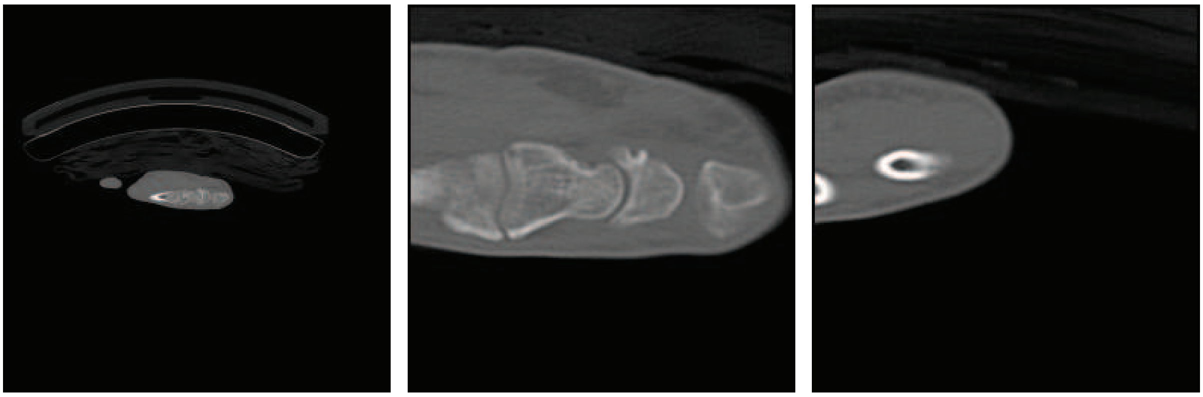
(c) Slice 73

Figure A.2: Slices of the CT carotid sequence [33].



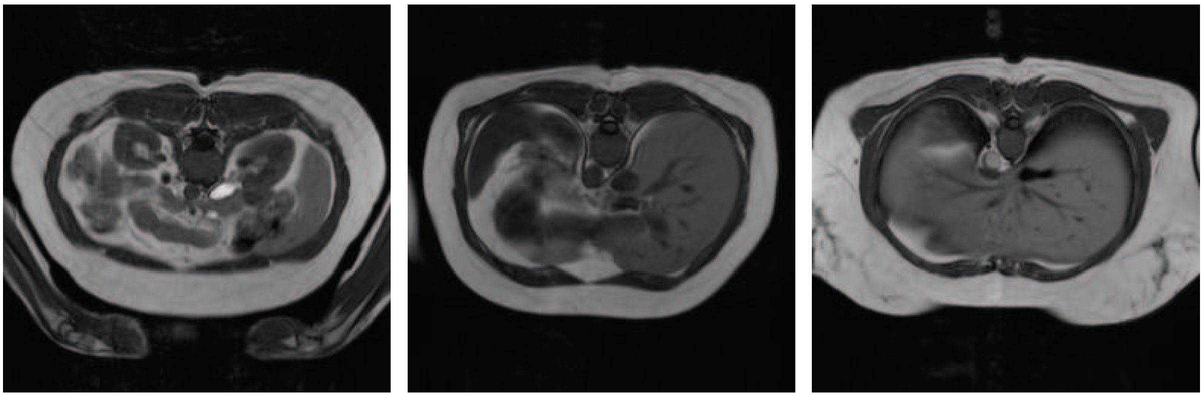
(a) Slice 0. (b) Slice 101 (c) Slice 202

Figure A.3: Slices of the CT skull sequence [33].



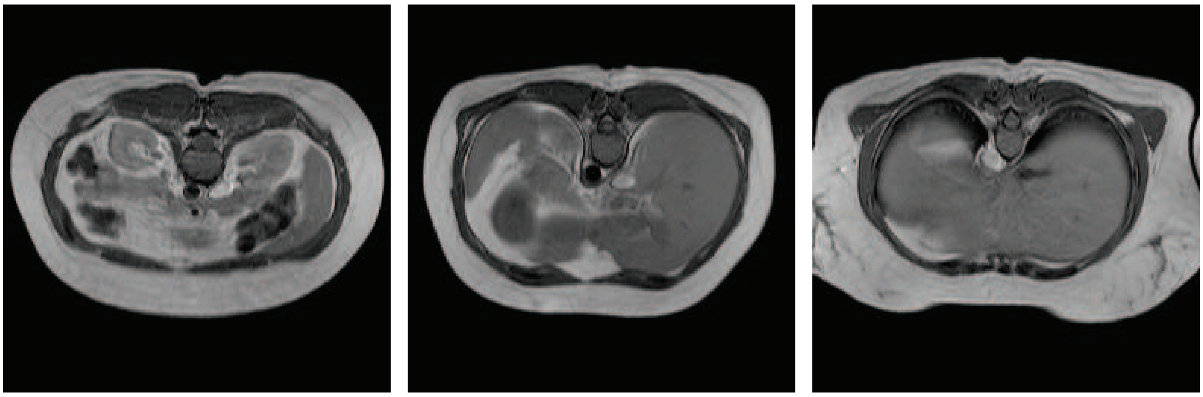
(a) Slice 0. (b) Slice 91 (c) Slice 182

Figure A.4: Slices of the CT wrist sequence [33].



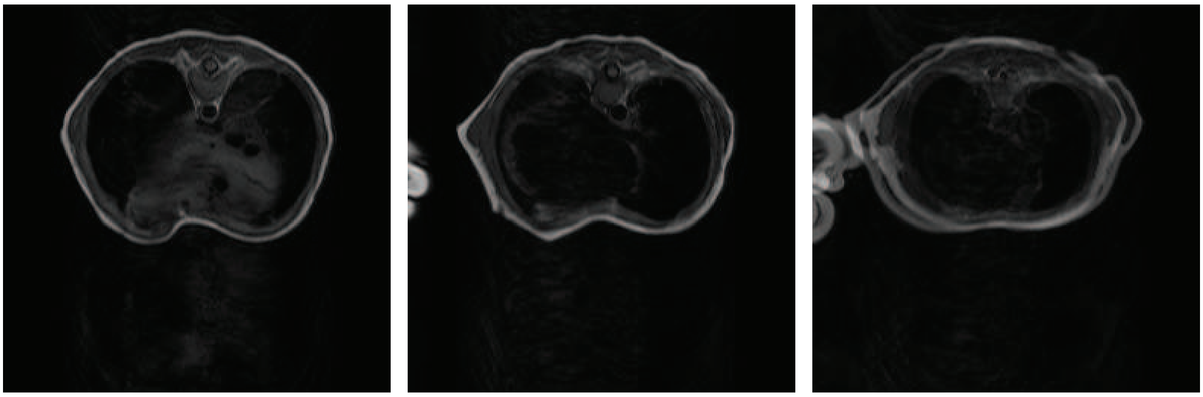
(a) Slice 0. (b) Slice 29 (c) Slice 57

Figure A.5: Slices of the MR liver_t1 sequence [33].



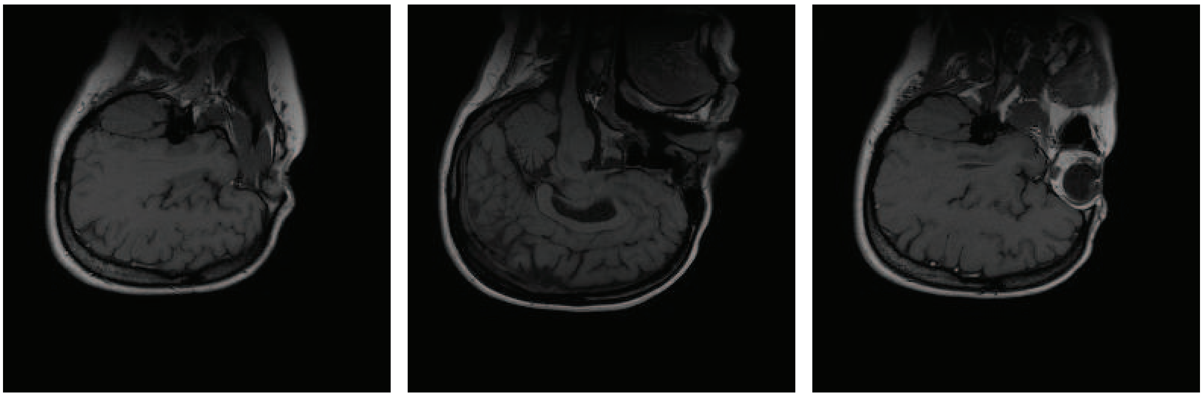
(a) Slice 0. (b) Slice 29 (c) Slice 57

Figure A.6: Slices of the MR liver_t2e1 sequence [33].



(a) Slice 0. (b) Slice 38 (c) Slice 75

Figure A.7: Slices of the MR ped_chest sequence [33].



(a) Slice 0. (b) Slice 29 (c) Slice 57

Figure A.8: Slices of the MR sag_head sequence [33].

Appendix B

Histogram Packing Detailed Results

Table B.1: Number of values actually present in the medical images from [61].

Sequence	Type	Height	Width	Depth	Expected Values	Used Values
cr_17218	CR	1792	2392	12	4096	2068
cr_17220		2048	2500	12	4096	3186
cr_17222		2392	1792	12	4096	2939
cr_4503		2010	1670	10	1024	256
cr_4507		1760	1760	10	1024	1024
cr_4509		2140	1760	10	1024	882
cr_pacem_1		1910	1716	16	65536	24180
cr_pacem_2		1965	1531	16	65536	28627
cr_rtg_jb		746	612	16	65536	3280
cr_siem_01_02		2128	1744	10	1024	913
cr_siem_14_02		2368	1760	10	1024	638
cr_slim_1		2031	1866	16	65536	26539
ct_135960_001	CT	512	512	16	65536	2442
ct_135960_005		512	512	16	65536	2806
ct_17		512	512	12	4096	1883
ct_27154		512	512	12	4096	1300
ct_29513		340	340	12	4096	2570
ct_29920		512	512	12	4096	1723
ct_3030		691	512	16	65536	778
ct_3071		512	512	16	65536	1696
ct_4006		512	512	16	65536	2100

Table B.1 – continued from the previous page

Sequence	Type	Height	Width	Depth	Expected Values	Used Values
ct_4087		512	512	16	65536	1731
ct_4165		512	512	16	65536	1735
ct_tk_kl_piers0021		512	512	16	65536	2644
mr_2321	MRI	512	512	16	65536	894
mr_2331		512	512	16	65536	893
mr_2337		512	512	16	65536	1047
mr_2371		512	512	16	65536	1415
mr_2412		512	512	16	65536	1300
mr_2807		256	256	16	65536	1858
mr_2882		512	512	16	65536	501
mr_2896		512	512	16	65536	604
mr_6624		256	256	16	65536	795
mr_6706		256	256	16	65536	1088
mr_6774		512	512	16	65536	1799
mr_6837		256	256	16	65536	1055
us_19773	US	480	640	8	256	256
us_27704		480	640	8	256	249
us_27743		480	640	8	256	246
us_28279		480	640	8	256	250
us_28282		480	640	8	256	247
us_28289		480	640	8	256	254
us_28322		480	640	8	256	213
us_28329		480	640	8	256	213
us_28348		480	640	8	256	217
us_3393		476	640	8	256	218
us_3403		484	584	8	256	256
us_3405		476	640	8	256	197

Table B.2: Compression results for images from [61], with and without histogram packing, using JPEG-LS (results in bpp).

Sequence	Original	Histogram Compression	Difference (in %)
cr_17218	5.226	4.813	7.9
cr_17220	3.788	3.797	-0.2
cr_17222	4.550	4.558	-0.2

Table B.2 – continued from the previous page

Sequence	Original	Histogram Compression	Difference (in %)
cr_4503	4.734	2.877	39.2
cr_4507	2.189	2.254	-3.0
cr_4509	4.236	4.125	2.6
cr_pacem_1	10.904	9.628	11.7
cr_pacem_2	10.538	9.535	9.5
cr_rtg_jb	11.029	6.720	39.1
cr_siem_01_02	5.243	5.094	2.8
cr_siem_14_02	2.924	2.520	13.8
cr_slim_1	10.760	9.643	10.4
ct_135960_001	6.767	3.583	47.0
ct_135960_005	6.706	3.641	45.7
ct_17	4.599	4.044	12.1
ct_27154	2.600	2.037	21.7
ct_29513	4.829	4.708	2.5
ct_29920	4.617	3.927	14.9
ct_3030	11.493	5.196	54.8
ct_3071	9.034	4.817	46.7
ct_4006	11.290	6.254	44.6
ct_4087	11.535	6.177	46.5
ct_4165	12.010	6.641	44.7
ct_tk_kl_piers0021	8.574	5.324	37.9
mr_2321	11.337	5.220	54.0
mr_2331	11.439	5.327	53.4
mr_2337	8.296	3.986	52.0
mr_2371	8.136	4.008	50.7
mr_2412	10.888	5.327	51.1
mr_2807	12.365	7.886	36.2
mr_2882	1.725	0.815	52.7
mr_2896	9.347	4.312	53.9
mr_6624	12.265	6.071	50.5
mr_6706	12.546	6.745	46.2
mr_6774	10.645	5.551	47.9
mr_6837	11.115	5.691	48.8
us_19773	2.278	2.284	-0.3
us_27704	3.111	3.102	0.3

Table B.2 – continued from the previous page

Sequence	Original	Histogram Compression	Difference (in %)
us_27743	3.232	3.218	0.4
us_28279	2.553	2.553	0.0
us_28282	2.783	2.784	0.0
us_28289	2.339	2.346	-0.3
us_28322	3.283	3.163	3.7
us_28329	3.558	3.425	3.7
us_28348	3.118	3.030	2.8
us_3393	2.584	2.481	4.0
us_3403	2.524	2.532	-0.3
us_3405	1.609	1.646	-2.3
Average	6.734	4.488	33.4

Appendix C

Contributions

This appendix presents a list of the published and submitted papers, resulting from the research work done during this dissertation.

C.1 Published

1. João M. Santos, André F. R. Guarda, Nuno M. M. Rodrigues, Sérgio M. M. Faria, "Contributions to lossless coding of medical images using Minimum Rate Predictors" in *2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, QC, 2015, pp. 2935-2939, [Online]. Available: `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7351340`.

2. André F. R. Guarda, João M. Santos, Luís A. da Silva Cruz, Pedro A. A. Assunção, Nuno M. M. Rodrigues, Sérgio M. M. Faria, "Orientation dependent inter-slice prediction for volumetric medical images" in *10th Conference on Telecommunications, Conftele*, 2015, Aveiro, Portugal, September 2015.

C.2 Submitted

3. André F. R. Guarda, João M. Santos, Luís A. da Silva Cruz, Pedro A. A. Assunção, Nuno M. M. Rodrigues, Sérgio M. M. Faria, "A Method to Improve HEVC Lossless Coding of Volumetric Medical Images", submitted to *Signal Processing: Image Communication*.