Università degli Studi di Cagliari
Electronic Engineering Department

Giaime Ginesu

Volumetric Data Processing and Compression

Ph.D. thesis

Advisor: Prof. Daniele D. Giusto

Dipartimento di Ingegneria Elettrica ed Elettronica

cnit consorzio nazionale interuniversitario per le telecomunicazioni

Rensselaer

UNIVERSITÀ DEGLI STUDI DI CAGLIARI
ELECTRONIC ENGINEERING DEPARTMENT

Giaime Ginesu

Volumetric Data Processing and Compression

Ph.D. thesis

Advisor: Prof. Daniele D. Giusto

Giaime Ginesu, November 2004

A Francesco, Mariarosa, Edmondo, Aida
e Vale per il nostro tempo prezioso!!

Thanks to:

all colleagues and friends at the MCLab, Prof. Pearlman, Sonia&Laurent, Adnan&Carla, Alessandro and Ali.

The optimist proclaims that we live in the best of all possible worlds; and the pessimist fears this is true.

Contents

Introduction

Objectives of the research

Volumetric images are becoming widely used in the scientific world. Medical applications, as well as remote sensing and physical-models simulation often require 3D volume representation, resulting in complex datasets. Such complexity encompasses the representation, processing and compression techniques involved in the management of digital volumetric images. Moreover, these operations are generally compulsory for the correct interpretation of the data and their convenient storage. Although a standardized framework for volumetric data processing and storage has not been currently adopted, much effort is being spent for its definition and acceptability in the scientific community. In addition to the research carried out by universities and high-tech companies, several organizations are working for the definition of such standard. The Joint Photographic Experts Group committee (ISO/IEC SC29 WG01), for instance, has recently launched the JPEG2000 Part10 activity, concerned with the coding of three-dimensional data and the extension of JPEG2000 from planar to volumetric images [JP3D]. Similarly, the Digital Media and Communications in Medicine [DICOM] Standard committee is working on the adoption and integration of multi-frame and 3D compression techniques in the medical field through its working groups 4 (compression) and 17 (3D).

This work, which is the result of my Ph.D. thesis in Electronic Engineering, addresses the problems of volumetric data processing and compression. The main contribution is the design of a novel volumetric coder based on Set Partitioning In Hierarchical Trees [Sai+J96], described in Chapter 2, and its extension with advanced ROI capability (Chapter 4). Most of the design and development work for the proposed coding scheme has been carried out during my stay as a visiting scholar at the Rensselaer Polytechnic Institute (Troy, NY) under the supervision of Prof. Pearlman. As a result, the goal of producing an operational experimental codec has been achieved and extensive testing has been conducted on the proposed algorithms. Another interesting contribution strictly related to the above mentioned activity is described in Chapter 3. Major effort has been spent for the development of several volumetric segmentation methods. These were successively used inside the coding framework to provide an accurate tool for semi-automatic region of interest identification. The most important contribution in this case consists in the definition of a segmentation method based on active contours and inter-slice prediction. Additionally, a seeded region growing method has been developed. Finally, another major contribution is reported in Chapter 5. Although the novel error concealment method for JPEG2000-compressed images is not directly related to volumetric coding, it may be considered as the basis for an extended work on our main topic of interest. In fact, the proposed method may be adapted to fit the 3D case for any scalable wavelet-based coder, such as our volumetric compressor. This is guaranteed by the algorithm being based on a Projections Onto Convex Sets approach, working in the spatial and wavelet domain through filtering and replacement respectively. Consequently, such procedure is not specialized for JPEG2000 only but may be easily modified for any wavelet scheme. The expected result should be an efficient concealment method able to avoid the need for data retransmission in transmission-critical environments.

Another topic related to image processing and compression and covered during my Ph.D. research, but not included in this document is the development of HVS-based metrics for quality assessment of lossy compressed medical images.

Extended index

An extended description of the thesis contents is provided in this paragraph.

Chapter 1: Volumetric Digital Images. The first chapter provides an overview on volumetric imaging and coding. The definitions of volumetric data and voxel are given together with several characteristics of such data modality. Attention is spent on the processing and representation of volumetric data and rendering techniques are briefly discussed. The principles of signal coding are presented and extended first to the case of 2D images and then to 3D image data. For the latter, several issues on coding accuracy and innovative features are discussed. Finally, we provide an extended outline of applications that require volumetric data, both in the remote sensing and medical field.

Chapter 2: Lossy to Lossless Coding of Volumetric Data. The second chapter describes a novel scheme for lossy to lossless volume data compression. The proposed method is based on the 3D extension of the SPIHT algorithm with packet wavelet transform. The need of unitary transform via the integer lifting scheme and the performance comparison of different integer filter kernels is discussed. A new coding tree-structure is defined in order to conform to the proposed 3D packet wavelet structure. The algorithm is extensively tested and shows very good performance, both for lossy and lossless coding.

Chapter 3: ROI Processing. In this chapter we discuss several algorithms for volumetric image segmentation. With the main purpose of Region of Interest extraction, a snake-based method for interactive volumetric data segmentation is presented. The proposed approach is based on a slice-oriented implementation of the theory of active contours and a nodes position estimation algorithm. Particular attention is dedicated to the preprocessing stage, while the whole algorithm is mainly designed for region of interest extraction in a selective volumetric compression framework. Several experiments are produced to confirm the validity of the scheme. An approach based on seeded region growing is then described. The procedure implementation follows a rather classical scheme but allows achieving good results at with low computational cost.

Chapter 4: Semantic-driven medical data compression with ROI. Starting from the results described in the previous chapters, a new coding scheme mainly intended for medical volumetric data is now presented. The proposed method is able to perform lossy to lossless or selective quality compression by taking into account the semantic characteristics of volumetric data. The system is based on a preprocessing stage for background suppression and a SPIHT-based coder with advanced ROI capability. The resulting approach reaches good performances and is particularly fit in a store-and-forward telemedicine framework.

Chapter 5: Concealment of JPEG2000 compressed images. In this chapter, we address the problem of concealing high-frequencies errors at the lower decomposition levels in JPEG2000, by proposing a new approach based on the theory of projections onto convex sets. The error effects are masked by iteratively applying two important procedures: low-pass (LP) filtering in the spatial domain and restoration of the uncorrupted wavelet coefficients in the transform domain. It has been observed that a uniform LP filtering brought to some undesired side-effects that negatively compensated the advantages. This problem has been overcome by applying an adaptive solution exploiting an edge map to choose the optimal filter mask dimension. Simulation results demonstrated the efficiency of the proposed approach.

Chapter 1

Volumetric Digital Images

This chapter provides a helpful overview of volumetric imaging. In Section 1.1 volumetric images are defined. Several issues on processing and representation of volumetric images are illustrated in Section 1.2. Section 1.3 describes the basics of signal compression. Section 1.4 describes the problems related to volumetric image coding. Finally, in section 1.5 we present several applications where volumetric imaging is extensively adopted.

1.1. Definitions

Advances in scientific imaging technology have made it possible to routinely acquire high-resolution 3D images using a variety of imaging modalities. The ability to inspect structural relationships in three dimensions and the ever improving quality of images have tremendously increased the number of scientific applications of volumetric data imaging.

A volumetric image is generally defined by a three dimensional matrix of integer or real values representing the local intensity of the signal of interest (Fig. 1.1). Depending on application, this may characterize the energy for the transmitted, reflected or emitted radiation, the local pressure, density, humidity or current speed, etc. As in the case of bi-dimensional images, the signal must be spatially sampled into many parallelepiped building blocks and its range is subjected to quantization in order to obtain the volumetric digital representation. The basic unit is generally referred to as *voxel*, the 3D counterpart of 2D pixel. Bit depth may vary from 8 to 32 bits depending on application. While 12bits volumes are of common use, 32bits floating point volumes are occasionally used for scientific models of physical datasets. The volume rendering of an engine CT scan is shown in Fig. 1.2. By varying the scalar opacity function values it is possible to fully illustrate the image content. A volumetric image may also be thought of as a collection of planar sections. This interpretation results particularly indicated when dealing with tomographic devices or multi-spectral acquisition systems.

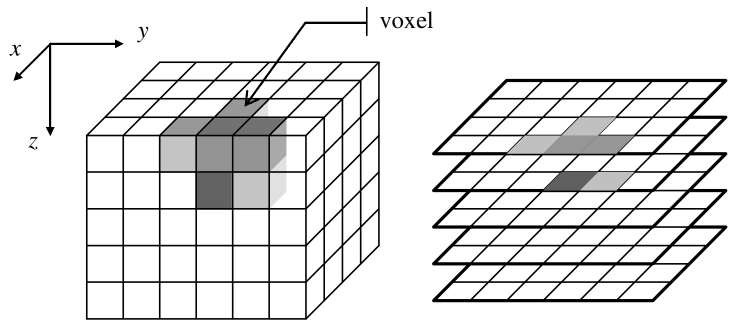


Fig. 1.1. 3D and stack representation of volumetric data.

Exceptional effort must be spent on data processing, representation and coding of volumetric images. In fact, the complexity of such data is such that the adoption of digital processing methods is compulsory for image interpretation and storage. Volumetric images usually present huge quantities of information that must be efficiently stored. Then, since the experimental data generally represent complex scenes and are affected by noise, they must be processed and segmented accordingly. Finally, the volumetric image representation itself is not a trivial task and the research pulls towards faster and more accurate rendering algorithms. These aspects will be discussed in Sections 1.2 and 1.4.

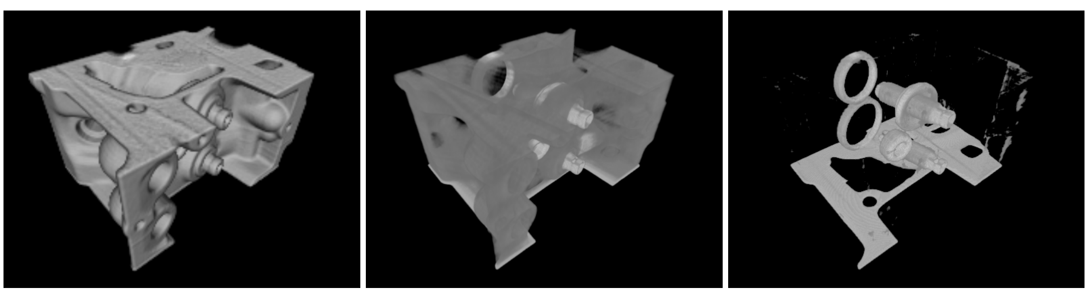


Fig. 1.2. Volume rendering of the engine CT scan obtained by varying the scalar opacity mapping function in order to show all volumetric information.

1.2. Processing and Representation of Volumetric Data

Volumetric image are regularly used for diagnosis, non destructive testing and, more generally, object analysis. Image processing is then essential for information extraction and measurement. Several methods can be found in the literature [CARA, WATT] and are widely used for segmentation, pattern detection and analysis. Most of them are based on 3D versions of the watershed transform, region growing, level sets, split and merge, quadtrees, model fitting, and other algorithms. It must be said that volumetric data processing generally requires user intervention. In fact, due to data complexity and the need for high accuracy, fully automatic recognition is usually not possible.

The representation of volumetric image data can be performed in several ways. The easiest method is to provide orthogonal projections of the selected scene and allow for slice selection (Fig. 1.3a). This method is equivalent to viewing single images and has the advantage of showing local details. On the other hand, it is not capable of offering a comprehensive representation. For such reasons, slice views are mainly used for image analysis and user-guided segmentation.

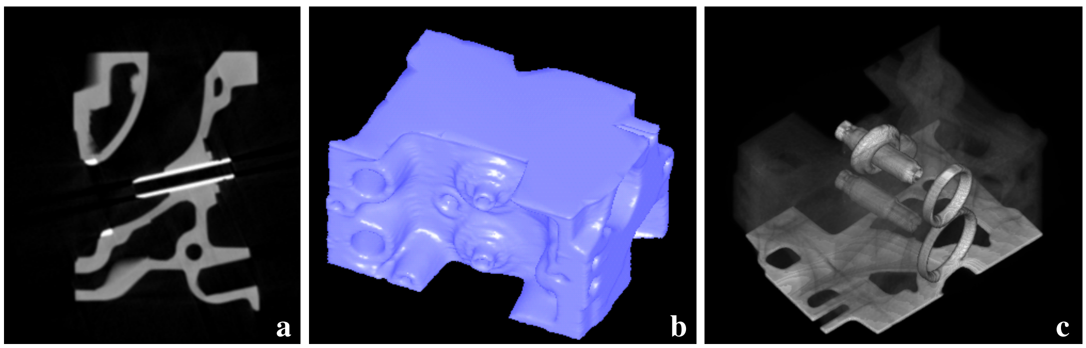


Fig. 1.3. Several representations of the engine volume; from left: planar section, 3D polygonal mesh rendering, volumetric rendering.

Another technique consists in generating a 3D polygonal mesh from some useful features. The resulting model can be manipulated with any CAD application and visualized through a 3D surface rendering algorithm. The result of such modality depends on the accuracy of the process for extracting the interesting features. In fact, the volumetric data must be first preprocessed and segmented in order to get the useful representation. In the example of Fig. 1.3b, for instance, the image has been first binarized with a user-defined threshold; the object contours have been computed for each slice and 3D interpolated in order to produce the 3D mesh. Another simple segmentation method consists in extracting the object *isosurfaces*, *i.e.* the 3D surfaces defined by voxels having similar value. The mesh approach has the advantage of creating a very simple and convenient representation of the selected data, since 3D vector data can be easily stored, manipulated, rendered and distributed. On the other hand, it provides only a partial representation of the original data and the conversion to surface data can sometimes be quite complex and may require major user intervention.

Volumetric rendering is the final solution for volume data representation. It is a computer graphic technique where the object of interest is sampled into many cubic building blocks. Similarly to 3D vector rendering, the objects are displayed by projecting the scene into a 2D pixel space (visual plane – VPN), given the observer's position (projection reference point – PRP) and light sources (LS), as shown in Fig. 1.4. Specific characteristics, such as material, color and transparency, can be assigned to each voxel and each VPN pixel is obtained as voxel integration along the projection path. An advantage of volume rendering is that the 3D volume can be displayed without any a-priori knowledge of the dataset. Moreover, because the entire dataset is preserved, any part, including internal details, can be viewed. In Fig. 1.3c the engine dataset is displayed by setting high transparency to the engine block in order to reveal inner structures. As a drawback, volumetric rendering involves very large memory utilization and requires significant computational power.

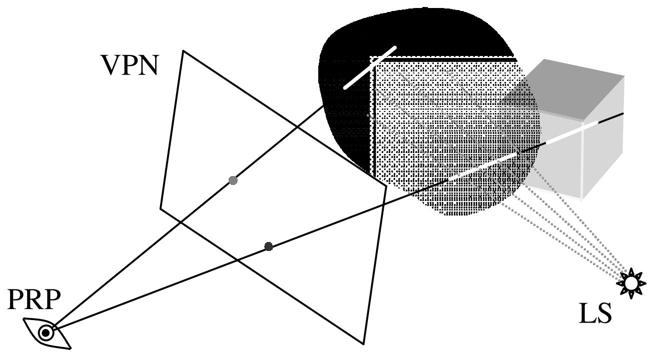


Fig. 1.4. Scheme for volumetric rendering.

1.3. Coding basics

The main objective of data compression is to simplify the management of information that would require an excessive storage capacity with its natural representation. The theory of data and image compression originates from the concepts of statistical and temporal redundancy and irrelevance. In fact, exploitation of the previously mentioned properties allows for the definition and implementation of all coding frameworks.

1.3.1. Symbol Sources

Information theory traditionally deals with symbol sources that output symbols belonging to a finite, predefined alphabet A. For instance, an alphabet can consist of all upper-case characters ($A = \{'A', 'B', 'C', \ldots, 'Z'\}$, all byte values ($A = \{0,1,2,\ldots,255\}$) or both binary digits ($A = \{0,1\}$).

When reading symbols from a symbol source, there is some probability for each of the symbols to appear. For totally random sources each symbol is equally likely, but random sources are also incompressible, and are not interesting for the purpose of this work. Probabilities give a means of defining the concept of symbol self-information, *i.e.* the amount of information a symbol carries.

Simply, the more probable an event is, the less bits of information it contains. If we denote the probability of a symbol $A[i]$ occurring as $p(A[i])$, the expression $\log_2(p(A[i]))$ gives the amount of information in bits that the source symbol $A[i]$ carries.

1.3.2. Message Entropy

The observation that the more probable a symbol is, the less information it carries raises the question of what is the information contents of a specific message, defined as the combination of several symbols. This brings to the concept of the entropy of a source. The measure of entropy gives the amount of information in a message and is calculated as:

$$H = -\sum_{i=0}^{N} p(A[i]) \cdot \log_2 (p(A[i])). \tag{1.1}$$

The entropy of a message is a convenient measure of information, because it sets the lower limit for the average codeword length for a block-variable code, for example Huffman code. Better compression may be achieved with a statistical compression method which only considers single-symbol probabilities. The average codeword length is calculated in an analogous way to the entropy, as:

$$L = \sum -l(i) \cdot \log_2 (p(A[i])), \tag{1.2}$$

where $l(i)$ is the codeword length for the i^{th} symbol in the alphabet. The difference between L and H gives an indication about the efficiency of a code. Smaller difference means more efficient code.

It is no coincidence that the entropy and average code length are calculated using very similar equations. If the symbol probabilities are not equal, a shorter overall message can be produced, *i.e.* shorter average codeword length or better compression, if shorter codes are assigned to symbols that are more likely to occur. It must be noticed that entropy is only the lower limit for statistical compression systems. Other methods may perform better, although not for all sources.

1.3.3. Codes

A code is any mapping from an input alphabet to an output alphabet. A code can be, for instance, $\{a, b, c\} = \{0, 1, 00\}$, but this code is obviously not uniquely decodable. If the decoder gets a code message of two zeros, there is no way it can know whether the original message had two a or a c.

A code is instantaneous if each codeword (a code symbol as opposed to source symbol) in a message can be decoded as soon as it is received. The binary code $\{a, b\} = \{0, 01\}$ is uniquely decodable, but is not instantaneous. The binary code $\{a, b, c\} = \{0, 10, 11\}$ on the other hand is an instantaneous code.

1.3.4. Compression of images

Compressing an image is significantly different than compressing raw binary data. In this case, general purpose compression algorithms can still be used, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This also means that lossy compression techniques can be used in this area.

Lossless compression involves with compressing data which, when decompressed, will be an exact replica of the original data. This is generally the case of binary data such as executables, documents etc. They need to be exactly reproduced when decompressed. On the other hand, pictorial images, and music too, need not be reproduced 'exactly'. An approximation of the original image is enough for most purposes, as long as the error between the original and the compressed image is tolerable. In the case of lossy compression, in fact, several properties of the human vision have been studied and are considered for discarding of image details, *i.e.* information, that result invisible or less significant to the human observer. Fig. 1.5 represents a simplified scheme for image compression, constituted by image transform, coefficients quantization and entropy coder.

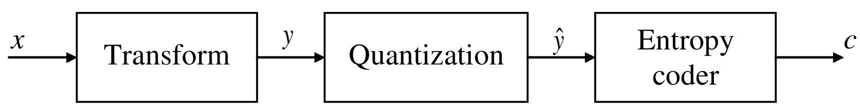


Fig. 1.5. Generic scheme for image compression.

1.3.5. Image classification and transform

An image is represented as a two-dimensional array of coefficients, each coefficient corresponding to the brightness level in that point. Most natural images have smooth color variations, with the fine details being represented as sharp edges in between the smooth variations. The smooth variations in color can be termed as low frequency variations and the sharp variations as high frequency variations.

The low frequency components constitute the base of an image, and the high frequency components (the edges which give the detail) add upon them to refine the image, thereby giving a detailed image. Hence, the smooth variations are demanding more importance than the details.

Separating the high and low frequency components of the image can be done in many ways. A number of spatial-frequency transforms have been developed in order to perform such task. The most interesting are probably the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT). Although inherently different, both are used to study the image properties in the transform domain, where the most energetic components, *i.e.* the most significant coefficients, are concentrated and separated from the detail information (Fig. 1.6).

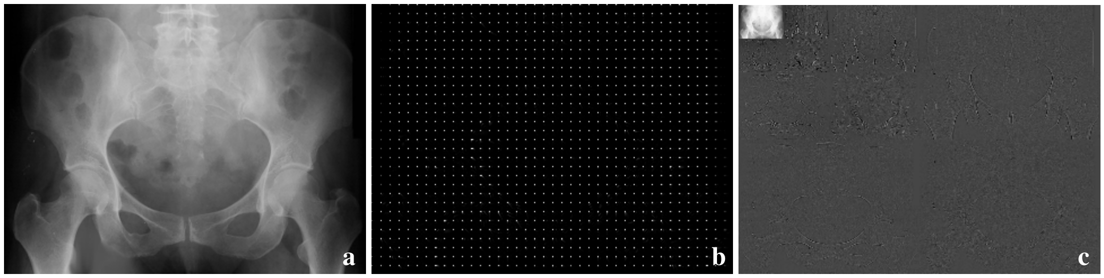


Fig. 1.6. Example of image transforms; original image (a), DCT (b) and 3-levels DWT (c).

1.3.6. Quantization

Quantization refers to the process of approximating the continuous set of values in the image data with a finite and preferably small set of values. The input to a quantizer is the original data, and the output is always one among a finite number of levels. The quantizer is a function whose set of output values are discrete, and usually finite. Obviously, this is a process of approximation, and a good quantizer is one which represents the original signal with minimum loss or distortion. Because of the intrinsic effect of approximating, thus altering, the original data, quantization must not be used when lossless compression is required.

There are two types of quantization: scalar and vector quantization. In scalar quantization, each input symbol is treated separately in producing the output, while in vector quantization the input symbols are grouped together in structures called vectors, and processed to give the output. The grouping of data and treating them as a single unit generally increases the optimality of the vector quantizer, but at the cost of increased computational complexity.

Back to scalar quantization, a quantizer can be specified by its input partitions and output levels. If the input range is divided into levels of equal spacing, then the quantizer is termed as Uniform, otherwise Non-Uniform Quantizer. A uniform quantizer can be easily specified by its lower bound and the step size. Also, implementing a uniform quantizer is easier than a non-uniform quantizer. In Fig. 1.7 is shown a uniform quantizer. If the input falls between $n \cdot r$ and $(n+1) \cdot r$, the quantizer outputs the symbol n.

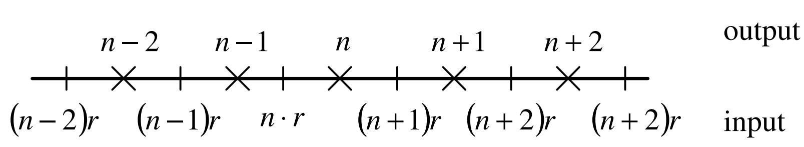


Fig. 1.7. Uniform scalar quantizer.

1.3.7. Bit allocation

The first step in compressing an image is to segregate the image data into different classes. Depending on the importance of the data it contains, each class is allocated a portion of the total bit budget, such that the compressed image has the minimum possible distortion. This procedure is called Bit Allocation.

The Rate-Distortion theory is often used for solving the problem of allocating bits to a set of classes, or for bitrate control in general. The theory aims at reducing the distortion for a given target bitrate, by optimally allocating bits to the various classes of data. One approach to solve the problem of Optimal Bit Allocation using the Rate-Distortion theory is given in [Ris91] and is explained below:

1. Initially, all classes are allocated a predefined maximum number of bits.
2. For each class, one bit is reduced from its quota of allocated bits, and the distortion due to the reduction of that 1 bit is calculated.
3. Of all the classes, the class with minimum distortion for a reduction of 1 bit is noted, and 1 bit is reduced from its quota of bits.
4. The total distortion for all classes D is calculated.
5. The total rate for all the classes is calculated as $R = p(i) \cdot B(i)$, where p is the probability and B is the bit allocation for each class.
6. Compare the target rate and distortion specifications with the values obtained above. If not optimal, go to step 2.

An alternate approach which is also mentioned in [Ris91] is to initially start with zero bits allocated for all classes, and to find the class which is most 'benefited' by getting an additional bit. The 'benefit' of a class is defined as the decrease in distortion for that class. The benefit of a bit is a decreasing function of the number of bits allocated previously to the same class. Both approaches mentioned above can be used to the Bit Allocation problem.

1.3.8. Entropy coding

After the data has been quantized into a finite set of values, it can be encoded using an Entropy Coder to give additional compression. By entropy, we mean the amount of information present in the data, and an entropy coder encodes the given set of symbols with the minimum number of bits required to represent them. Two among the most popular entropy coding schemes are Huffman coding and Arithmetic coding.

1.3.9. Error Metrics

There exist several error metrics for evaluating image coding algorithm performance. Among these, the error metrics generally used to compare the various image compression techniques are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error. The mathematical formulae are:

$$
\begin{aligned}
MSE &= \frac{1}{N \cdot M} \sum_{i=0}^{N-1} \sum_{j+0}^{M-1} \left[I(x, y) - \hat{I}(x, y) \right]^2 \\
PSNR &= 20 \cdot \log_{10} \left(\frac{255}{\sqrt{MSE}} \right)
\end{aligned}
\tag{1.3}
$$

where $I(x, y)$ is the original image, $\hat{I}(x, y)$ is the approximated version, *i.e.* the co-decompressed image, and M, N are the image dimensions. A lower value for MSE means lesser error, and, as seen from the inverse relation between the MSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of the ratio of Signal to Noise represents a better approximation, where the 'signal' is the original image, and the 'noise' is the error in reconstruction. Although MSE and PSNR provide a fast objective measurement of the quality of approximated signals, they may not be completely reliable when dealing with images. In fact, the process of human vision implies a number of non-linear masking effects that deeply affect the subjective evaluation of compressed images. These considerations are the basis for Human Visual System (HVS) metrics.

1.4. Coding of Volumetric Data

The compression of volumetric data is even more relevant than that of conventional images, since the former generally require huge storage capacities. For instance, an average raw CT dataset made of 100 planar sections with a spatial resolution of 512×512 pixels and 12bits bit-depth requires roughly 50Mb for storage. Since a typical radiology department may produce dozens of such exams daily, it is easy to imagine that any storage capacity can be easily saturated in a relatively short time.

Although volumetric image data can be seen as the extension of conventional images, the methodologies for coding such data cannot follow the same line of thought. In fact, due to the peculiarity of such images and their adoption in diagnosis, treatment planning and scientific applications, special attention should be spent when designing a new compression scheme. Volumetric data present important structural peculiarities to be taken into account when defining the new methodology. Tomographic devices, for instance, generate a collection of planar sections that is generally non-isotropic, *i.e.* the resolution along intra-plane directions is different than that along inter-slice direction. A typical example is computer tomography, where planar resolution is generally finer than 1mm while inter-slice resolution may be of several mms. In this case, the compression algorithm will not be able to detect and exploit similar statistical redundancy along different directions. In Fig. 1.8 is shown an example of brain MRI scan where the resolution along the inter-slice axis is almost six times that along planar directions; the sagittal and coronal views remarkably show the effects of under-sampling.

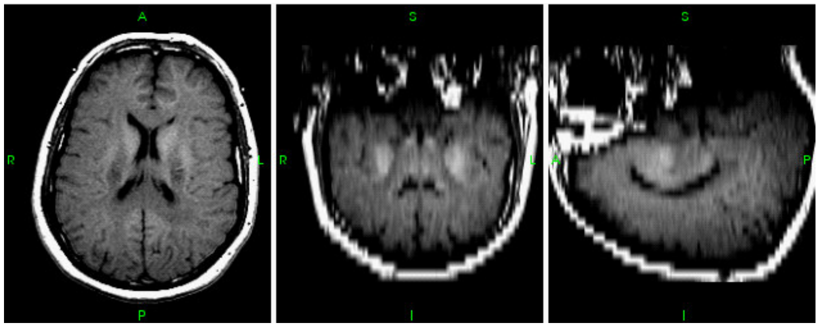


Fig. 1.8. Example of brain MRI showing different resolutions along different directions.

Hyperspectral imaging (Section 1.5.1) is another example that requires particular care, since each plane represents exactly the same scene recorded at a different wavelength. This means that adjacent planes may show the same structural information with very different values. In this case, a preprocessing stage for data linearization is generally required.

The specific application where volumetric images are required is another key aspect. As will be discussed in the following paragraph, diagnostic or scientific purposes generally involve lossless compression. However, new visually-lossless or high performing compression techniques should be better evaluated for such applications. Moreover, lossy compression may be required for fast representation or in transmission critical environments, such as telemedicine frameworks based on mobile devices (Fig 1.9).

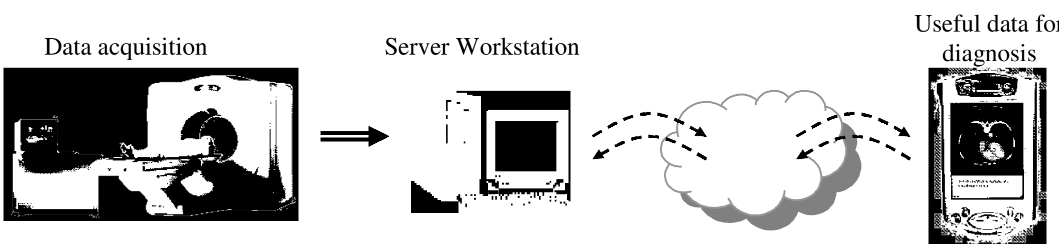


Fig. 1.9. Mobile telemedicine framework.

1.4.1. Lossy and Lossless Coding

The distinction between lossy and lossless compression has been defined in Section 1.3.4. The only compression modality for volumetric images that is free from any legal issue is, in fact, lossless or reversible coding. Such methods are principally based on the exploitation of statistical redundancy in the information stream with the minimization of entropy. Lossless compression may be generally represented by the scheme in Fig. 1.5 without the quantization block. An important limitation of lossless methods is that the resulting compression ratios are generally low, in the range between 1.5 and 5. An overview of reversible compression techniques applied to medical images can be found in [Kud+92] and [Wuo+95].

Lossy techniques, on the other hand, result in image degradation, which can be more or less visible. Several works addressed the problem of designing a *visually-lossless* or *near-*

lossless compression framework, *i.e.* lossy compression schemes capable of producing a coded image that is indistinguishable from the original for the human observer. Examples of such techniques can be found in [Wu+03], [Aia+99] and [Tes+96]. Since such definitions generally refer to subjective evaluations, their adoption for scientific and diagnostic purposes is not completely accepted. Radiologists, for instance, are generally skeptical on lossy techniques, even though current state of the art coders are really capable of high quality compression. Moreover, it must be said that actual output data cannot be considered as plain raw data, since they are generally the result of some preprocessing that is applied at the sensor source. In fact, it is rarely considered that the original data themselves are often affected by system artifacts. Even analogical supports, like films, are subjected to visual artifacts, such as time averaging due to exposition duration or blurring due to slight movements.

Another approach for data demanding high coding quality is that of *lossy to lossless* compression. As will be shown in Chapter 2, the algorithms that follow such paradigm allow for information scalability. The output data stream of a lossy to lossless coder is organized in such a way that it can be truncated at any or a number of specific points. Then, the resulting image will be reconstructed up to a certain quality level, proportional to the truncation point. In addition to the concept of plain scalability, lossy to lossless coders are capable of fully reversible compression if no truncation is performed. Such features are particularly useful for volumetric data. Lossy to lossless coding, for instance, consents to find a compromise between data accuracy and storage size or transmission bandwidth requirements. Moreover, scalability allows for the same data to be decoded in different ways (*code once, decode many times* approach), depending on the specific application or transmission requirements.

It is very important to notice that a standard metric for defining and measuring the visual quality of compressed volumetric images has not been yet adopted. JPEG2000 [J2Kmed], for instance, proposed a special implementation for biomedical applications, but its suitability has not been universally accepted in the field. Then, much effort should be spent in the design and evaluation of an efficient and representative quality metric. By the time this thesis work is being written, an activity on quality metrics for measuring the reliability of lossy medical image compression systems is in progress [Mas+04]. The proposed method will be based on a HVS-oriented preprocessing stage and several factors for measuring both coder-dependent artifacts and the effects of compression on the results of common medical processing algorithms.

1.4.2. Advanced Coding Features

In recent years, image coders have been enriched with a number of advanced features for better representation and data processing. As mentioned in Section 1.4.1, a scalable bit-stream is organized in such a way that the image data can be truncated and reconstructed up to certain quality or bitrate levels. Scalability allows for compression rate customization, depending on the specific application and transmission requests. Almost all modern coding frameworks, like JPEG2000 [J2K], SPIHT [Sai+J96] and SPECK [Isl+99], implement scalability. In Fig. 1.10 the fibroadenoma mammography has been coded with JPEG2000; from left to right the images were compressed to 563, 729, 1182 and 19857 bytes respectively. In Fig. 1.11 another example of bitrate scalability is provided. In this case a tomographic slice has been coded with the basic SPIHT algorithm at lossless, 0.5bpp and

0.125bpp rate respectively. The images on the bottom illustrate the increasing approximation introduced by wavelet coefficient quantization.

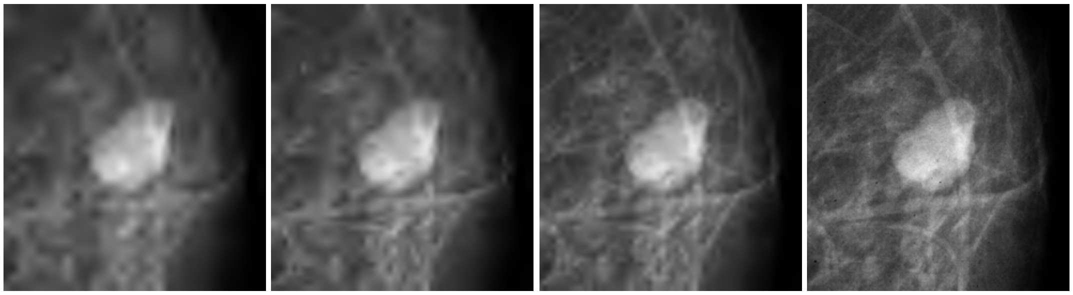


Fig 1.10. Example of bitrate scalability.

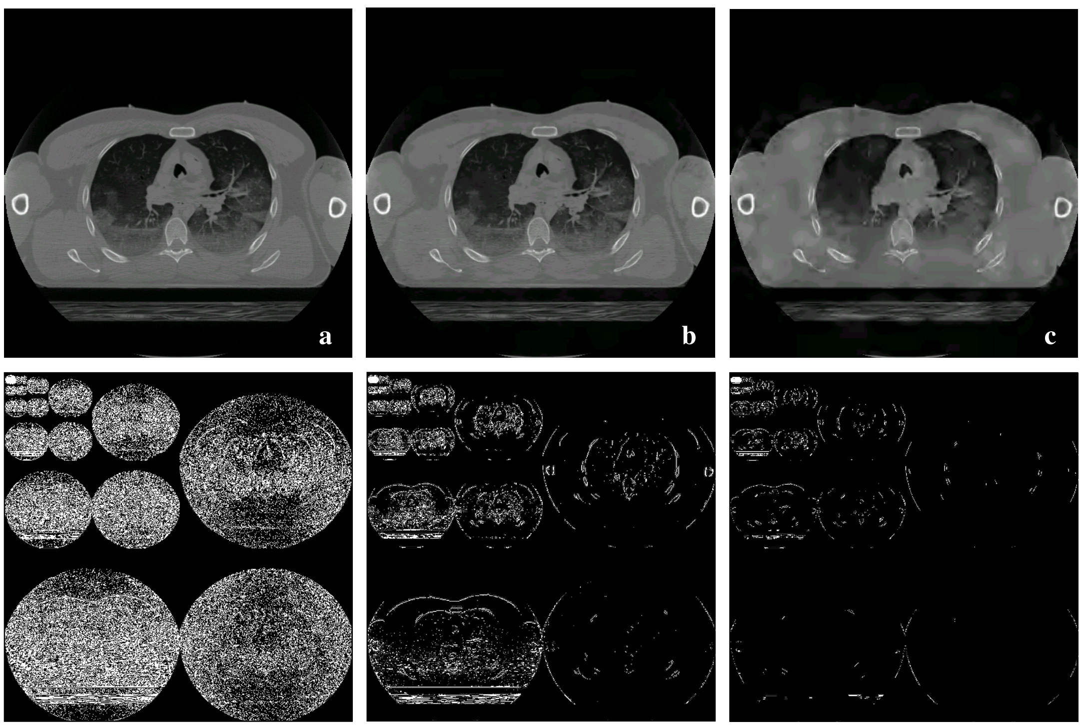


Fig. 1.11. Chest (slice) coded with SPIHT algorithm using 5 levels integer lifting wavelet decomposition (5/3 filter kernel). On top is the reconstructed image, down is the wavelet coefficients after decoding; a) lossless (2.78bpp) b) 0.5bpp (38dB) c) 0.125bpp (29dB).

Region of Interest (ROI) coding is another interesting coding feature. It is particularly useful when the scene, *i.e.* the image or volume, presents regions with different semantic significance. The basic idea is to offer support for an object-oriented image coding technique, which starts from the scene segmentation into interesting objects and background. A different set of coding parameters is then chosen for each of the acquired regions. A typical choice is to assign a different priority level to ROIs and background.

While the segmentation task is out of the scope of this discussion, several techniques may be adopted, mostly requiring user interaction. Object prioritization may be performed in different ways. The *max-shift* approach is probably the most used. It consists in up-shifting all transform coefficients that belong to the object region by a value, U, while the background samples have zero up-shift. The value of U is selected to be sufficiently large so that the background and foreground samples can be distinguished on the base of the decoded quantization indices alone. This method is rather efficient and does not require the transmission of the object shape information. Moreover, it enables support for arbitrary shaped ROI with minimal complexity by defining different up-shift values, U_i, depending on the desired priority level. Another prioritization technique exploits the scalability properties of modern coders. *Block-based* approaches consist in sequencing the compressed data information so that the coefficients belonging to the object region are transmitted first. An example of ROI coding is shown in Fig. 1.12; the satellite image of New York City has been coded with JPEG2000 and rectangular ROI centered over Manhattan with 50, 65, 80 and 95 quality levels from left to right respectively. The detail windows show the regions belonging to the background (back) and to the ROI (obj); the actual ROI is represented in dashed line. It is easy to see that the background quality is sacrificed for the object representation so that the object quality appears unaltered for increasing coding quality, while the background varies conspicuously.

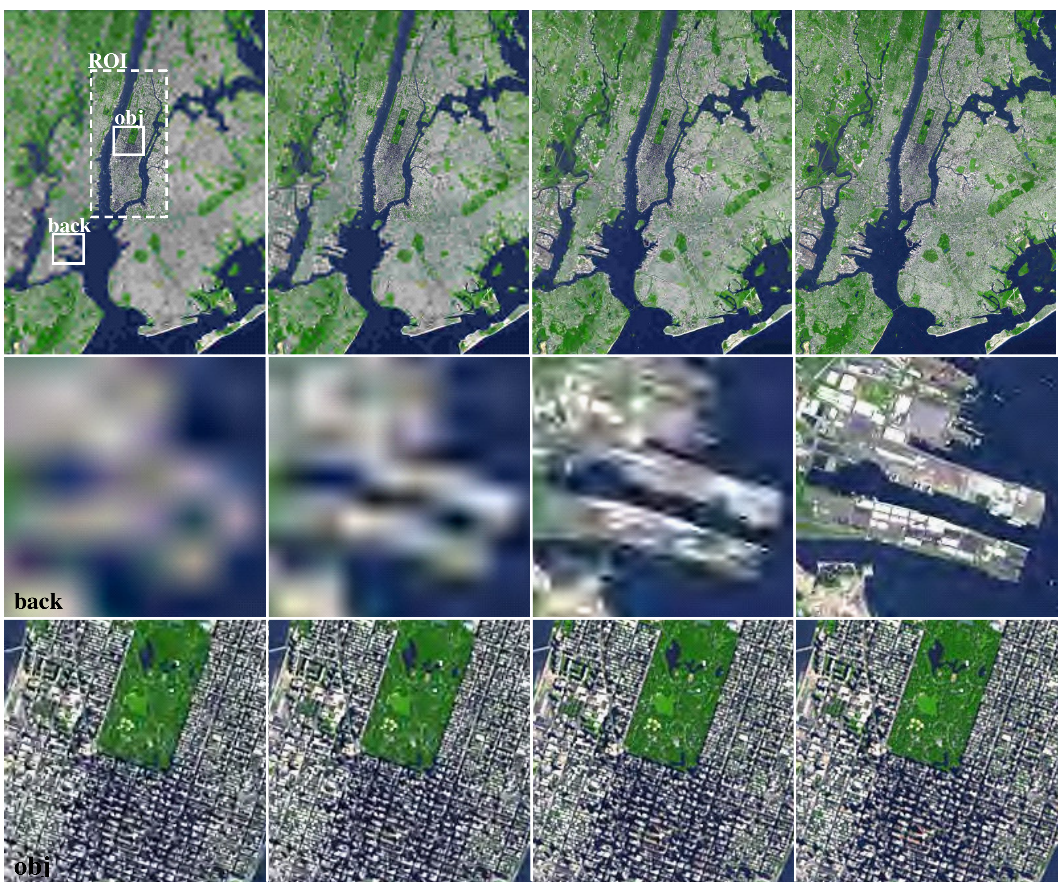


Fig. 1.12. ROI coding example.

1.5 Applications

Although a number of scientific applications make extensive use of volumetric imaging, the medical field is certainly the most important recipient of such technique. In fact, medical diagnosis methodologies often require the collection of volumetric data in order to image entire body sections and analyze the organs conditions [SUET]. At the same time, recent technical advances have made possible the development of several subsurface remote sensing techniques that also rely on volumetric data. An overview of several volumetric imaging techniques is given in the following.

1.5.1. Hyperspectral Imaging

Multispectral remote sensors such as the Landsat Thematic Mapper and SPOT XS produce images with a few relatively broad wavelength bands. Hyperspectral remote sensors, on the other hand, collect image data simultaneously in dozens or hundreds of narrow, adjacent spectral bands. These measurements make it possible to derive a continuous spectrum for each image cell, as shown in the illustration below. After adjustments for sensor, atmospheric, and terrain effects are applied, these image spectra can be compared with field or laboratory reflectance spectra in order to recognize and map surface materials such as particular types of vegetation or diagnostic minerals associated with ore deposits. Hyperspectral images contain a wealth of data, but interpreting them requires an understanding of exactly what properties of ground materials we are trying to measure, and how they relate to the measurements actually made by the hyperspectral sensor.

The volumetric data produced with hyperspectral sensors may be considered as a special case of volumetric images, since each planar section represents exactly the same scene recorded at a different wavelength.

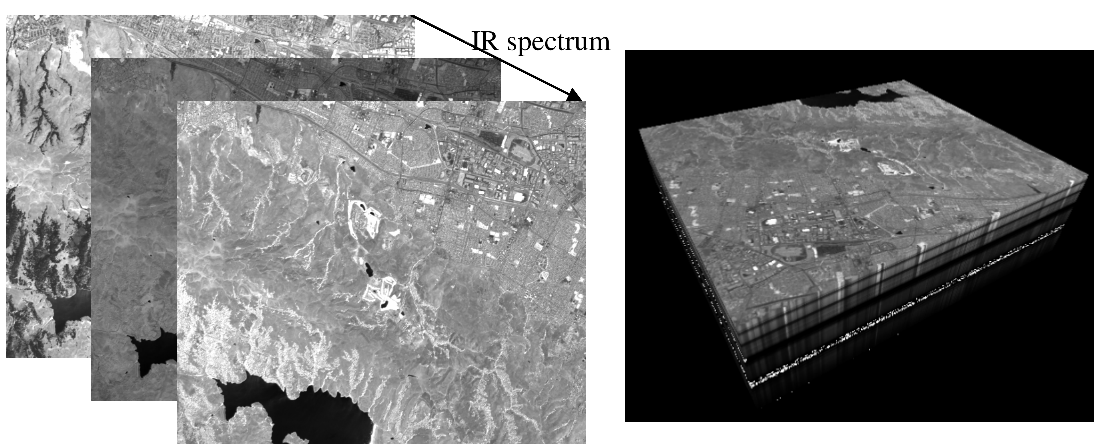


Fig. 1.13. Aerial hyperspectral data from the AVIRIS database [AVI].

1.5.2. Ground Penetrating Radar

Ground penetrating radar (GPR for short) is a noninvasive electromagnetic geophysical technique for subsurface exploration, characterization and monitoring. It is widely used in locating lost utilities, environmental site characterization and monitoring, agriculture, archaeological and forensic investigation, unexploded ordnance and land mine detection, groundwater, pavement and infrastructure characterization, mining, ice sounding, permafrost, void, cave and tunnel detection, and other applications.

GPR uses electromagnetic wave propagation and scattering to image, locate and quantitatively identify changes in electrical and magnetic properties in the ground. It is similar to ordinary radar, except that it uses lower frequencies, 25MHz to a few GHz, is extremely wide band and the antennas transmit directly down. It may be performed from the surface of the earth, in a borehole or between boreholes, from aircraft or satellites. It has the highest resolution in subsurface imaging of any geophysical method, approaching centimeters under the right conditions. Depth of Investigation varies from less than a meter to over 5400 meters, depending upon material properties. Typical depths of investigation in fresh-water saturated, clay-free sands are about 30 meters. Detectability of a subsurface feature depends upon contrast in electrical and magnetic properties, and the geometric relationship with the antenna, *i.e.* feature size, shape, and orientation. Quantitative interpretation through modeling can derive from ground penetrating radar data such information as depth, orientation, size and shape of buried objects, density and water content of soils, and more.

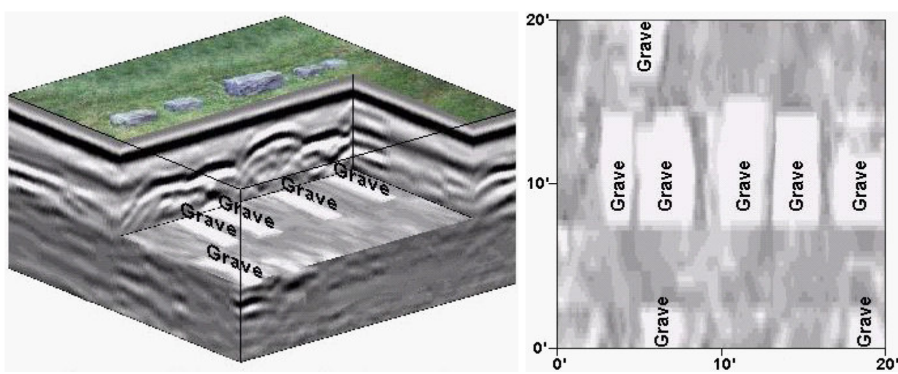


Fig.1.14. Example of GPR used for archaeological site investigation.

1.5.3. Ultrasound Imaging

Ultrasound is an imaging method that uses high-frequency sound waves to produce precise images of structures within the body. During an ultrasound exam, a technician presses a small transducer (Fig. 1.15) against the patient’s skin. The transducer generates high frequency sound waves, which are reflected by the tissue interfaces. At the same time, the transducer is able to receive the reflected sound waves and the resulting pattern of sound reflection is processed by a computer to produce a still or moving image.

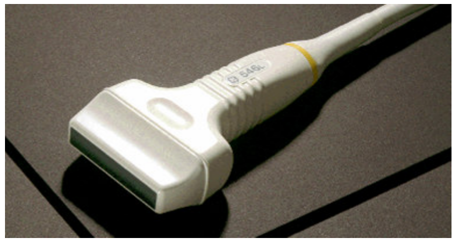


Fig. 1.15. Ultrasonic transducer.

Ultrasound offers several advantages over other imaging techniques. It has a relatively low cost and is a safe procedure. Image modalities include real-time scanning that allows imaging of moving objects, *e.g.* fetal mobility, and the recent introduction of 3D-4D ultrasound (Fig. 1.16). Although ultrasound is a very useful tool, it does have some limitations. Sound doesn' t travel well through air or bone, so ultrasound isn' t effective at imaging areas of your body that have gas in them, or that are obscured by bone. Moreover, the signal is very sensitive to noise and the resulting images are often affected by reconstruction distortions problems, so that images appear noisy and unclear.

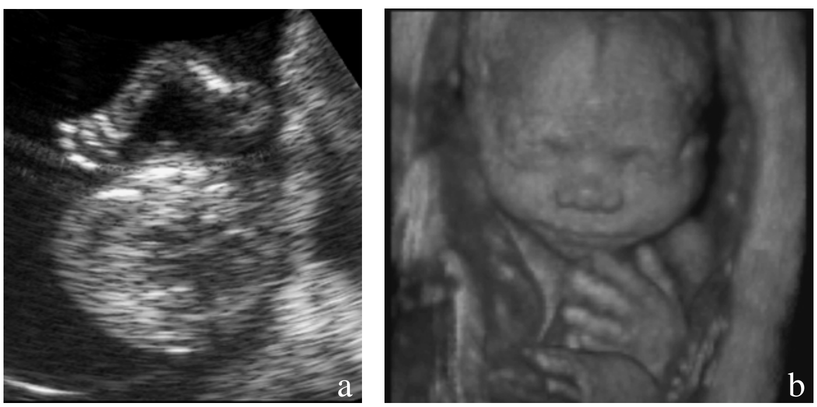


Fig. 1.16. Example of fetal ecography: 2D (a) and 3D-4D (b) reconstruction.

1.5.4. Confocal Microscopy

Confocal microscopy offers several advantages over conventional optical microscopy, including controllable depth of field, the elimination of image degrading out-of-focus information, and the ability to collect serial optical sections from thick specimens. The key to the confocal approach is the use of spatial filtering to eliminate out-of-focus light or flare in specimens that are thicker than the plane of focus (Fig. 1.17). There has been a tremendous explosion in the popularity of confocal microscopy in recent years, due in part to the relative ease with which extremely high-quality images can be obtained from specimens prepared for conventional optical microscopy, and in its great number of applications in many areas of current research interest.

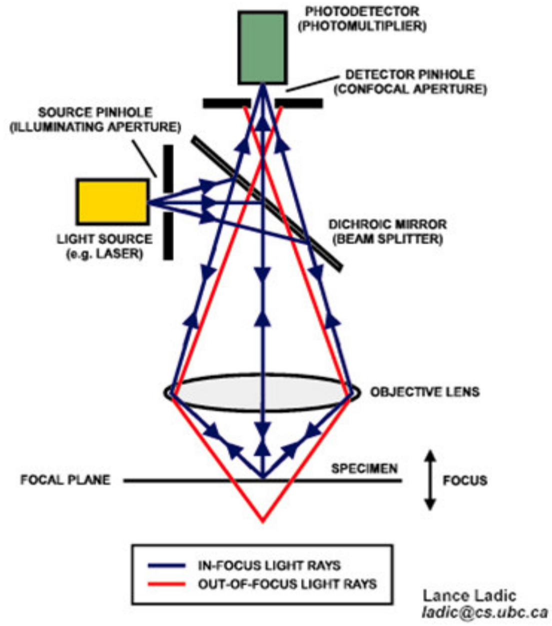


Fig. 1.17. Confocal microscope setup.

A number of different imaging modes are used in the application of confocal microscopy to a vast variety of specimen types. They all rely on the ability of the technique to produce high-resolution images, termed optical sections, in sequence through relatively thick sections or whole-mount specimens. Based on the optical section as the basic image unit, data can be collected from fixed and stained specimens in single, double, triple, or multiple-wavelength illumination modes, and the images collected with the various illumination and labeling strategies will be in register with each other. Live cell imaging and time-lapse sequences are possible, and digital image processing methods applied to sequences of images allow z-series and three-dimensional representation of specimens, as well as the time-sequence presentation of 3D data as four-dimensional imaging. Reflected light imaging was the mode used in early confocal instruments, but any of the transmitted light imaging modes commonly employed in microscopy can be utilized in the laser scanning confocal microscope. An example of confocal data is shown in Fig. 1.18. The two images are obtained by focusing a corn grain at different plane depths.

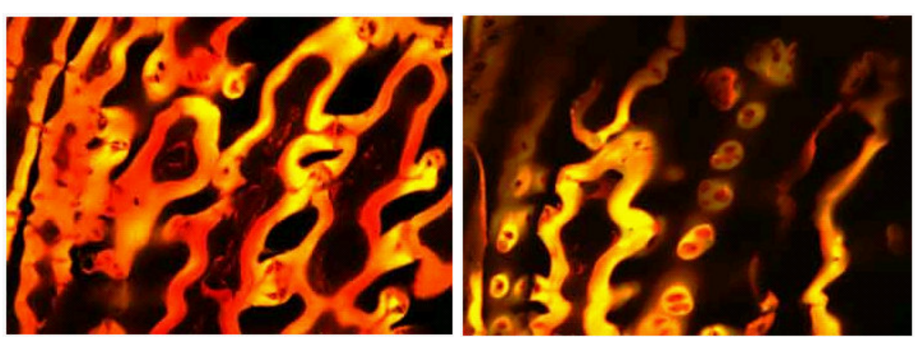


Fig. 1.18. Two z-axis confocal microscopy scans from a corn grain.

1.5.5. Computer Tomography

Thanks to the technical improvement and the significant drop of their cost, CT devices are possibly the most popular devices for image-based diagnosis. The output from a CT device is a collection of trans-axial matrices, *i.e.* slices, which are perpendicularly aligned to the axis defined by the patient's spinal chord. Each slice represents one section of the patient's body with a thickness between 1 and 10 mm. The working scheme for a typical closed-magnet CT device is shown in Fig. 1.19.

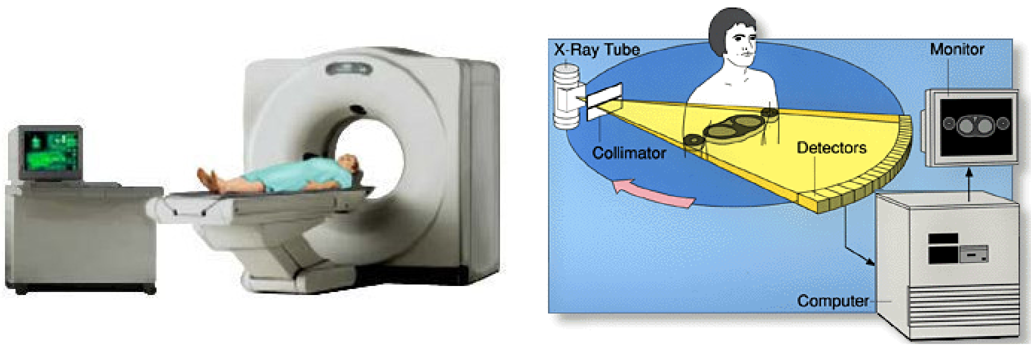


Fig.1.19. Example of CT imager and its working principle.

Schematically, the acquisition is performed by rotating an X-ray source and opposite array of X-ray detectors around the patient's body. Each detector records the transmitted X-ray energy so that an absorption measure can be produced along each projection ray. After a sufficient number of absorption measurements is computed (generally a minimum rotation of 180 degrees), a filtered backprojection algorithm is applied to the projection data (synogram) in order to reconstruct the slice image in the spatial domain. By shifting the X-ray source and detectors along the patient's axis, a volume image can be acquired as collection of slices.

Generally, a slice resolution from 64×64 to 1024×1024 pixels can be achieved, while the pixel size may vary between 0.5-2mm and is homogeneous in the trans-axial plane. The number of slices depends on the distance between adjacent acquisition steps and the size of the organ to be imaged. Each pixel defines the radiation absorption characteristics in a small volume of the body. An example of CT data is shown in Fig. 1.20, with the typical axial, sagittal and coronal projections and 3D volumetric rendering. Currently, CT is the only imaging modality for which a standard measure is defined. The Hounsfield Unit, or HU, assumes water as reference material (0 HU) and reaches, for instance, -1000 HU for air and above 200 HU for bone structures.

Fig. 1.20. CT volume data example

Modern CT devices allow for the acquisition of several slices per second, while a full body scan may take up to hours. The radiation dose is comparable with that of radiographic devices. Although 3D reconstructions can be produced with any series of slices, the image fidelity depends on the integrity and resolution of sampled data. Consequently, the acquisition parameters, such as the planar resolution, slice thickness and inter-slice distance, are important factors for reconstruction quality. However, it must be noticed that although it is possible to obtain high-resolution scans and very detailed images, this may require higher radiation doses and longer acquisition times. In particular, longer acquisition times may produce image distortion due to the patient's movement and may induce patient's discomfort because of the claustrophobic environment.

The last generation of CT devices implements both continuous spiral scan and multi-slice detector array acquisition. Such improvements allow achieving ultra fast acquisition times and are particularly indicated for diagnosis of moving objects such as organs while breathing or fetus.

1.5.6. Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is an imaging technique used to produce high quality images of the inside of the human body. MRI is based on the principles of nuclear magnetic resonance (NMR). The technique started out as a tomographic imaging technique, that is it produced an image of the NMR signal in a thin slice through the human body and has advanced to a volume imaging technique. In fact, MRI gets around the impossibility of imaging objects smaller than the wavelength of the energy being used by producing images based on spatial variations in the phase and frequency of the radio frequency energy being absorbed and emitted by the imaged object.

Magnetic resonance imaging is based on the absorption and emission of energy in the radio frequency range of the electromagnetic spectrum. More precisely, MRI produces images based on spatial variations in the phase and frequency of the radio frequency energy being absorbed and emitted by the imaged object. The human body is primarily fat and water, resulting in approximately 63% hydrogen atoms. Since hydrogen nuclei have the highest gyromagnetic ratio, the NMR signal can be obtained by measuring the relaxation time of hydrogen nuclei after a RF pulse excitation (Fig. 1.21). MRI is, in principle, much safer for the patient than X-ray based imaging, since it does not involve the exposition to ionizing radiation.

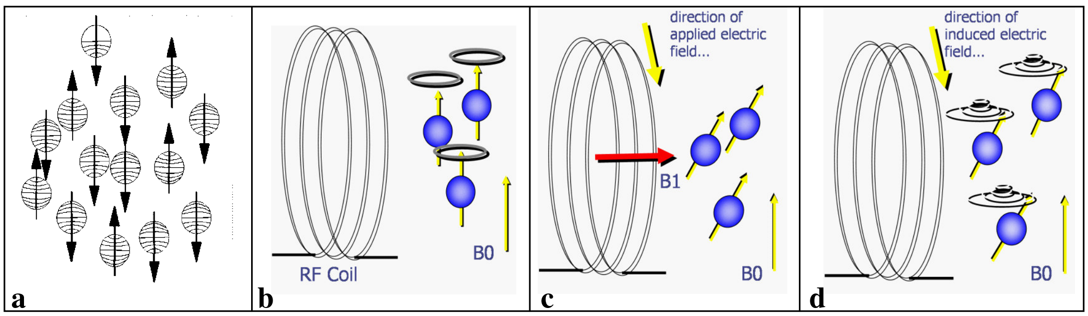


Fig. 1.21. Mechanism for MR imaging. A strong uniform magnetic field, B0, results in the polarization of hydrogen nuclei (a); nuclei start to precess around B0 (b); when a RF pulse is applied, a new field, B1, is induced orthogonally to B0 and the nuclei alignment is rotated accordingly (c); when the RF pulse is turned off, the nuclei return to the starting equilibrium condition (d). The relaxation time is measured and the resulting signal constitutes the base for MR imaging.

An example of MR volumetric data is shown in Fig. 1.22. The image shows the patient's head and brain; it can be noticed that this imaging modality offers produces finer details and is able to better discriminate tissue types than CT imaging.

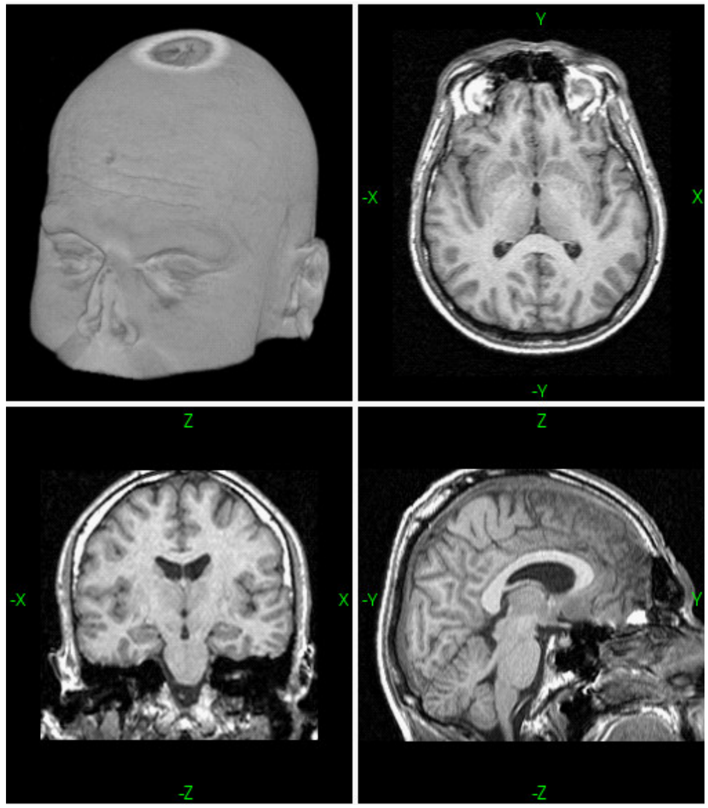


Fig. 1.22. MR volume data example.

The MRI apparatus is composed by four principal components: the main magnet, the gradient coils, the RF coils and RF detectors (antennae). The main magnet is the most expensive component and is generally made of a massive coil of superconducting wire that must be kept at a temperature close to absolute zero (4.2° K) in order to achieve a resistance approximately equal to zero. This requires the adoption of expensive cooling systems based on liquid helium. Once current is caused to flow in the coil it will continue to flow as long as the coil is kept at liquid helium temperatures. Such magnets are able to produce extremely strong fields (1.5T on average, but may reach 9.5T) in order to obtain a strong signal and high quality images.

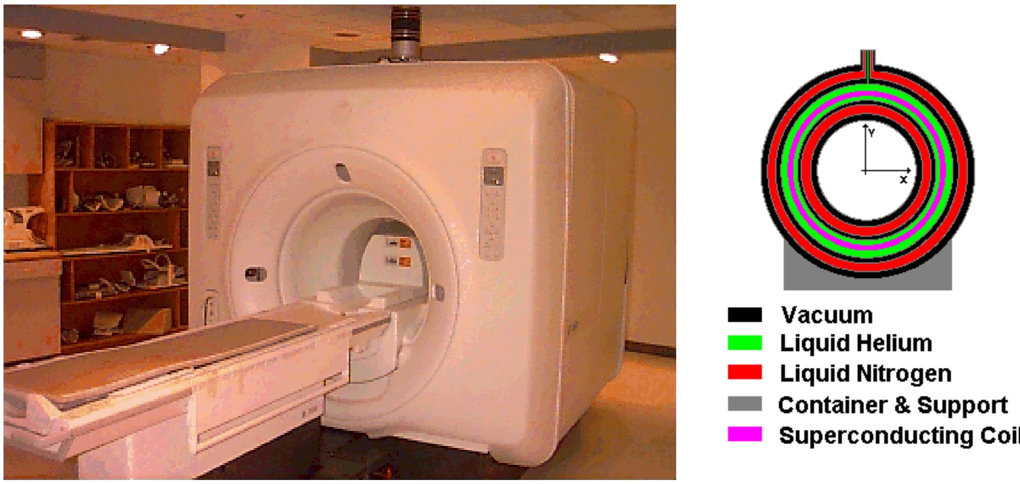


Fig.1.23. Example of MR imager and its cross-section construction scheme.

The gradient coils produce the gradients in the B0 magnetic field (Fig. 1.23). Assuming the standard magnetic resonance coordinate system, a gradient in B0 in the Z direction is achieved with an antihelmholtz type of coil. Current in the two coils flow in opposite directions creating a magnetic field gradient between the two coils. The new fields, B, add to the B0 field at one coil while subtract from the B0 field at the other coil. X and Y gradients are created by a pair of figure-8 coils.

RF coils create the B1 field which rotates the net magnetization in a pulse sequence. They also detect the transverse magnetization as it precesses in the XY plane. The RF coil on an imager can be likened unto the lens on a camera. A good imaging site will have several imaging coils to handle the variety of imaging situations which might arise.

An imaging coil must resonate, or efficiently store energy, at the Larmor frequency. All imaging coils are composed of an inductor, or inductive elements, and a set of capacitive elements. The resonant frequency, ν, of an RF coil is determined by the inductance (L) and capacitance (C) of the inductor capacitor circuit:

$$\nu = \frac{1}{2\pi\sqrt{LC}} . \tag{1.4}$$

Some types of imaging coils need to be tuned for each patient by physically varying a variable capacitor. The other requirement of an imaging coil is that the B1 field must be perpendicular to the B0 magnetic field.

Finally, the quadrature detector is a device which separates out the Mx' and My' signals from the signal from the RF coil. For this reason it can be thought of as a laboratory to rotating frame of reference converter.

1.5.7. *Nuclear Medicine Imaging*

Nuclear medicine imaging refers to all imaging techniques involving the introduction of a radionuclide into the body via injection in liquid or aggregate form or inhalation in gaseous form. The radionuclide introduced into the body is often chemically bound to a complex that acts characteristically within the body; this is known as a tracer. It results that any increased physiological function will usually mean increased concentration of the tracer. Hundreds of medical radionuclides have been implemented in order to detect different metabolic phenomena. For example, the ligand methylene-diphosphonate (MDP) can be preferentially taken up by bone. By chemically attaching technetium-99m to MDP, radioactivity can be transported and attached to bone for imaging. In this way, specific functional processes can be observed and measured. The radiation emitted from the radionuclide inside the body is usually detected using a gamma-camera. Traditionally, gamma-cameras have consisted of a gamma-ray detector, such as a single large sodium iodide scintillation crystal, coupled with an imaging sub-system such as an array of photo-multiplier tubes and associated electronics. Then, one significant difference between NMR and CT is that for NMR the signal source is within the patient placed in an unknown position, while in CT the X-ray beam is directed from the outside and detected after transmission.

The most diffuse devices for nuclear imaging are SPECT (Single Photon Emission Computed Tomography) and PET (Positron Emission Tomography). SPECT extends the use of gamma-cameras to tomographic acquisition by rotating up to three detectors around the patient’s body. PET consists in a ring of detectors that identify the photon pair generated by the positron-electron annihilation traveling in opposite directions (Fig. 1.24).

Both devices allow for the reconstruction of a volumetric data. In general, the resulting image has a low resolution, since the detection of gamma rays implies counting accumulation and the detectors don’t allow for great spatial precision. Moreover, images appear noisy since the strength of the detected signal is significantly lower than that for CT.

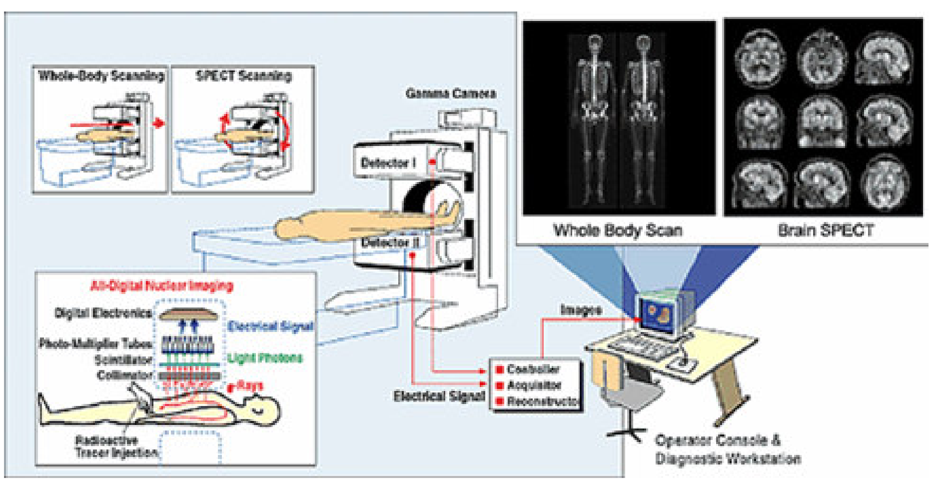


Fig. 1.24. Scheme of gamma camera application: whole body scan and SPECT.

1.1.8. Other applications

Volumetric images are adopted in a number of other scientific fields. These include: astronomy, *e.g.* stellar maps, meteorology, *e.g.* current flows, temperature or pressure distributions, geology, *e.g.* terrain composition and stability, and fluid dynamic (Fig. 1.25).

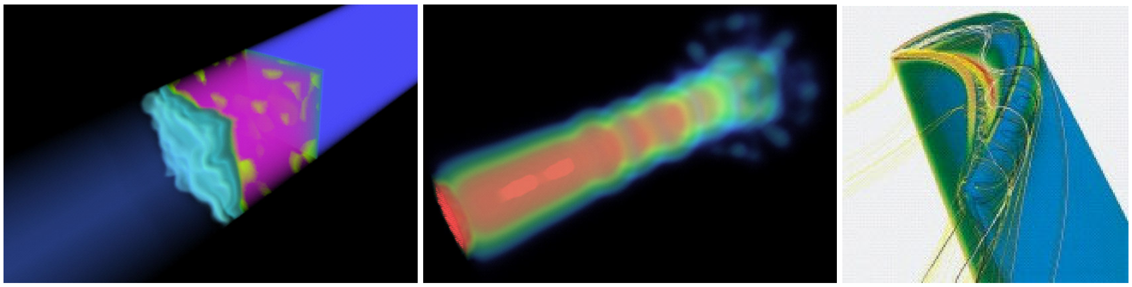


Fig. 1.25. Examples of volume data from fluid dynamics simulations

Chapter 2

Lossy to Lossless Coding of Volumetric Data

2.1. Introduction

Volumetric imagery finds many applications in the scientific and medical fields. Many diagnostic devices, such as computer-aided tomography (CT), magnetic resonance (MR) or positron emission tomography (PET) produce volume data as natural output. Such image modality offers several advantages for visualization, processing and diagnostic purpose, but has the drawback of critical storage requirements. Moreover, thanks to the technological progress, last generation sensors offer ever increasing spatial and intensity resolutions, resulting in the growth of storage space requirement and processing complexity.

For such reasons, volumetric image compression is highly desirable. Several techniques have been investigated in the literature. Given the nature of the data and all possible applications, a coding scheme should offer various features; progressive data transmission, or scalability, is the property that allows for obtaining low-resolution or low-quality versions of a requested image and progressively refine the representation through use of additional data. Such feature is mostly helpful in the case of lossy-to-lossless compression allowing to choose the most convenient representation for any given application requirement or hardware configuration (computational power and network capability). Region of interest (ROI) coding allows coding a significant portion of the data with better quality than the rest of the scene. Finally, random access to the data volume or part of it is useful to retrieve portions of the data without decompressing the whole set, which may result in faster computation time for particular applications.

Another important requirement is the capability of up-to-lossless coding. In the case of medical imaging, lossless or near-lossless compression is often mandatory, while lossy compression is accepted for fast browsing or representation purpose [Ham98]. In fact, discarding of even small details might result in the loss of important information and, consequently, severe diagnosis faults. A good compression scheme should then offer a reasonable trade-off between the natural rate-distortion performance and the various features described above.

In 1996, Said and Pearlman developed a wavelet based image codec, Set Partitioning In Hierarchical Trees (SPIHT) [Sai+J96], that offers competitive coding performance together with simplicity and flexibility. In the past, several works have addressed the problem of volumetric image compression. Kim *et al.* [Kim+00] introduced a novel video coder obtained as a three-dimensional extension of the SPIHT algorithm. Many studies are related to lossy-to-lossless coding of volumetric and medical data with integer wavelet transforms. Xiong *et al.* [Xio+03] proposed a modified version of 3D SPIHT and 3D ESCOT with the introduction of the packet wavelet decomposition used in this work and the study of context modeling to obtain an efficient arithmetic encoding of the wavelet coefficients. Schelkens *et al.* [Sch+03] gave an overview of several techniques and proposed a new method based on quad-tree and block-based coding. They also provided a new 3D DCT-based scheme. In another work, Schelkens *et al.* [Sch+00] introduced a 3D cube splitting method, modified the 3D SPIHT and extended JPEG 2000 to three-dimensional images to provide a comparison between the techniques. Integer wavelet transforms for three-dimensional compression are further investigated by Bilgin *et al.* [Bil+00], who also introduced a scheme based on 3D zero-tree coding. Kim *et al.* [Kim+99] employed a slice-based subdivision of the volumetric image and tested several integer wavelet kernels together with the 3D SPIHT algorithm in order to obtain lossless compression.

Other works try to optimize the compression through object-based methods. This is generally done through segmentation of the raw volume to extract useful information and discard of the background and subsequent application of particular coding scheme. Examples of such approaches are the works from Ichimura [Ich98] and Menegaz *et al.* [Men+02]. Visualization-oriented compression is considered in other studies. Bajaj *et al.* [Baj+01] adopted a lossy compression scheme combined with a technique to weight voxel values according to their importance for visualization. Ihm *et al*. combined the Haar wavelet transform and an array-based method for classification of relevant coefficients to obtain fast random access to the compressed bitstream [Ihm+98]. The same goal was pursued by Nguyen *et al.* [Ngu+01], with an approach based on decomposing the image into unit volumes and coding all relevant bits for each significant coefficient, instead of following a bitplane strategy. Finally, Rodler implemented a method based on temporal prediction between adjacent slices in [Rod99], and proposed the use of hashing for the storage of the sparse wavelet coefficients in [Rod01].

This work proposes a new lossy to lossless volumetric coding scheme (3D-SPIHTp hereafter) based on packet integer wavelet transform and SPIHT. Particular attention is spent in the customization of the SPIHT coder in order to comply with the chosen decomposition structure. The resulting coding engine, aside from offering good compression performance, provides flexibility and room for further enhancement.

Chapter 2 is structured as follows. In Section 2.2 the background of this research is briefly described. Attention is spent on the theory of wavelet transforms, packet wavelets and the SPIHT algorithm by Said and Pearlman. The proposed algorithm is presented in Section 2.3. In Section 2.4 the results are shown for different datasets and modalities and compared with other techniques. Finally, in Section 2.5 are the conclusions.

2.2. Background

2.2.1. Lifting Integer Wavelet Transforms

Discrete wavelet transform (DWT) is widely used in signal processing (Fig. 2.1). It relies on the property of compacting most of the signal energy in a few relevant coefficients and defining a larger number of "detail" coefficients for better approximation of the reconstructed signal.

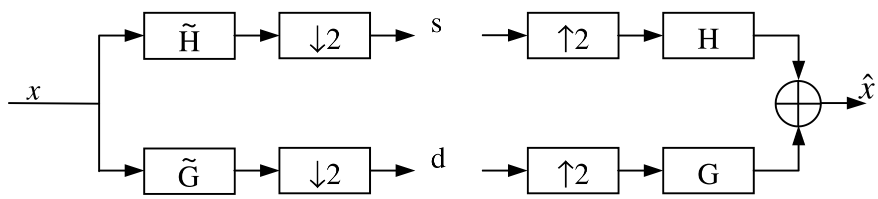


Fig. 2.1. Discrete wavelet transform: 1-dimensional analysis and synthesis stages

The lifting scheme was presented by Sweldens [Swe96, Dau+98] to allow for an efficient implementation of the discrete wavelet transform, with perfect reconstruction granted by the structure of the scheme itself. While the wavelet transform has shown good performance in terms of energy compaction, its classical implementation results in a certain complexity. Another characteristic is that DWT coefficients are generally real numbers, thus requiring further approximation and information loss when employed for image compression. Through the lifting scheme an integer transformation can be obtained by rounding each filter output to the nearest integer.

The input numerical signal is split into two streams (Fig. 2.2), corresponding to even and odd samples. At each lifting step, one signal is convolved with the lifting filter and added to the other. After k steps, the scheme will produce two output streams, corresponding to the low- and high- pass coefficients. In general, lifting steps using the low-pass filter are called *prediction steps*, while those with high-pass filter *update steps*. The synthesis filter is simply obtained by reversing the operations order and the filter signs. Finally, the integer version of the transform differs only in the rounding operation performed after each convolution (Fig. 2.3).

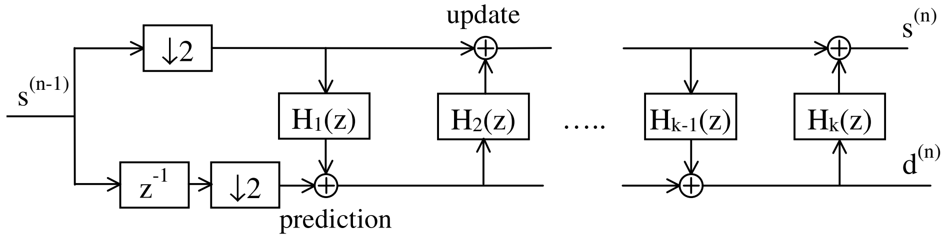


Fig. 2.2. Wavelet transform based on the lifting scheme

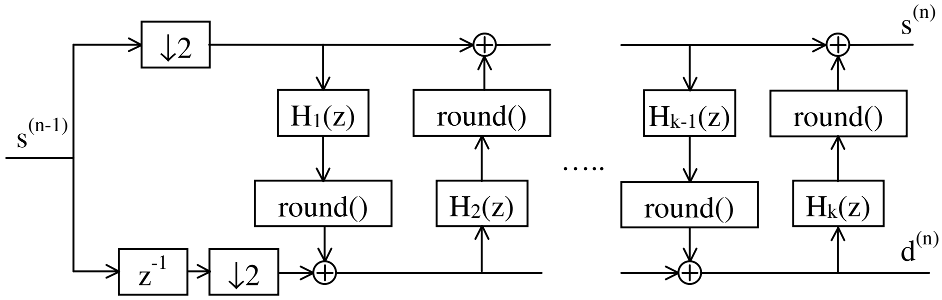


Fig. 2.3. Integer wavelet transform based on the lifting scheme

The advantages of such transform are that it conspicuously reduces the number of arithmetic operations compared to the filter-bank implementation and its structure guarantees full reversibility, regardless of the filter used. As a main drawback, it has been observed that integer transforms offer worse energy compaction performance compared to the real case.

For our research, several filters have been considered and evaluated, mostly derived from interpolative filter. We denote with $c_{i,j}$ the input signal at the i^{th} step with j^{th} position, while with $l_{i,j}$ and $h_{i,j}$ the low- and high-pass outputs respectively.

I(2,2) pair:

$$\begin{aligned} h_{n,m} &= c_{n-1,2m+1} - \lfloor 1/2\left(c_{n-1,2m} + c_{n-1,2m+2}\right) + 1/2 \rfloor \\ l_{n,m} &= c_{n-1,2m} + \lfloor 1/4\left(h_{n,m-1} + h_{n,m}\right) + 1/2 \rfloor \end{aligned} \tag{2.1}$$

I(2,4) pair:

$$\begin{aligned} h_{n,m} &= c_{n-1,2m+1} - \lfloor 1/2\left(c_{n-1,2m} + c_{n-1,2m+2}\right) + 1/2 \rfloor \\ l_{n,m} &= c_{n-1,2m} + \lfloor 19/64\left(h_{n,m-1} + h_{n,m}\right) - 3/64\left(h_{n,m-2} + h_{n,m+1}\right) + 1/2 \rfloor \end{aligned} \tag{2.2}$$

I(4,2) pair:

$$\begin{aligned} h_{n,m} &= c_{n-1,2m+1} - \lfloor 9/16\left(c_{n-1,2m} + c_{n-1,2m+2}\right) - 1/16\left(c_{n-1,2m-2} + c_{n-1,2m+4}\right) + 1/2 \rfloor \\ l_{n,m} &= c_{n-1,2m} + \lfloor 1/4\left(h_{n,m-1} + h_{n,m}\right) + 1/2 \rfloor \end{aligned} \tag{2.3}$$

I(4,4) pair:

$$\begin{aligned} h_{n,m} &= c_{n-1,2m+1} - \lfloor 9/16\left(c_{n-1,2m} + c_{n-1,2m+2}\right) - 1/16\left(c_{n-1,2m-2} + c_{n-1,2m+4}\right) + 1/2 \rfloor \\ l_{n,m} &= c_{n-1,2m} + \lfloor 9/32\left(h_{n,m-1} + h_{n,m}\right) - 1/32\left(h_{n,m-2} + h_{n,m+1}\right) + 1/2 \rfloor \end{aligned} \tag{2.4}$$

I(2+2,2) pair:

$$\begin{aligned} h^1_{n,m} &= c_{n-1,2m+1} - \left\lfloor 1/2\left(c_{n-1,2m} + c_{n-1,2m+2}\right) + 1/2 \right\rfloor \\ l_{n,m} &= c_{n-1,2m} + \left\lfloor 1/4\left(h^1_{n,m-1} + h^1_{n,m}\right) + 1/2 \right\rfloor \\ h_{n,m} &= h^1_{n,m} - \left\lfloor 1/16\left(-l_{n,m-1} + l_{n,m} + l_{n,m+1} - l_{n,m+2}\right) + 1/2 \right\rfloor \end{aligned} \tag{2.5}$$

S+P:

$$\begin{aligned} h_{n,m} &= c_{n-1,2m+1} - c_{n-1,2m} \\ l_{n,m} &= c_{n-1,2m} + \left\lfloor \left(h_{n,m} + 1\right)/2 \right\rfloor \\ \hat{h}_{n,m} &= h_{n,m} - \left\lfloor \alpha_{-1}\left(l_{n,m-2} - l_{n,m-1}\right) + \alpha_0\left(l_{n,m-1} - l_{n,m}\right) + \alpha_1\left(l_{n,m} - l_{n,m+1}\right) - \beta\left(h_{n,m+1}\right) + 1/2 \right\rfloor \end{aligned} \tag{2.6}$$

The S+P filter was introduced by Said and Pearlman [Sai+S96]. It is an improvement of the S transform, obtained with the addition of an extra lifting step for the high-pass coefficients. Three different sets of parameters α_{-1}, α_0, α_1, β are suggested by the authors and were considered in this chapter (A, B, C).

2.2.2. Packet Wavelet Transform

Through the recursion of wavelet transformation steps, the input signal can be further decomposed into several subbands while its energy is further compacted into a smaller number of coefficients. Several recursion schemes have been used for data compression. The dyadic decomposition is certainly the most known; it relies on iterating the transformation stage on the low-passed signal only (Fig. 2.4). Other schemes are generally grouped in the category of *Packet Wavelets*, defined as any one of a collection of orthonormal transforms, each of which can be readily computed using a simple modification of the pyramid algorithm for the DWT.

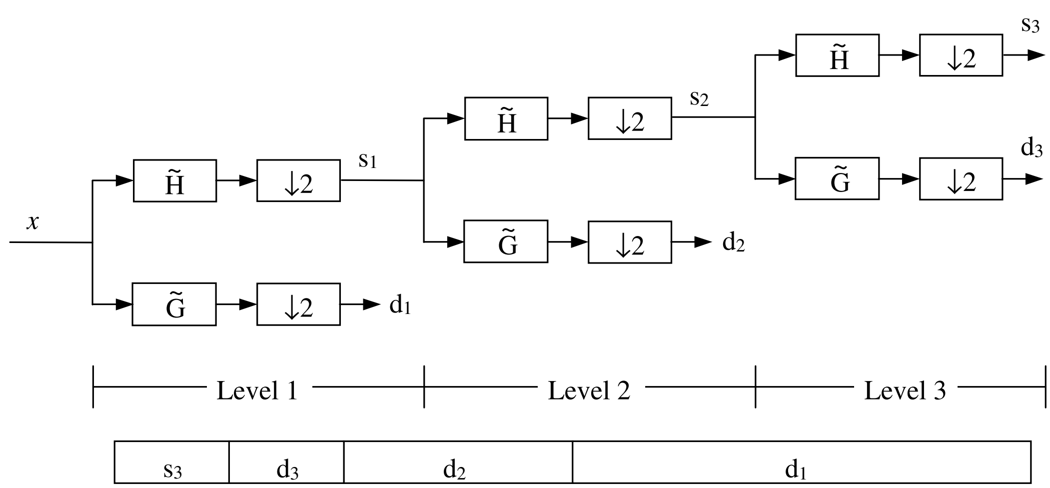


Fig. 2.4. 3-levels 1D dyadic wavelet decomposition analysis

Our choice to implement the scheme depicted in Fig. 2.5 was taken, among other reasons which will be further discussed, in order to avoid potential complications related to the scaling of the integer lifting scheme. While all previous filter pairs (2.1-2.6) meet the perfect reconstruction constraint, they result in non-unitary transforms. For this reason, a scaling factor of $\sqrt{2}$ or $1/\sqrt{2}$, deriving from the lifting decomposition, is generally needed after each filtering operation. In the case of two dimensional signals, such as conventional images, application of the transform produces a scaling of 2 or 1/2, factors of perfect integer precision. On the other hand, three dimensional signals, such as volumetric data, cannot rely on such property. By using the decomposition scheme of Fig. 2.5 and an even number of decomposition stages, such drawback can be avoided, as shown in Fig. 2.6. It must be noticed that since the scaling is done with integer arithmetic or, equivalently, bit shifting, least significant bits will be lost and lossless reconstruction would be harmed. Therefore, all scale factors are multiplied in order to produce all nonnegative powers of 2. These scale factors are inverted after decoding the wavelet coefficients. From a practical point of view, scaling is generally required when dealing with lossy to lossless coding. In fact, while perfect reconstruction is granted even with unscaled transform, scaling allows for correct significance assignment to the transformed coefficients, thus providing correct rate-distortion in lossy coding.

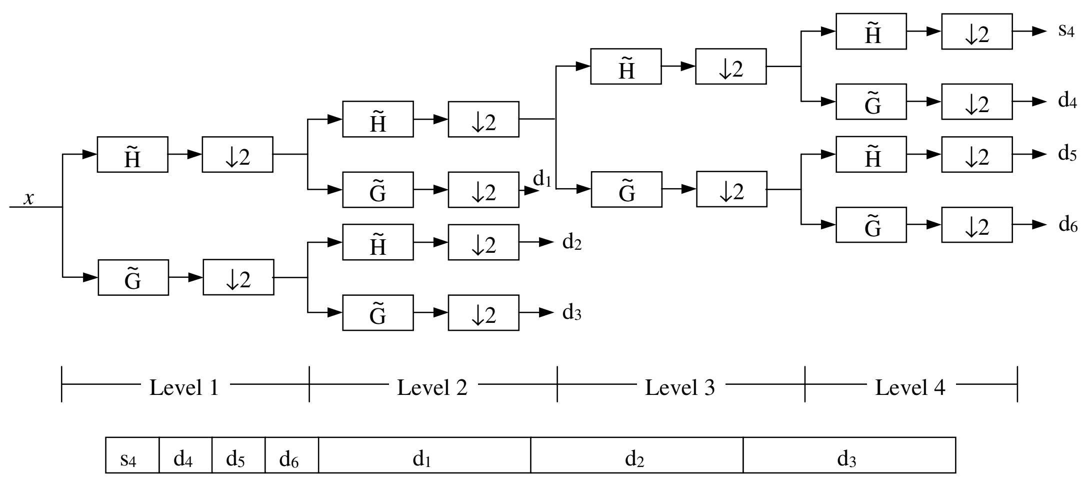


Fig. 2.5. 4-levels 1D packet wavelet decomposition analysis

decompositions

<table>
<tr><td>1:</td><td colspan="8">$\sqrt{2}$</td><td colspan="8">$1/\sqrt{2}$</td></tr>
<tr><td>2:</td><td colspan="4">2</td><td colspan="4">1</td><td colspan="4">1</td><td colspan="4">1/2</td></tr>
<tr><td>3:</td><td colspan="2">$2\sqrt{2}$</td><td colspan="2">$2/\sqrt{2}$</td><td colspan="4">1</td><td colspan="4">1</td><td colspan="4">1/2</td></tr>
<tr><td>4:</td><td>4</td><td>2</td><td>2</td><td>1</td><td colspan="4">1</td><td colspan="4">1</td><td colspan="4">1/2</td></tr>
</table>

Fig. 2.6. Scaling factors for several decomposition stages (1-dimensional signal) of the adopted packet wavelet transform

2.2.3. SPIHT

Set Partitioning In Hierarchical Trees was introduced by Said and Pearlman [Sai+J96] as a novel low-complexity and efficient image coder. It allows for progressive transmission by sorting the transform coefficients in order of significance and exploiting the redundancy in the tree structure of the wavelet transformed image.

The algorithm relies on three linked lists, called list of insignificant sets (LIS), list of insignificant pixels (LIP) and list of significant pixels (LSP), which are dynamically modified during execution. LIP and LSP elements represent single pixels, and consist of a set of coordinates (i, j). LIS elements represent coefficient sets and include type information (A if representing all descendants, B if representing all descendants except the offspring), together with the position indication.

The method consists of two main steps called *sorting* and *refinement* pass. During the sorting pass, pixels and sets are explored following the tree structure shown in Fig. 2.7. One pixel is defined significant when its value, $c_{i,j}$, satisfies:

$$2^n \le \left|c_{i,j}\right| \le 2^{n+1} \tag{2.7}$$

where n decrements at each pass and is initialized to be:

$$n = \left\lfloor \log_2\left(\max_{(i,j)}\left\{\left|c_{i,j}\right|\right\}\right)\right\rfloor \tag{2.8}$$

Similarly, sets are considered significant when at least one of their descendant coefficients satisfies Equation 2.7. Each time one pixel/set is tested for significance, an output bit is provided. When the significance test returns 1, the element is moved from the insignificant list to the significant one. In case of pixel significance, an additional sign bit is produced: 1 or 0, for positive and negative result respectively.

During the refinement pass, an output bit for each n^{th} bit of all significant coefficients (belonging to the LSP) of the previous iteration is produced.

A completely embedded and progressive bit-stream can be obtained by iterating the previous steps. The decoder structure is then similar to that of the coder, except for the progressive update of the reconstructed image. The main advantages of SPIHT are its efficiency, adaptability and precise rate control. In fact, it is possible to stop the coder/decoder at any time, in order to obtain one exact bitrate. Alternatively, running the process until the last bitplane results in lossless coding, provided the wavelet coefficients were not subjected to quantization.

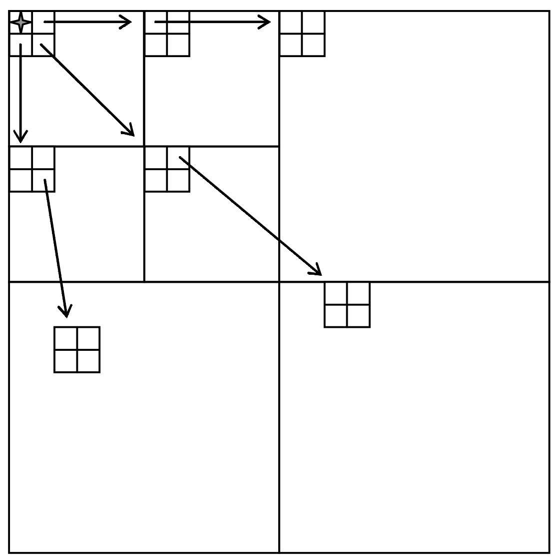


Fig. 2.7. Hierarchical tree decomposition of the wavelet transform

2.3. Processing Structure

In Section 2.2 several topics related to transform and coding have been considered. These are the starting points to build the codec infrastructure, which will be discussed in the following. The general coding scheme is described in Section 2.3.1, while issues related to the decomposition scheme, refinement step and context entropy coding are covered in Sections 2.3.2 to 2.3.4.

2.3.1. Coding Scheme

The proposed volumetric coder adopts the general scheme for transform-based coders illustrated in Fig. 2.8. We first define a *tile* as any volumetric block extracted from a given dataset; then, the input volumetric data can be initially tiled in GOS (Group Of Slices) or blocks, so that each tile is considered independently. Tiling results in performance degradation in terms of coding efficiency, but allows for faster processing. This is particularly relevant for volume data of large dimension, where memory allocation during the processing can be prohibitive.

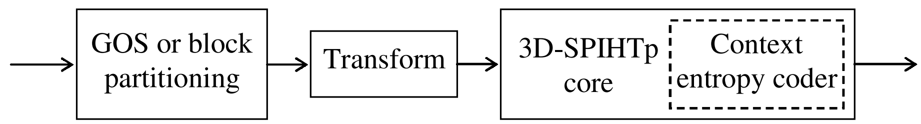


Fig. 2.8. Encoder scheme

Each tile is further transformed by the 3D integer lifting wavelet transform described in Section 2.2.1 and 2.2.2 and illustrated in Fig. 2.9. The separable transform is first applied along the z axis (z or *inter-slices* dimension) following the packet wavelet decomposition illustrated in Section 2.2.1. The dyadic transform is then applied to each slice (xy or *intra-slices* dimension). Thanks to this scheme, a different number of decomposition levels is applicable to the inter- and intra-slice transform. At the same time, we can choose among the integer filters defined in (2.1) to (2.6) to be applied along z or xy independently. Given the general anisotropic nature of volumetric datasets, *i.e.* the intra-slice sampling rate is generally smaller than that along inter-slice axis, the feature considered previously is useful to choose the best combination of filter type and decomposition levels. Directional anisotropy implies that voxel correlation is generally greater inside each slice than between slices, allowing for better intra-slice redundancy exploitation. For such reason, it is possible to choose integer filters of smaller size and a smaller number of decomposition levels for implementing the transform along the slice axis, while opting for more complex filters and more decomposition levels for intra-slice transform. Such choice is also dependant on the data size. To sum up, besides offering the integer scaling capability discussed in Section 2.2.2, the packet transform allows to avoid the limitations of dyadic decomposition, which implies symmetry even when such property is not granted.

Each transformed tile is then passed to the 3D-SPIHTp core. This block provides scalable data redundancy exploitation by decomposition of the transformed tile into hierarchical tree. The main part is an extension of the SPIHT algorithm to 3D data that implements the packet decomposition hierarchical tree. Further context entropy coding can be implemented inside the core itself and will be discussed in Section 2.3.4.

The decoding process simply follows the encoding in reverse order; 3D-SPIHTp decoder, inverse integer lifting filters and block composer are provided in order to perform the full decoding stage.

2.3.2. 3D Packet Decomposition

In order to exploit the same packet structure of the wavelet transform, the SPIHT coder must be accurately modified and integrated. We first consider the structure of the packet hierarchical tree illustrated in Fig. 2.9. While intra-slice decomposition follows the conventional dyadic structure, inter-slice samples are not related by a factor 2 scaling of coordinate indices. In particular, while the highest level subband, LL_N, has only one correspondent plane in each other subband of same level, LH_N, HL_N and HH_N, each of these can be considered parent of 4 other planes at the corresponding subband at $N-1$ level. This rule becomes regular after the $N-1$ level, so that each plane of a certain subband is the parent of 4 other planes at the corresponding subband at the lower decomposition level.

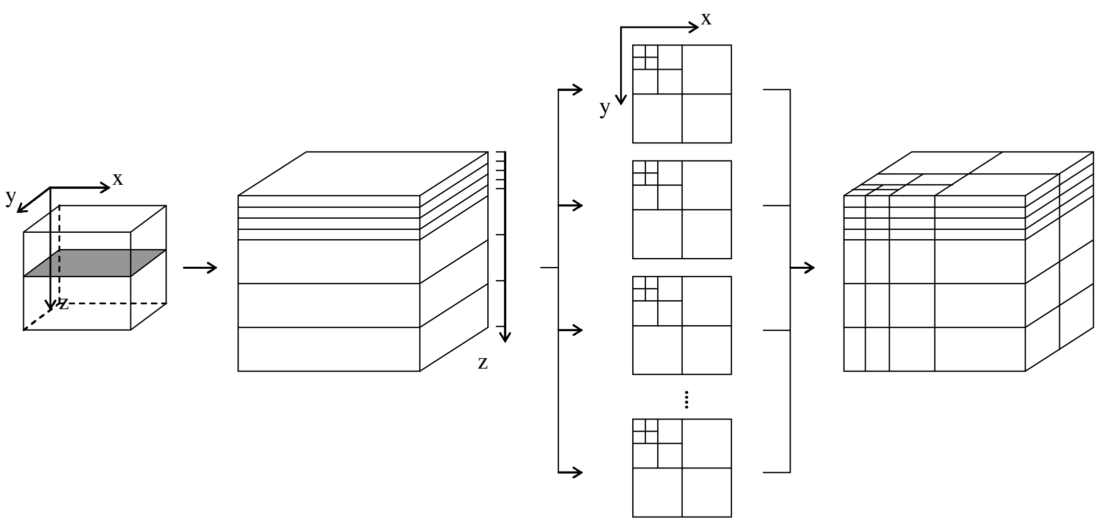


Fig. 2.9. The 3D packet decomposition used for the work with 3 *xy* levels and 4 *z* levels.

This structural choice (illustrated in Fig. 2.10) allows for the exploration of the whole volume with correct hierarchical dependence. In all cases, child planes are retrieved with a scaling factor and a shifting coefficient, instead of a simple scaling of 2 as for dyadic decomposition. To identify subbands we can introduce the notation XX_nYY_m, indicating with XX_n the subband from the inter-slice transform and with YY_m that from the intra-slice transform. Then, as in the example of Fig. 2.10, we can locate the children of LL_2LL_2 by applying:

$$z_{child} = i \cdot size_{band} + z_{parent} \quad i = 1,2,3 \tag{2.9}$$

where z_x is the location of x on the slice axis and $size_{band}$ is the z-dimension of the parent subband. This is true for the relation among n level subbands. All other dependencies can be found with:

$$z_{child} = 4 \cdot z_{parent} + i \quad i = 0,1,2,3 \tag{2.10}$$

In the SPIHT algorithm the structural relation is evidenced by information of the LIP and LSP, with the definition of two types of sets: set with insignificant descendants (A-sets) and set with insignificant descendant of offspring (B-sets). In order to exploit the previously described structure the coder has been modified with the introduction of four new set types.

Types **X** and **x** designate root sets for subband decompositions along the z axis at the highest z-decomposition level, while **Z** and **z** represent sets for subbands decomposition along the z axis at lower z-decomposition levels. Capital or lower-case letters indicate the corresponding set with insignificant descendants or insignificant descendant of offspring respectively, as for A and B sets. Following the same notation used in [Sai+J96], we define:

$\mathcal{O}(i, j, k)$: set of coordinates of all offspring of node (i, j, k);

$\mathcal{D}\ (i, j, k)$: set of coordinates of all descendants of the node (i, j, k);

$\mathcal{L}(i, j, k) = \mathcal{D}(i, j, k) - \mathcal{O}(i, j, k)$;

$S_n(x)$: significance test operation on x at bit-plane n; returns 1 if x is significant, 0 otherwise;

Then, the following code is added to the sorting pass:

If the entry is of type **X** then

- output $S_n(i, j, k)$;
- if $S_n(\mathcal{D}(i, j, k)) = 1$ then
- for each $(u, v, w) \in \mathcal{O}(i, j, k)$ do:
  - output $S_n(u, v, w)$;
  - if $S_n(u, v, w) = 1$ then add (u, v, w) to the LSP and output sign of $c_{u,v,w}$;
  - if $S_n(u, v, w) = 0$ then add (u, v, w) to the LIP;
  - if v or w is odd then add (u, v, w) to the LIS as sets of type **A**;
  - if v or w is even then if $\mathcal{D}(i, j, k) \neq 0$ then move (i, j, k) to the end of LIS as an entry of type **x** and process step **x**; otherwise remove (i, j, k) from the LIS;

if the entry is of type **x** then

- output $S_n(\mathcal{L}(i, j, k))$;
- if $S_n(\mathcal{L}(i, j, k)) = 1$ then
  - add each $(u, v, w) \in \mathcal{O}(i, j, k)$ to the end of the LIS as an entry of type **Z**;
  - remove (u, v, w) from the LIS;

A similar instruction with Z instead of X and z instead of x is implemented for the Z and z sets. A major difference resides in the number and position of descendants for sets Z and z, but the structure is similar. It is important to notice that the significance testing function is modified according to the new tree organization.

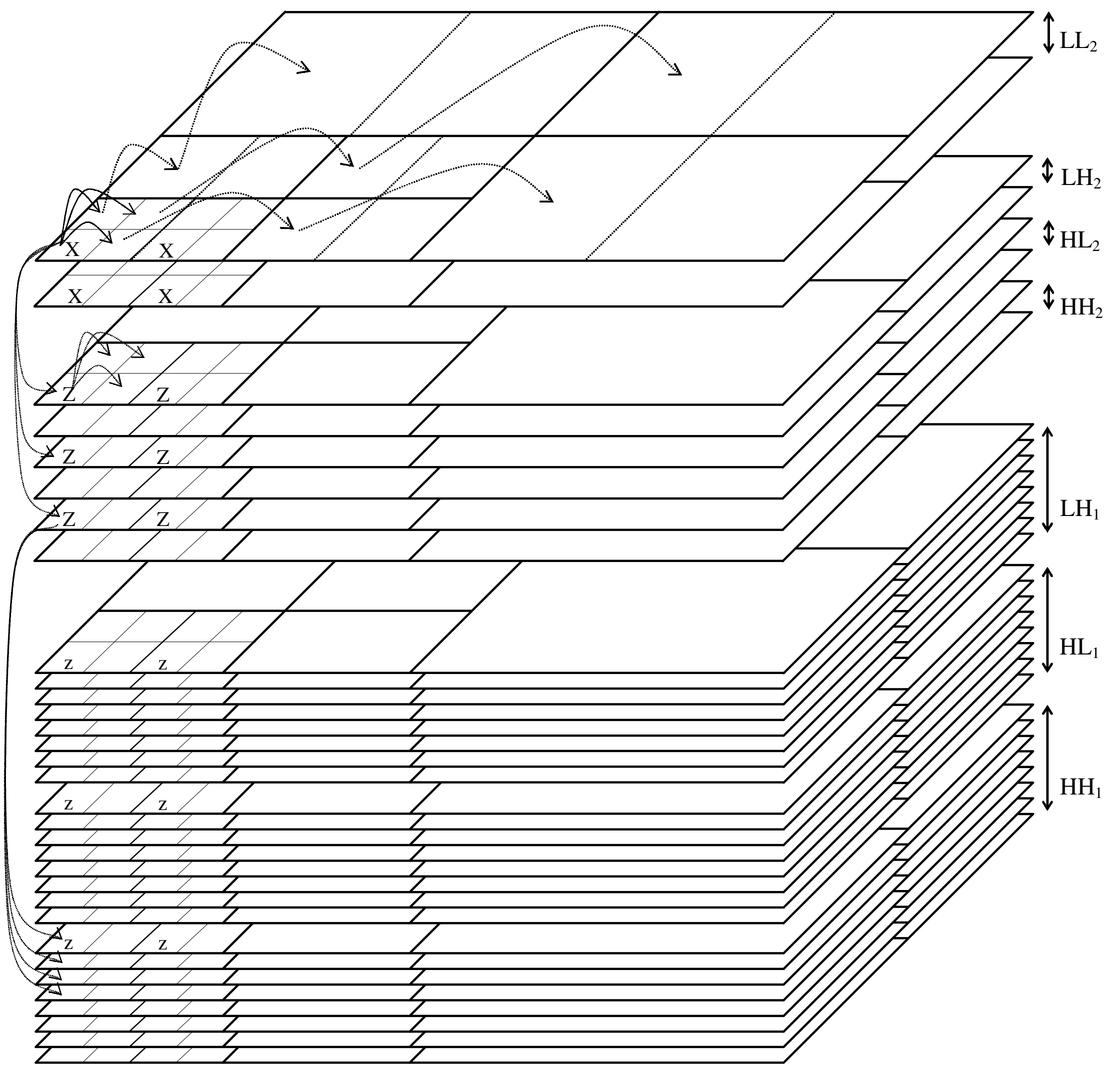


Fig. 2.10. Sample packet tree structure for the 3D-SPIHTp codec.

2.3.3. Refinement Pass Considerations

During the wavelet decomposition, each subband coefficient is scaled according to its spatial location to obtain a unitary transform. Given the chosen structure for packet wavelet decomposition along the slice dimension and intra-slice dyadic wavelet decomposition, multiplicative scaling factors can be chosen to be always powers of 2, as explained in Section 2.2.2. Such operation is equivalent to bit-shifting the coefficients.

During the refinement pass, which is performed after all sets and insignificant voxels have been tested for significance in comparison to a certain bit-plane n, coefficients belonging to the LSP are refined by outputting their n^{th} bits. Consequently, the refinement pass produces predictable 0's correspondent to the least significant bits of all scaled coefficients. If one coefficient was scaled up by a factor of 4, for instance, the refinement pass would produce two 0's correspondent to the two coefficient's least significant bits. These output bits are predictable, therefore avoidable, since directly dependent on the applied scaling factor. Such trend is particularly relevant for lossless compression, when the input data is processed until the last bit-plane. In this case, each scaled coefficient produces a negligible output bit each time it is compared with a threshold lower than its scaling factor.

In order to solve such a drawback, a simple conditional scheme can be added to the co-decoder; location of each coefficient together with dataset size, decomposition levels and current bit-plane must be considered to decide for that element's further processing. As an example, if the candidate coefficient during the scan related to bit-plane 1 belongs to a subband scaled up by a factor of 4 (*i.e.* HL_2LH_2), it should be excluded from the refinement. The same mechanism applies to the decoding stage, since information on location is discovered from the tree structure, while current bit-plane, data size and decomposition levels are known to the decoder.

2.3.4. Context Entropy Coding

The 3D-SPIHTp codec is able to exploit spatial and frequency redundancy of the wavelet compressed volume while ordering its hierarchical structure to provide a fully scalable bitrate. However, the output produced by each coding stage may still be affected by a certain amount of correlation. In order to increase performance, additional entropy context coding can be applied. While for 2D compression additional entropy coding provides only very limited gain with the SPIHT algorithm, it has been noted that for 3D methods this technique is generally imperative.

Particular attention is spent on the output deriving from the processing of A sets during the sorting pass, which constitutes 50% of the coded stream on average. Each time an A set is considered significant, a significance output bit is produced for each of its four offsprings. Consecutively, each time one of its offspring is found significant, a corresponding sign bit is produced. This leads to 81 possible symbols of different length. As an example, if no offspring is found significant the string 0000 is output, while if the upper-right offspring is positive significant, 01000 is produced.

In Fig. 2.11 the symbol probability has been averaged over four different datasets, each coded with four different parameter combinations. It can be noticed that smaller symbol length does not always correspond to a higher probability. In particular, sets with one significant offspring are more probable than sets with no significant offspring, which are coded by a symbol of smaller length. At the same time the negative sign appears more probable; this information can be related to the choice of wavelet filters, but is nonetheless interesting for entropy coding purposes. We can also notice that almost 60% of occurrences is supplied by 10 symbols only.

To increase performance, we adopted a simple adaptive Huffman coder. As a result, the A-set output decreased by an average factor of 11%, resulting in a final gain of approximately 5%. Such figures can be further increased by implementing more sophisticated techniques and extending such considerations to other sources of output.

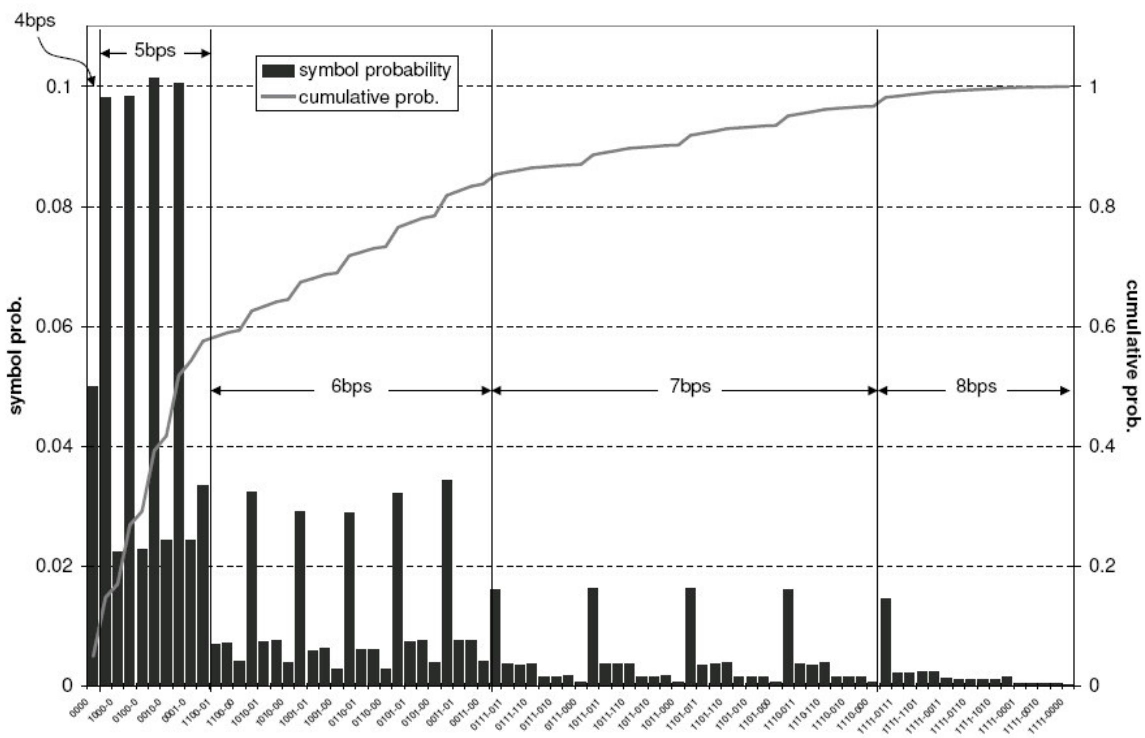


Fig. 2.11. Study of A-set behaviour.

2.4. Results and Discussion

In order to evaluate the performance of the proposed method we consider the same 8bit CT and MR volumetric medical dataset used in [Xio+03] and [Bil+00] for easy comparison (Table 2.I). The images present different techniques, sizes, and information content. Then, the following parameters are considered:

- Decomposition levels (intra- and inter-slice)
- Filter type (intra- and inter-slice)
- Bitrate

For lossy compression, the peak SNR (PSNR) is used as objective quality metric; its definition for 8bit signals is:

$$\mathrm{PSNR} = 20\log_{10}\frac{255}{\sqrt{\mathrm{MSE}}}, \tag{2.11}$$

where MSE is the mean square error between original and reconstructed volumes. Although such metric may be inadequate to evaluate comprehensively the effect of compression of medical data, it constitutes the term of comparison for its average significance and its simplicity. In Section 2.4.1 lossless results are discussed, while in Section 2.4.2 lossy compression is considered.

TABLE 2.I. Description of the medical datasets used for experiments

Name	Description	Voxel size	Volume size
CT_skull	Tripod fracture	0.7x0.7x2	256x256x192
CT_wrist	Healing scaphoid fracture	0.17x0.17x2	256x256x128
CT_carotid	Internal carotid dissection	0.25x0.25x1	256x256x64
CT_aperts	Apert's syndrome	0.35x0.35x2	256x256x96
MR_lever_t1	Normal liver	1.45x1.45x5	256x256x48
MR_lever_t2	Normal liver	1.45x1.45x5	256x256x48
MR_sag_head	Left exopthalmos	0.98x0.98x3	256x256x48
MR_ped_chest	Congenital hearth disease	0.78x0.78x5	256x256x48

2.4.1. Lossless Performance

3D-SPIHTp is tested on the whole CT and MR datasets. All coding results are reported in terms of bits per voxel (bpv) based on real compressed file size.

In Tables 2.II and 2.III we compare the different integer wavelet filters described in Section 2.2.1 for the CT and MR datasets respectively. Filter types refer to the intra-slice transform, while the adopted inter-slice transform is (2,2) for all experiments. In all cases, a single coding unit equal to the full slice number was used to compress the volumes. It can be noticed that the (2+2,2) filter kernel outperforms all other kernels for all datasets except for *MR_liver_t2*, where (4,2) prevails. The results are comparable with those reported in [Xio+03] and [Bil+00].

TABLE 2.II. Comparison of different wavelet transforms on the CT datasets

	Dataset:	*CT_skull*	*CT_wrist*	*CT_carotid*	*CT_aperts*
Wavelet Transform	(2,2)	2.074162	1.382904	1.588280	1.081957
	(2,4)	2.084404	1.385376	1.596066	1.087734
	(4,2)	2.072756	1.372028	1.581793	1.081394
	(4,4)	2.072871	1.372230	1.581924	1.082203
	(2+2,2)	**2.067591**	**1.369475**	**1.579716**	**1.078274**
	S+P A	2.202656	1.560496	1.647465	1.173991
	S+P B	2.178009	1.585457	1.617043	1.182898
	S+P C	2.165796	1.574711	1.602703	1.175991

TABLE 2.III. Comparison of different wavelet transforms on the MR datasets

	Dataset:	*MR_liver_t1*	*MR_liver_t2*	*MR_sag_head*	*MR_ped_chest*
Wavelet Transform	(2,2)	2.360825	1.743492	2.307030	1.881426
	(2,4)	2.385874	1.752467	2.307953	1.882786
	(4,2)	2.386182	**1.720655**	2.308436	1.878120
	(4,4)	2.385406	1.720858	2.306656	1.877863
	(2+2,2)	**2.382884**	1.721860	**2.300779**	**1.874376**
	S+P A	2.545245	1.802205	2.445971	2.193359
	S+P B	2.538325	1.767794	2.489944	2.265028
	S+P C	2.515851	1.735677	2.494804	2.265028

Comparison of different coding unit sizes is shown in Table 2.IV. The CT_wrist dataset has been coded using 5 decomposition levels, (2+2,2) filter intra-slice, variable decomposition levels and (2,2) filter inter-slice. GOS of 16, 32, 64, and 128 slices are reported. It is possible to notice the decreasing trend of the coding rate while GOS size increases. However, for constant GOS size, a higher number of z decomposition levels do not provide better performance. This could be explained by the reduced inter-slice correlation, proven by the significant difference between transversal and planar resolution (ratio of 0.085).

TABLE 2.IV. Comparison of different coding unit sizes and z-decomposition levels for the *CT_wrist* dataset coded with the (2+2,2) filter and 5 intra-slice decomposition levels

GOS size	z-decomposition levels	bpv
16	2	1.426689
32	2	1.393799
64	2	1.375358
64	4	1.382973
128	2	1.369475
128	4	1.375102

Finally, Table 2.V represents the compression results for the *MR_ped_chest* dataset with different integer wavelet filter types and combinations of coding unit and decomposition levels. In this case, *lvxy* and *lvz* indicate the number of intra- and inter-slice decomposition levels respectively. We can see that the (2+2,2) filter kernel offers the best result in all cases. Moreover, performance increases from left to right, following the increase of GOS size and inter-slice decomposition levels. Although comparable to the state of the art, lossless results are inferior to those produced by Cho *et al.* [Cho+] in their latest research with an asymmetric tree structure implementation of 3D SPIHT.

TABLE 2.V. Comparison of different coding unit sizes and z-decomposition levels for the *MR_ped_chest* dataset coded with the (2+2,2) filter and 5 intra-slice decomposition levels

Wavelet transform	GOS 16 5lvxy, 2lvz	GOS 32 5lvxy, 2lvz	GOS 64 5lvxy, 2lvz	GOS 64 5lvxy, 4lvz
(2,2)	2.011711	1.967993	1.930326	1.881426
(2,4)	2.013733	1.969574	1.931879	1.882786
(4,2)	2.012524	1.967979	1.929983	1.878120
(4,4)	2.012136	1.967930	1.929777	1.877863
(2+2,2)	**1.998730**	**1.955578**	**1.918274**	**1.874376**
S+P A	2.283400	2.253378	2.233408	2.193359
S+P B	2.336155	2.311707	2.297321	2.265028
S+P C	2.326920	2.304028	2.290434	2.265028

2.4.2. Lossy Performance

Thanks to SPIHT property of precise rate control and idempotence, lossy coding results can be easily generated by setting the desired decoding bitrate once the dataset has been losslessly compressed. In order to evaluate lossy performance, we first consider the complete *CT_skull* dataset. As coder parameters we choose 5 levels of intra-slice decomposition, 2 levels inter-slice decomposition, (2+2,2) intra-slice and (2,2) inter-slice integer filters. Two typical compression rates (0.1bpp and 0.5bpp) are chosen and results showing PSNR/slice are illustrated in Figs. 2.12 and 2.13. For each of these experiments, three GOS sizes are chosen: 16, 32 and 192 (the whole volume) slices. In the case of 16 and 32 slices, the GOSs were processed separately at the desired bitrate, since no rate-allocation technique is provided.

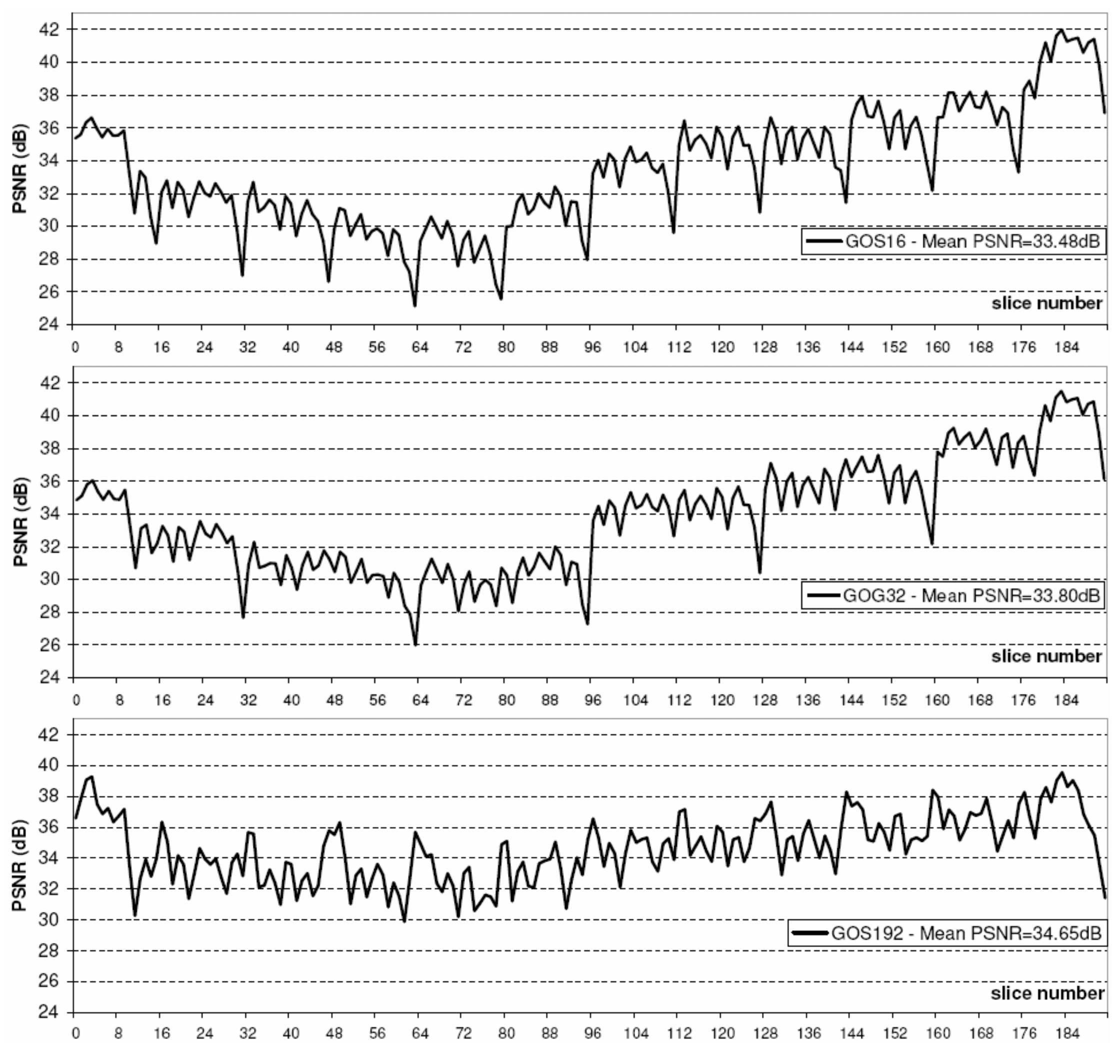


Fig. 2.12. Lossy compression results for volume *CT_skull* at 0.1bpp.

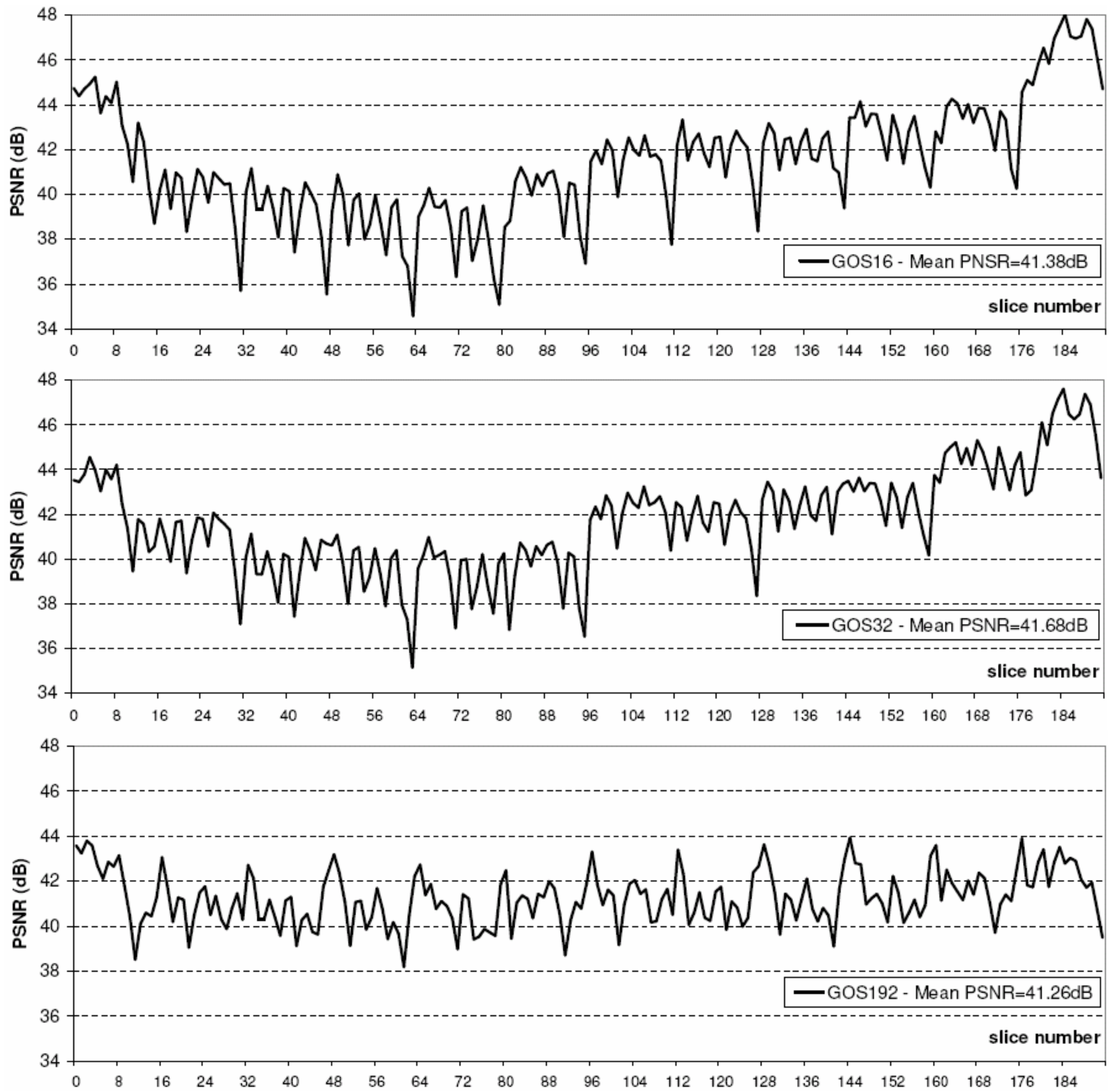


Fig. 2.13. Lossy compression results for volume *CT_skull* at 0.5bpp.

The large PSNR variation is typical of 3D compression, as reported also in [Xio+03]. Such phenomenon depends both on image contents and transform operation. However, it can be observed that the use of larger GOSs tends to regularize the PSNR variability over different slices. In fact, the PSNR variance is visibly smaller for both bitrates when the largest GOSs are considered, even though such choice does not seem to provide a significant average PSNR improvement. In the case of small GOSs, we can observe the steep drop in PSNR that corresponds to GOS boundaries; this behavior is similar to the effect of tiling adopted by conventional block coders and can be diminished with the use of point-symmetric lifting wavelet filters [Kha+02].

Results show that an average PSNR of 33÷34dBs is obtained at 0.1bpp while a value of 42.5 is reached at 0.5bpp. These figures are comparable to the state of the art.

In Fig. 2.14 we show the rate-distortion curve for datasets *CT_wrist*, *MR_liver_t2* and 128 slices of *CT_skull*. The compression was performed by setting 5 levels of intra-slice decomposition, 2 levels inter-slice decomposition, (2+2,2) intra-slice and (2,2) inter-slice integer filters and full-slices GOS. The decoding was simply obtained by setting the desired rate in a range from 0.125 to 0.9375bpp with a step size of 0.0625bpp. The different PSNR results are imputable to image content; the dataset *CT_wrist*, for instance, has a large percentage of background. Sudden variations of PSNR can be observed at 0.5 and 0.875bpp. Such behavior is mainly produced by the transform operation that is only approximately uniform and the introduction of a lower order bitplane. However, the curve shape follows the expected increasing trend.

Finally, some pictorial examples of lossy-compressed volumes are shown in Fig. 2.15. For each dataset, the same slice coded at 0.125, 0.5bpp and the original are shown. At 0.125bpp coding artifacts can be observed, but perceptual quality is very acceptable.

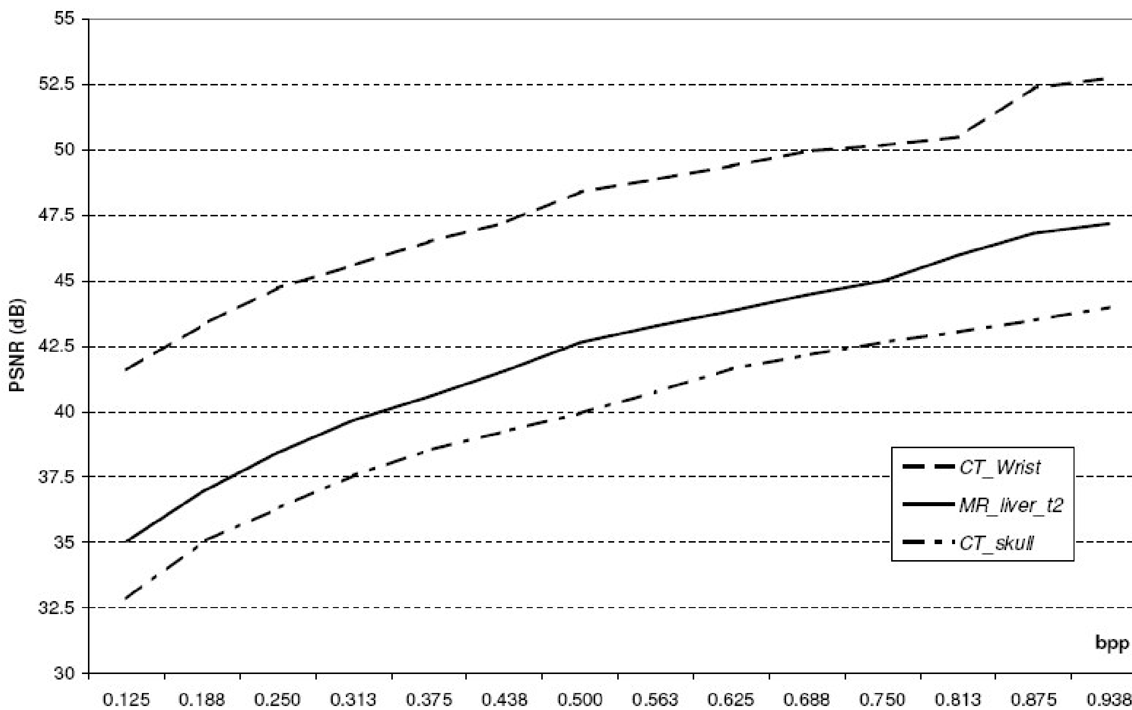
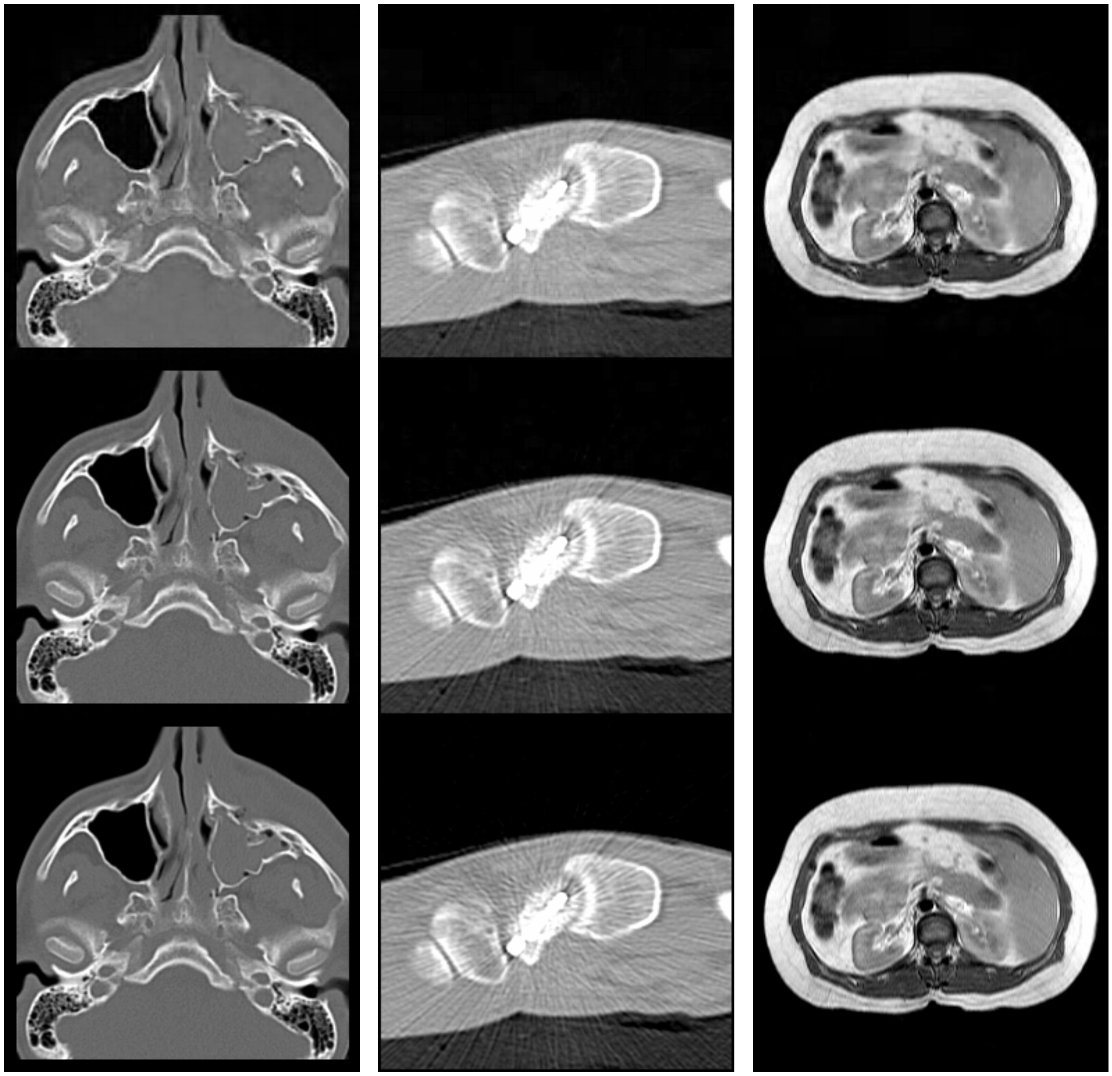


Fig. 2.14. Rate-distortion curve for three volumetric datasets.



(a) (b) (c)

Fig. 2.15. Example of lossy compression result for three volumetric datasets; slice 60 of *CT_skull* (a), slice 38 of *CT_wrist* (b) and slice 13 of *MR_liver_t1* (c). From top to bottom: 0.125bpp, 0.5bpp and original.

2.5. Conclusions and Future Work

In this section we presented a new lossy to lossless volumetric coding paradigm based on integer wavelet transform and SPIHT. Considerations on data structure, 3D integer wavelet transform and scaling brought to the choice of adopting a packet wavelet decomposition scheme along the slice dimension. The proposed decomposition scheme was implemented and tested in order to offer a good flexibility. Several parameters, such as filter type and number of decomposition levels can be chosen independently along different dimensions, allowing for coding optimization.

Subsequently, the basic SPIHT coder has been extended and extensively modified to take advantage of the hierarchical subband structure created by the adopted wavelet transform. Issues on the refinement pass and context entropy coding are discussed to obtain additional coding gain. Finally, the proposed scheme is extensively tested on a well-known dataset both for lossy and lossless performance. Results obtained for both lossy and lossless compression are good and comparable to those of the state of the art. The coder succeeds in reaching good performance for lossy to lossless compression, confirming the validity of the proposed scheme.

Chapter 3

ROI Processing

3.1 Introduction

Nowadays, several scientific fields, such as medicine (CT, MR, PET), physics (atomic and molecular models) or meteorology (flow and pressure models), make extensive use of volumetric data representation. The increasing diffusion of multi-dimensional data has, in turn, boosted the need for segmentation and coding technologies already acknowledged in the still picture and image sequence cases.

Multi-dimensional image segmentation finds many applications. The definition of regions of interest (ROIs) can be embedded in the image codec as an extra feature that allows assigning higher quality to desired portions of an image, while approximating the rest of the scene with a coarser representation. Another application requiring ROI definition is object-based lossless compression. Examples of these applications are the work of Ueno and Pearlman [Uen+03], Menegaz and Thiran [Men+02] and Gokturk *et al.* [Gok+01]. Segmentation helps in the visualization of complex data, a problem of particular importance for image analysis, *e.g.* medical diagnosis, and non-intrusive inspection and testing. Finally, segmentation facilitates the quantitative image analysis task, as for features count and measurement. Each of these tasks has specific requirements in terms of precision, reliability and robustness, whereas a successful segmentation routine depends on the complexity and the acquisition quality of the studied data.

Several techniques for medical and volumetric data segmentation have been implemented in the literature. Gamio *et al.* [Gam+04] employed normalized cuts for the segmentation of vertebral bodies from MR images. Fan *et al.* [Fan+02] adopted parallel genetic algorithms in an active model framework. In [Kuh+97], Hühne *et al.* implemented a visualization oriented segmentation approach through watershed and region growing. Other works rely on the theory of level sets, as in [Ho+02] and [Dro+01], where a multi-grid method for the model's propagation is implemented. Snakes are considered by Tek *et al.* [Tek+96] and van

Ginneken *et al.* [Gin+02], where optimal local features are adopted in order to characterize the active shape landmarks.

3.2 Snake-based Approach

3.2.1 Introduction

This research proposes a semi-automatic method for object segmentation from volumetric data. The proposed scheme relies on the active contour theory, first introduced by Kass *et al*. in [Kas+87] and extends it through a slice-based approach. Major effort is spent in the preprocessing stage and in the correlation between adjacent slices. The technique offers a valid solution for a number of applications. It is mainly designed for ROI identification and extraction, which finds primary application in object-oriented coding.

In Section 3.2.2 the theory of snakes is briefly described. In Section 3.2.3 a scheme for the complete algorithm is provided and the method for exploiting adjacent slice correlation is described. Section 3.2.4 deals with the preprocessing stage. In Section 3.2.5 results are shown and discussed. Finally, in Section 3.2.6 conclusions are drawn.

3.2.2. Snakes

The traditional snake model is based on the energy expression:

$$E = \int_0^1 \frac{1}{2}\left[\alpha\left\|\boldsymbol{x}'(s)\right\|^2 + \beta\left\|\boldsymbol{x}''(s)\right\|^2\right] + E_{ext}(\boldsymbol{x}(s))ds\,, \tag{3.1}$$

where $\boldsymbol{x}(s) = [x(s), y(s)]$, and $s \in [0,1]$. The terms α, β control the snake's tension and rigidity respectively, while $\boldsymbol{x}'$, $\boldsymbol{x}''$ are the first and second order derivatives of $\boldsymbol{x}(s)$in respect of s. The terms inside the square brackets represent internal forces, regulated by the snake physics, while the other term is the external potential, derived from the image in order to be minimum near interesting features. Internal forces work against stretching (elasticity α) and deformation (rigidity β), while the external force attracts the snake towards the desired features.

In order to produce a vector field, the energy in (3.1) has to be minimized. This can be obtained by satisfying Euler's equation and considering the deformable contour dynamically, by treating $\boldsymbol{x}$ as time-dependant aside space-dependant: $\boldsymbol{x} = \boldsymbol{x}(s,t)$. We obtain:

$$\boldsymbol{x}_t(s,t) = \alpha\boldsymbol{x}''(s,t) - \beta\boldsymbol{x}''''(s,t) - \nabla E_{ext}\,. \tag{3.2}$$

With the convergence of $\boldsymbol{x}(s,t)$ the term $\boldsymbol{x}_t(s,t)$ tends to, thus solving (3.2). Such method has three main disadvantages: the convergence to concavities is hard, it is sensible to the snake initialization due to limited extension of the vector field and a criterion is required in order to halt iterations.

Several techniques have been proposed in order to solve such drawbacks. Some examples are the introduction of the *Gradient Vector Flow* (GVF) and generalized GVF by Prince and Xu [Xu+M98, Xu+D98] to solve the problem of convergence. The dependency on initialization was treated with the introduction of a distance-force potential field [Coh+93]. Finally, conditions to stop the snake iterations were provided with *simulated annealing* [COUR].

In this work, the generalized GVF is considered. The GVF is defined as the vector field $\boldsymbol{v}(x,y) = (u(x,y), v(x,y))$ that minimizes the energy expression:

$$E = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\boldsymbol{v} - \nabla f|^2 \, dxdy \,, \tag{3.3}$$

where $f(x,y)$, is the edge map derived from the original image $I(x,y)$. It can be observed that when $|\nabla f|$ is small, the partial derivative term prevails. Then, in order to minimize the integral, such term has to be small. In other words, we are imposing that in homogeneous areas, the field presents slow variation. On the other hand, when $|\nabla f|$ is large, the term $|\boldsymbol{v} - \nabla f|$ prevails, so that minimizing E means setting $\boldsymbol{v} = \nabla f$. Wherever the image shows interesting features, the vector field is set to be strong and orthogonal to the features. The parameter μ is used to balance the previously described terms and should be proportional to the noise (stronger noise $\rightarrow$ larger μ).

Operatively, in order to compute the vector field the generalized GVF [Xu+M98] has been chosen to avoid a smoothing effect in the proximity of borders:

$$\begin{aligned} u_t &= g(\|\nabla f\|)\nabla^2 u - h(\|\nabla f\|)(u - f_x) \\ v_t &= g(\|\nabla f\|)\nabla^2 v - h(\|\nabla f\|)(v - f_y) \end{aligned} \,, \tag{3.4}$$

where $g(\|\nabla f\|) = e^{-(\|\nabla f\|/K)}$ and $h(\|\nabla f\|) = 1 - g(\|\nabla f\|)$. Once the field has been computed, the snake contour is initialized near to the desired object. It is subsequently deformed in order to follow the field potentials by iteratively shifting and interpolating its points. Such iterative deformation process is halted by setting a threshold to the average differential distance of the nodes movements. Since the vector field not always has null components corresponding to edge features, the snake nodes may not reach stability even when the contour reaches the exact edge. To avoid such situation, the field components matching the desired feature are automatically set to zero.

The snake's deformation is controlled by the parameters α and β. Since these parameters affect the contour smoothness and rigidity, their values may depend on the considered application and the image overall quality. Large values of tension and rigidity, for instance, produce smoother contours and eventually avoid convergence to wrong features produced by noise, but may prevent convergence to correct weak edges. For such reasons, a good preprocessing stage is strongly required and is discussed in Section 3.2.4. Nonetheless, small values of α and β have been generally used in this work, *i.e.* $\alpha = 0.02$ and $\beta = 0.01$.

3.2.3. Processing Structure

Active contours have proven particularly efficient for medical segmentation, offering numerous advantages over other techniques. In fact, although relatively complex, they provide an intuitive interaction choice for initialization and produce a vector representation of the segmented object. In order to achieve full 3D segmentation through a snake based method, a possible solution is to extend the snake model to 3D. This consists in defining an active mesh, instead of contour, and extending the solution of (3.1) to 3D:

$$\begin{aligned} u_t &= g(\|\nabla f\|)\nabla^2 u - h(\|\nabla f\|)(u - f_x) \\ v_t &= g(\|\nabla f\|)\nabla^2 v - h(\|\nabla f\|)(v - f_y) \\ w_t &= g(\|\nabla f\|)\nabla^2 w - h(\|\nabla f\|)(w - f_z) \end{aligned} \quad , \qquad (3.5)$$

being $\mathbf{v}(x,y,z) = (u(x,y,z), v(x,y,z), w(x,y,z))$ the vector field. The main drawback is computational complexity. The allocation and manipulation of 3D vector matrices is required for such solution. As a comparison, the computational complexity of a 3D method corresponds to that of the slice-based implementation to the power of 2, taking into account the preprocessing stage, field generation and manipulation and the snake deformation. Furthermore, the space gradient operator may not bring to relevant results, since volumetric data resolution is often not isotropic along x, y, and z axes. Finally, user interaction requires the initialization of a starting 3D mesh, which is less intuitive than defining a 2D curve.

For such reasons, a slice-based approach has been preferred. The method consists in choosing a starting significant slice and initializing the active contour. Once the planar snake converges, its contour coordinates are considered as starting point for the processing of adjacent slices. Through such approach, computational complexity is limited, since only two or three 2D buffers have to be allocated and the field computation and contour deformation are performed in 2D. On the other hand, the method is not appropriate for intricate objects, such as intestine convolutions. This is a limitation for the proposed algorithm, but it does not undermine its usefulness in many applications where a compact ROI needs to be determined.

In order to iterate the processing, a method for predicting snake point positions between adjacent slices is required. By identifying with *snake processing* the collection of field computation and contour deformation, the algorithm is defined as follows:

- Snake initialization in the chosen slice
- Snake processing in the chosen slice
- Iterate:
    - o Snake contour prediction in adjacent slice
    - o Snake processing in the adjacent slice

A linear motion-detection algorithm has been implemented and tested to predict contour points position. Assuming that image sampling along the slice direction is not too coarse, the interesting features can be tracked between adjacent slices. Although inherently different, slices are treated like frames. Objects discovered in one frame is expected to be found somewhere in adjacent frames, not far from its previous position. Such determination allows for point translation, thus object deformation. The prediction is not exact, but is expected to lead towards desired features. The proposed procedure is illustrated in Fig. 3.1. To speed up the processing, the search method is designed to scan pixels along lines instead of areas.

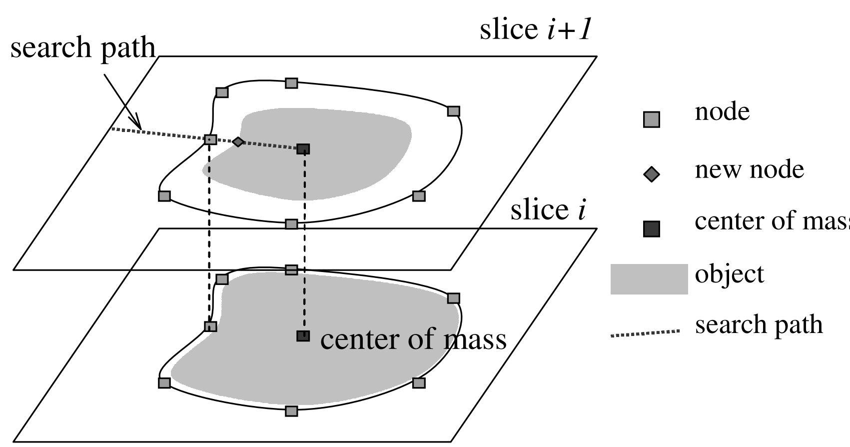


Fig. 3.1. Linear motion detection scheme.

The center of mass of the object in slice i is first computed, while its snake points are projected onto slice $i+1$. Then, for each point, the new node position is sought among those pixels belonging to the segment that connects the previous center of mass with the considered node. A distance metric, computed from luminance value and edge steepness in a surrounding of the current pixel, is used to choose between candidate nodes. Two examples of search patterns with different node density are shown in Fig. 3.2.

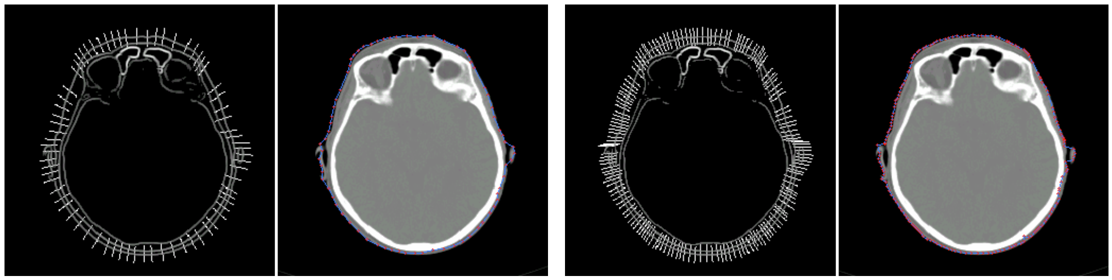


Fig. 3.2. Search patterns for the linear motion detection algorithm.

3.2.4. Preprocessing

It has been noticed how the snake evolution depends on the vector field which, in turn, is the result of significant features extraction from the input image. This last aspect is then essential for the success of segmentation and much attention must be paid to it. The goal is to obtain an optimal image of the desired features, in order to set the snakes parameters in a general sense and predict and control the snake evolution. Since the most important features for characterizing the image are its edges, the preprocessing stage is meant to generate a complete and reliable edge map from the given data. Two main blocks are then required: a low-pass filtering needed to suppress noise components, useless details and mask any shading effect and an edge detection algorithm that extracts the edge features neither producing over detection of edges nor missing any significant edge.

The first preprocessing attempt follows a classic scheme, consisting in a smoothing filter followed by edge detection and binarization. In particular, several techniques have been tested. The Gaussian smoothing filter with $\sigma = 2$ has been used as simple LP filter (3.6), while the best edge detection algorithm was provided by the Prewitt compass kernel (3.7), followed by Otsu's threshold prediction [Ots79].

$$W = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \begin{bmatrix} w(-1,-1) & w(-1,0) & w(-1,1) \\ w(0,-1) & w(0,0) & w(0,1) \\ w(1,-1) & w(1,0) & w(1,1) \end{bmatrix}; \qquad (3.6)$$

$$w(s,t) = \frac{1}{2\pi\sigma^2} e^{-[s^2+t^2]/2\sigma^2}$$

$$H_0 = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}, \quad H_{45} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} \qquad (3.7)$$

Although the vector field produced after these operations appeared regular and smooth, the resulting snake was not acceptable since it did not follow the region contour. The main reason for such behavior is that edges are thickened both due to the Gaussian filter and the low threshold level, so that convergence is not granted or double ridges are produced. The thickening effect increases with higher σ, while low threshold levels are required in order not to discard of any interesting edges. For such limitations, this method is not sufficient for the task.

At first stage, the local maxima map extracted from the image gradient has been considered in order to avoid the previous drawbacks. Further improvement can be obtained by considering more advanced preprocessing filters. In fact, the application of a non-linear LP filter results in considerable simplification of the original image, leading to a better starting point for the processing. An adaptive bilateral filter has been chosen for the task. Such filter is based on the observation that the kernel coefficients can be weighted not only by their spatial relationship, but also by their photometric distance [Tom+98]. Given the starting image, $f[x, y]$, the filtered pixel is then:

$$g[x, y] = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w \cdot f[x+s, y+t] \Big/ \sum_{s=-a}^{a} \sum_{t=-b}^{b} w, \qquad (3.8)$$

where:

$$w = w_d[(x, y), (x+s, y+t)] \cdot w_r[f[x+s, y+t], f[x, y]] = e^{-\frac{s^2+t^2}{2\sigma_d^2}} e^{-\frac{(f[x+s, y+t]-f[x, y])^2}{2\sigma_r^2}}. \qquad (3.9)$$

The kernel shape changes according to the image in the application point. The results are comparable with those of most anisotropic filters. However, high complexity is avoided, since no iterative processing is required to compute the filtering mask. The appropriate setting of σ_d and σ_r allows for different degrees of smoothing. The filter preserves edges and detail structures and allows for further image simplification. The preprocessing stage is then:

- Bilateral filtering with: $\sigma_d = 3$, $\sigma_r = 5$;
- Bilateral filtering with: $\sigma_d = 3$, $\sigma_r = 15$;
- Prewitt-compass gradient computation;
- Binarization (Otsu's threshold determination);
- Local maxima computation.

The first step of bilateral filtering is used to remove noise without losing edge information, while the second for normalizing homogeneous regions. Results are shown in Fig. 3.3.

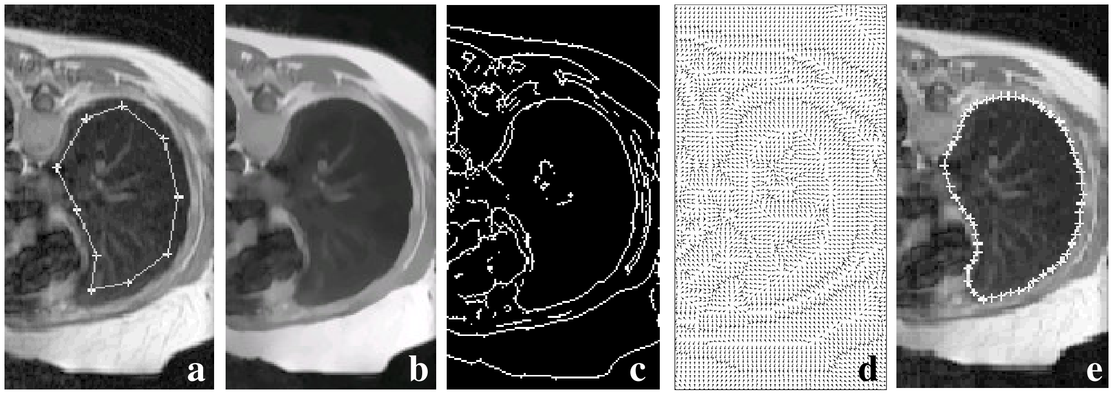


Fig. 3.3. Preprocessing: a) original image with snake initialization, b) adaptive bilateral filter, c) edge detection, d) vector field, e) extracted snake.

3.2.5. Results

The proposed algorithm has been tested on four medical volumetric datasets (Table 3.I).

TABLE 3.I. Medical datasets used for experiments

Name	*Description*	*Volume size*
CT_skull	Tripod fracture	256x256x192
CT_carotid	Internal carotid dissection	256x256x64
CT_foot	Foot - multiple fractures	384x384x41
MR_liver_t2	Normal liver	256x256x48

The segmentation of the brain structure and outer skull from the *CT_skull* dataset is shown in Fig. 3.4. Since the proposed algorithm provides a vector description of the segmented object, such information can be used for mesh generation. This feature is particularly interesting for representation and shape information compression purposes. In Fig. 3.4, the details of three slices are represented.

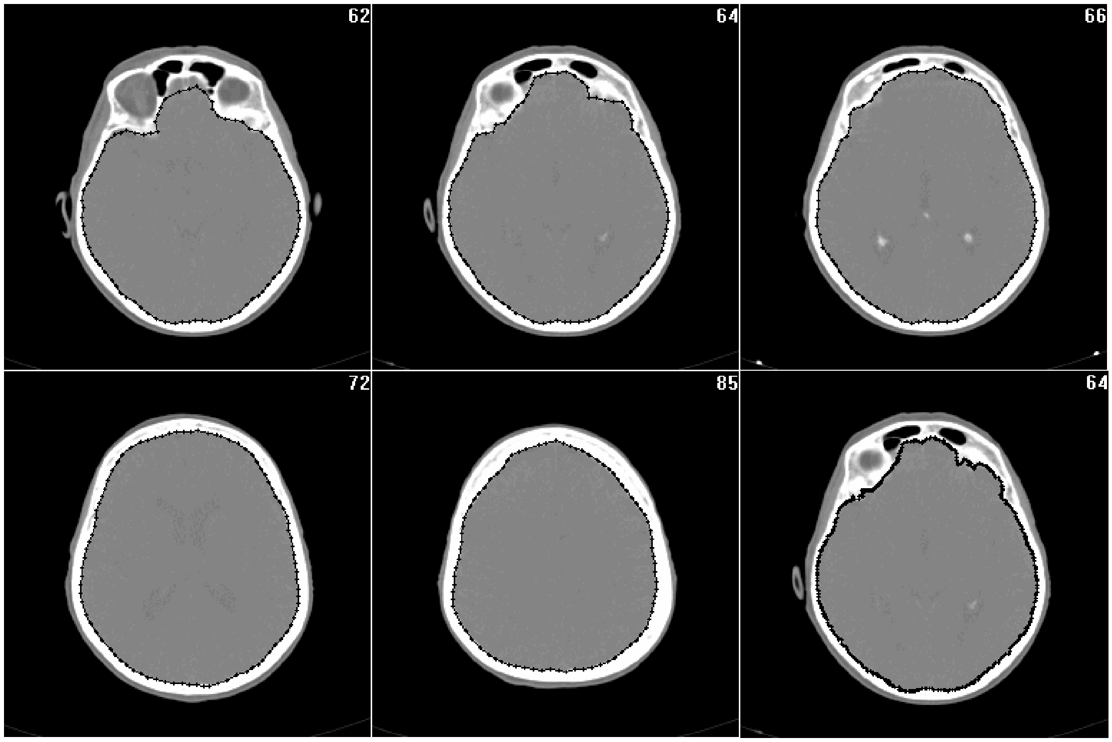


Fig. 3.4. Processing applied to *CT_skull*. The first five slices result from setting 3÷5 pixels inter-node distance. Last image represents slice 64 computed with 1÷2 pixels inter-node distance.

A slice view of *MR_liver_t2* is provided in Fig. 3.5. In this case, the extraction of the hypocondrium is achieved. Such application is well representative of a typical ROI segmentation task for subsequent object-based or selective volumetric coding.

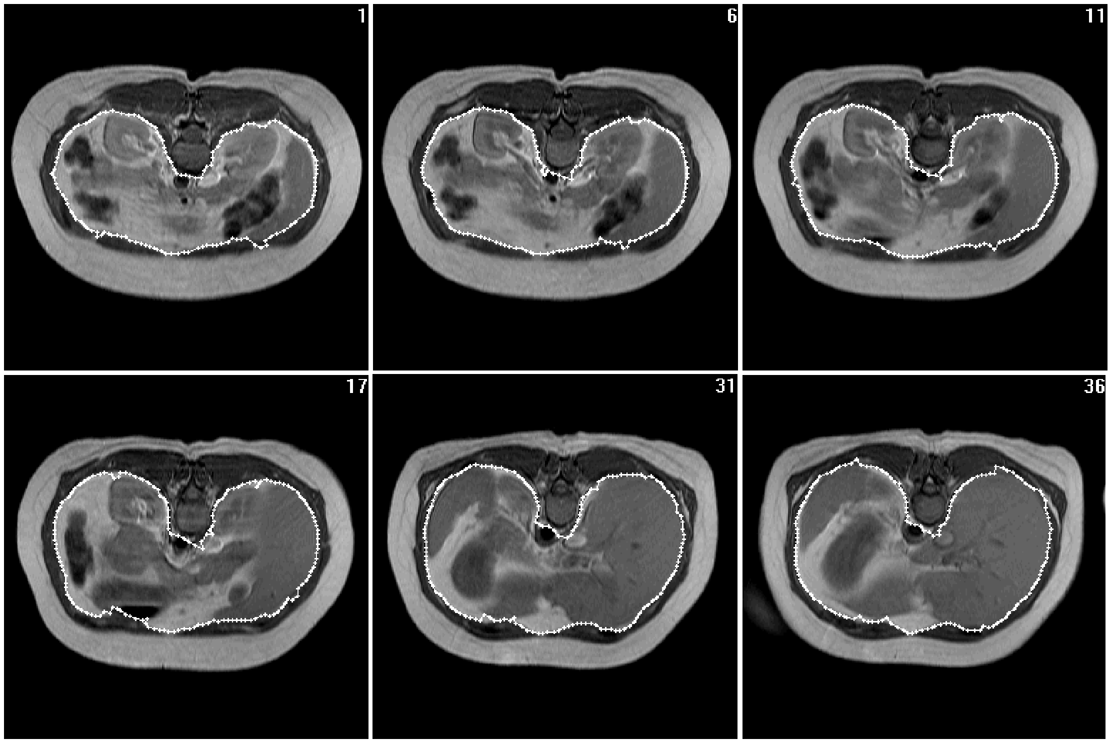


Fig. 3.5. Processing applied to *MR_liver_t2*.

Finally, in Fig. 3.6, the segmentation of the right foot from the background is considered. The first image was obtained by equalizing the original image in order to enhance the background noise. The proposed method is able to follow the desired feature regardless of the strong noise from the nearby object.

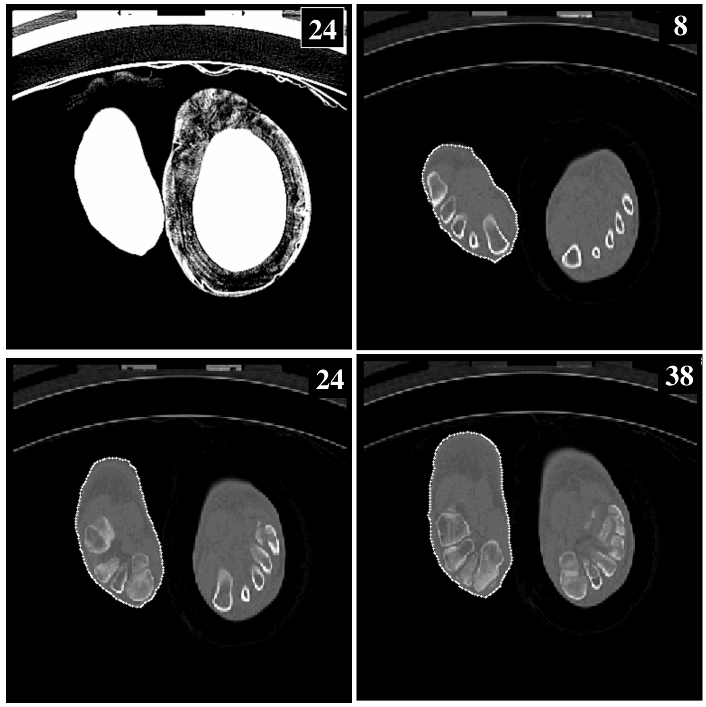


Fig. 3.6. Processing applied to *CT_foot*

3.2.6. Conclusions

In this chapter we proposed a new scheme for region of interest segmentation of volumetric image data. A slice-based scheme based on active contours has been designed and implemented. The preprocessing stage has been extensively discussed to reach a satisfactory starting point. Testing of the algorithm has been carried out on different medical datasets and has shown interesting results. The algorithm is particularly fit for ROI determination in volumetric coding frameworks.

3.3 SRG-based Approach

3.3.1 Introduction

The Seeded Region Growing algorithm, introduced by Adams and Bischof [Ada+94], is possibly one of the most successful region-based segmentation methodologies, *i.e.* it consists in the direct individuation of significant regions from the given image. Such methods, differently from edge detection based algorithms, such as active contours, are based on neighbor pixel grouping supported by a homogeneity criterion (Fig. 3.7).

Being R the entire region, the segmentation process may be described as the partitioning of R into n sub-regions, $R_1, R_2, \ldots, R_n$, so that:

a. $\bigcup_{i=1}^{n} R_i = R$;

b. R_i are **connected regions**, with $i = 1,2,\ldots,n$

c. $Ri \cap Rj = \emptyset$, $\forall i, j \, : \, i \neq j$, i.e. regions must be **disjoint**;

d. $P(R_i) = TRUE$, with $i = 1,2,\ldots,n$;

e. $P(Ri \cup Rj) = FALSE$, $\forall i, j \, : \, i \neq j$;

where $P(R_i)$ is a logical operator defined on R_i, and $\emptyset$ is the empty ensemble. The first definition, also called the completeness segmentation condition, guarantees that the partitioning process results in the full region classification of the starting scene. The rule c. guarantees that each image pixel is assigned to one region only, while d. and e. state that a given property must be satisfied by all pixels of a given region, *e.g.* all pixels must have the same gray level value, and that such property must not be verified when considering two different regions. Such logical operator is called *homogeneity criterion*.

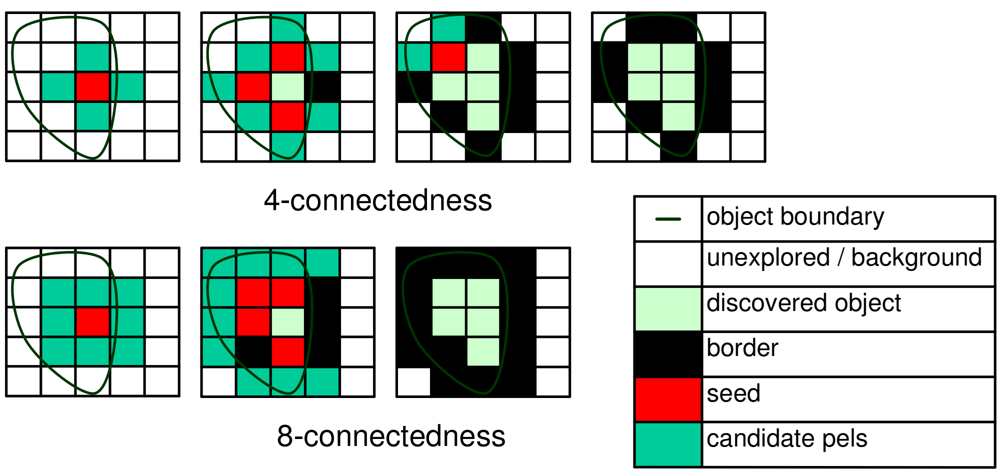


Fig.3.7. Evolution of the region growing mechanism in the case of 4- and 8-connectedness.

The homogeneity criterion is often based on simple considerations, such as the definition of a threshold value between pixels belonging to the same region. While, the choice for the threshold value may be problematic, since image-dependent, the final result is generally very sensitive to noise.

The Seeded Region Growing algorithm may be seen as a subclass of Region Growing methods, with the definition of “seeds” to distinguish between significant and insignificant regions from the image. It will be shown that SRG is particularly fit for multi-dimensional implementation, because of the possibility to easily extend the 2D case based on symmetry.

3.3.2 Seeded Region Growing

The growing mechanism is controlled by the choice of a small number of starting points, called *seeds*. Such points may be selected by the user, resulting into a semi-automatic segmentation method, or automatically, through the definition of distinctive characteristics for the interesting features.

In both cases, the fist step is the selection of the seeds, grouped into n clusters, $A_1, A_2, \dots, A_n$. Each cluster is generally formed by one seed only, but may include several seeds. During each iteration step, one pixel is added to one of the clusters, A_i, $i = 1, \dots, n$. If we consider the system after m iterations and call T the collection of pixels which have not been assigned to any cluster but are adjacent to one of them, we have:

$$T = \left\{ x \notin \bigcup_{i=1}^{n} A_i \wedge \exists k : N(x) \cap A_k \neq \emptyset \right\}, \tag{3.10}$$

where $N(x)$ is the collection of pixels, which are adjacent to the point x (Fig. 3.8). At each step, the algorithm considers one pixel from T and assigns it to one of the existing neighboring regions A_i. Then, a similarity distance is computed between each point of $N(x)$ and the neighboring clusters, in order to add each entry to T in order of importance. A typical distance is:

$$\delta\ (x, A_i) = \left| f(x) - \frac{1}{N_i} \sum_{j=1}^{N_i} f(y_j) \right|, \quad \text{with} \quad y \in A_i(x), i = 1,2,\dots,n, \tag{3.11}$$

where $f(x)$ is the image value at pixel x. Equation (3.11) evaluates the absolute difference between the gray level value of x and the average gray level value of cluster A_i.

The growing process consists in finding a point $z \in T$ and the corresponding cluster A_i that minimizes the distance metric (3.11):

$$\delta(z, A_i) = \min_{x \in T, k \in [1,n]} \{\delta(x, A_k)\}. \tag{3.12}$$

If the distance value $\delta(z, A_i)$ is less than a given threshold, th, the pixel is added to cluster A_j, otherwise another cluster is chosen so that:

$$A = \arg\min_{A_k} \{\delta(z, A_k)\}. \tag{3.13}$$

When $\delta(z, A) < th$ is verified, the pixel z is added to the cluster A. If neither (3.12) nor (3.13) are verified by any existing cluster, then the considered pixel is significantly different from any neighboring cluster and a new cluster, A_{n+1}, is created. All cases require the update of the cluster statistics, while the set T must be updated continuously.

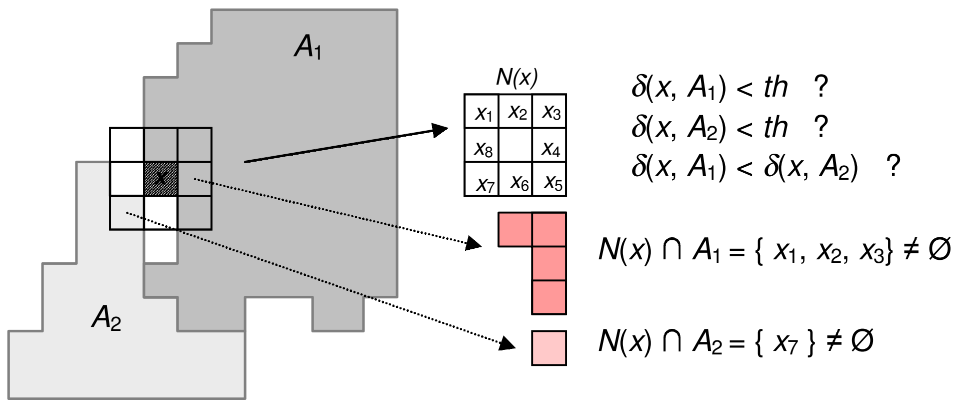


Fig. 3.8. SRG algorithm behavior.

Some critical aspects of SRG are:

1. dependency from the starting seed position: in case of noisy images, the seeds may correspond to corrupted pixels and may alter the statistics for the given cluster, resulting in false segmentation;
2. relevance of the order chosen for considering the candidate points from the set T: by choosing a different order than that of minimum similarity distance, *e. g.* random order, the cluster statistics may be significantly altered, resulting in false segmentation.

The first effect is typical of the SRG behavior. In fact, SRG tends to polarize the search&add process towards regions that are discovered during the starting phase. This problem can be solved by performing two runs. The fist rough segmentation is used to acquire region information. Subsequently, a second segmentation process is performed to refine results based on the previous step data. Moreover, the adoption of a preprocessing stage based on a piecewise smoothing algorithm allows to minimize errors from noise artifact. Anisotropic or adaptive filters [Lin+01] are particularly fit for such purpose, since they allow for homogeneous region smoothing while preserving significant edges and details.

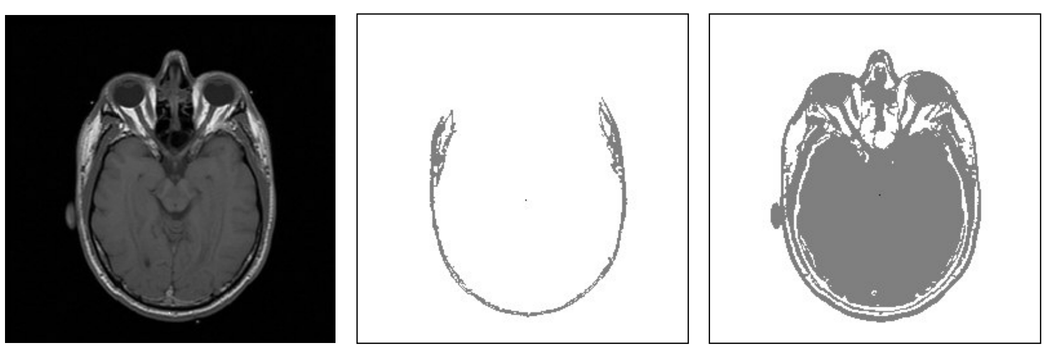
The second effect implies that the set T must be strictly ordered, based on the minimum similarity distance between each point x and its neighboring clusters. Each time a candidate point is added to its nearest cluster, the cluster statistics vary. Then, the distance between the points in the set T and all the clusters should be updated at each processing step, in order to renew the set order. However, R. Adams and L. Bischof [Ada+94] found that result variations may be negligible among near iterations. Moreover, while such continuous update is not particularly expensive in terms of computational cost in the 2D case, it may become challenging in the 3D case.

It must be considered that the proposed method requires the user-guided selection of the initial seeds and is thus subjected to errors. Automatic seed discovery may be implemented by a watershed-like system, but the user interaction might be required anyway through the selection of one or more seeds among those automatically determined anyway. Similarly, the determination of the threshold value th is a critical task, which can be automated by considering the histogram characteristics and fixing a tolerance level on the local variance value. In general, the process of automatic segmentation can be efficiently achieved only for specific tasks, while generality implies the adoption of semi-automatic systems and the registration of parameters for each application case.

The SRG algorithm considered here is optimized for single region segmentation. This is done for both application necessity and minimization of computational cost. In this case, a single cluster, A, is considered with its average gray level value. The distance metric becomes then:

$$\delta\ (x, A) < th\ . \tag{3.14}$$

An example of such simple SRG segmentation is shown in Fig. 3.9 (a,b). In this case, the threshold value has been manually fixed to 170 for the cranial bone region. It can be noticed that the same value applied to the brain region results in severe over-segmentation (Fig. 3.9c). In this case, an acceptable result may be obtained with a smaller threshold value, $th = 45$, as shown in Fig. 3.10. This behavior evidences the need for both an automatic threshold determination algorithm and a preprocessing stage for minimizing dependency of results on noise artifacts and local discontinuities. In fact, over-segmentation can be observed in Fig. 3.10 also, possibly due to the presence of continuity paths.



a b c Fig. 3.9. Example of single cluster SRG segmentation with *th*= 170: original image (a), bone segmentation (b), attempt of brain segmentation (c).

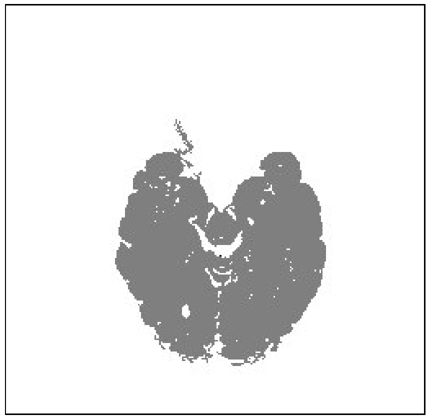


Fig. 3.10. SRG brain segmentation from Fig. 3.8a with th = 45.

Two methods for optimizing the algorithm are then:

1. the adoption of a preprocessing stage to guarantee clear image contours and to correct false continuity paths. Although this task may not be completely solved, adaptive LP filters are a good approximation of the required solution, thanks to their property of spatial-adaptivity and edge-preserving smoothing capability. In particular, bilateral filtering is mostly indicated for the task. The result of SRG segmentation with bilateral filter preprocessing is show in Fig. 3.11. It can be noticed that the object of interest, *i.e.* the brain, appears well defined and free from over-segmentation artifacts. However, the algorithm is still strongly dependent on the threshold value. By observing Fig. 3.11, for instance, it can be shown that cranial bone segmentation is subjected to several discontinuities, which require the adoption of a high and precise threshold level. Such undesirable effect may be minimized by considering the edge information and extending the algorithm to 3D for volumetric image data, as will be discussed in Section 3.3.3.
2. the extension of the homogeneity criterion with other considerations, such as local gradient information. This change helps avoiding the case of spurious additions, occurring when the candidate point presents a noticeable border, but its gray level value is not sufficiently different from the average value of the candidate cluster. In Fig. 3.13 is shown the segmentation result of Fig. 3.9a with the new condition:

$$\delta(x, A) + \mu \cdot |\nabla f| < th_c . \tag{3.15}$$

It can be noticed (Fig. 3.13) that the segmented object is better defined than in the previous segmentation attempt. In particular, the spurious appendix visible in the upper-left part of the image of Fig. 3.11, is not included in the new result. Moreover, the described modifications result in the increase of the acceptable threshold range for obtaining an satisfactory result.

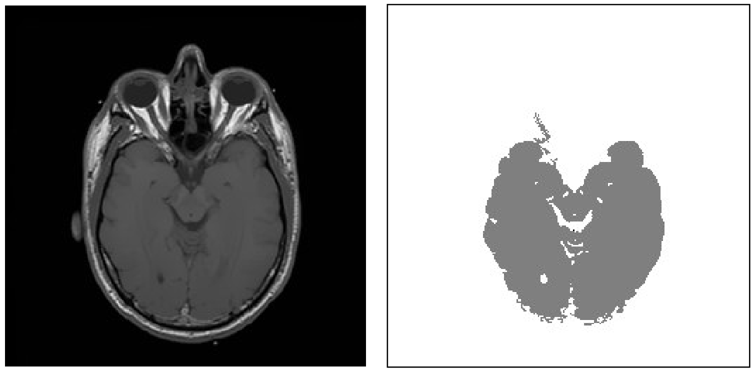


Fig. 3.11. SRG brain segmentation after preprocessing with bilateral filter; from left: preprocessed image and final result.

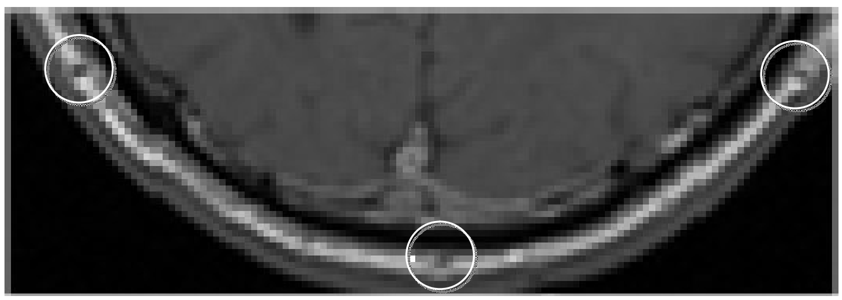


Fig. 3.12. Cranial bone detail.

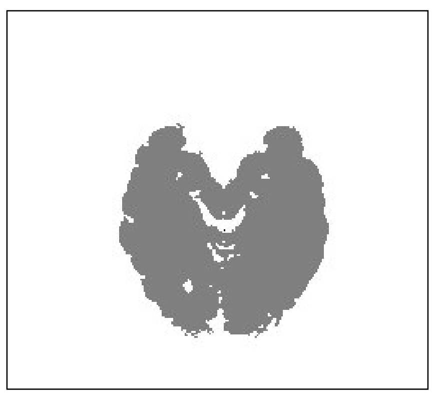


Fig. 3.13. SRG segmentation with gradient information and pre-processing.

3.3.3 Extension to 3D

Thanks to its relatively simple implementation, seeded region growing can be effectively developed and applied to volumetric image data. In particular, 3D segmentation may be performed by extending the search mechanism, the statistics and the candidate list to the third dimension. The similarity metrics remain identical (3.13, 3.14, 3.15), except for the new spatial dimension. Moreover, extending the algorithm to 3D generally results in a more reliable segmentation process, which may solve continuity errors, as those depicted in Fig.

3.12. In fact, in the 2D case even a single discontinuity may interrupt the object connectedness and may block the correct evolution of the candidate pixel search. In the 3D case, such event is considerably less probable, since the connectedness rule is extended to the adjacent slices.

As a first attempt, the *head* dataset has been segmented with the plain SRG algorithm, without any preprocessing or any consideration on the gradient. Results are shown in Fig. 3.14. The main limitation in this case comes from the arbitrary choice of the threshold value, th. With high threshold values, spurious regions are mistakenly added to the segmented object, while low threshold values bring to under-segmentation (Fig. 3.15).

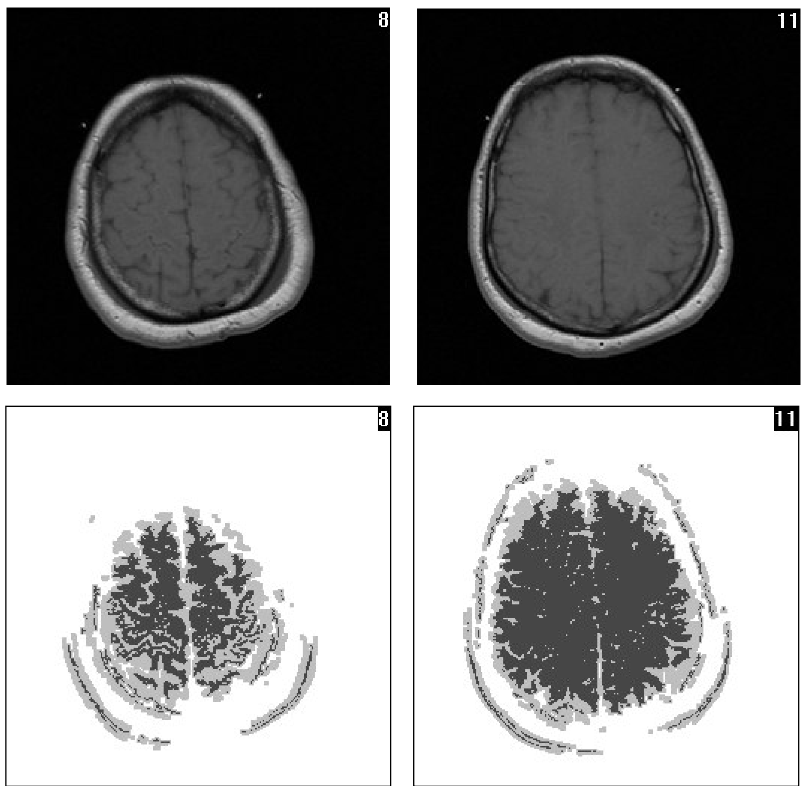


Fig. 3.14. 3D SRG segmentation: original slices (top), and corresponding segmented results (bottom) with th = 20.

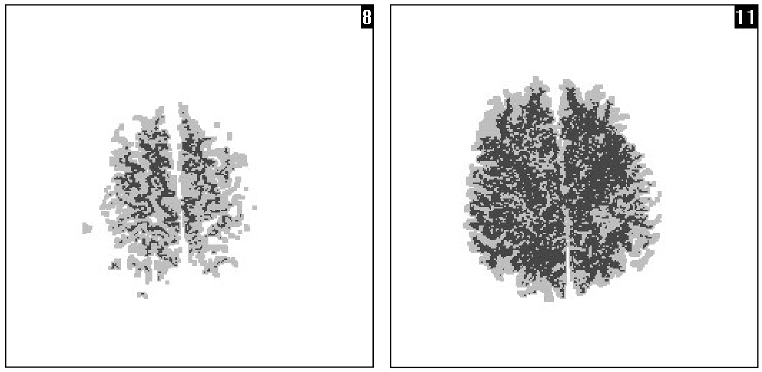


Fig. 3.15. 3D SRG segmentation with th = 10.

Better results may be obtained by applying the bilateral preprocessing filter. Several improper details which were previously added are now excluded from the segmented object. However, the result is still unacceptable.

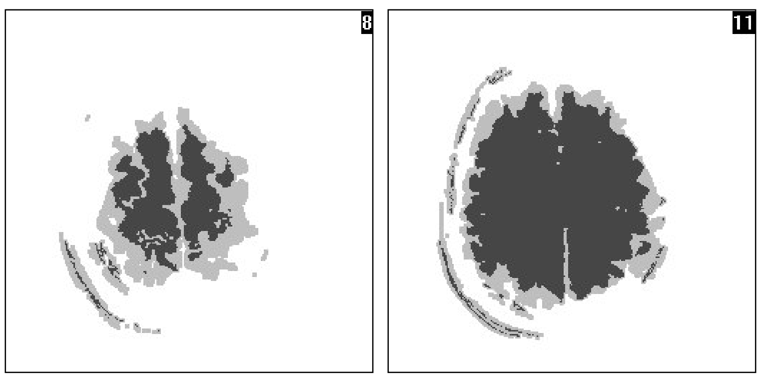


Fig. 3.16. Results of SRG segmentation and bilateral filtering.

Finally, the candidate pixel gradient is considered and added to the similarity condition as in (3.15). Results for several slices are shown in Fig. 3.17.

As mentioned earlier, the bone segmentation is generally more difficult if considered for each plane than in the 3D case. In fact, it can be noticed that the segmented object is free from artifacts or spurious parts (Fig. 3.18 and 3.19).

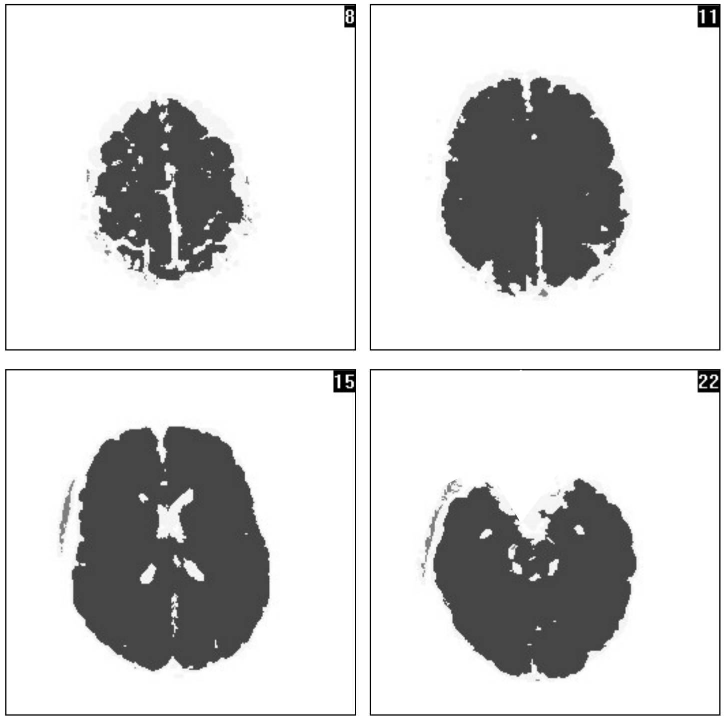


Fig. 3.17. 3D SRG segmentation with gradient information.

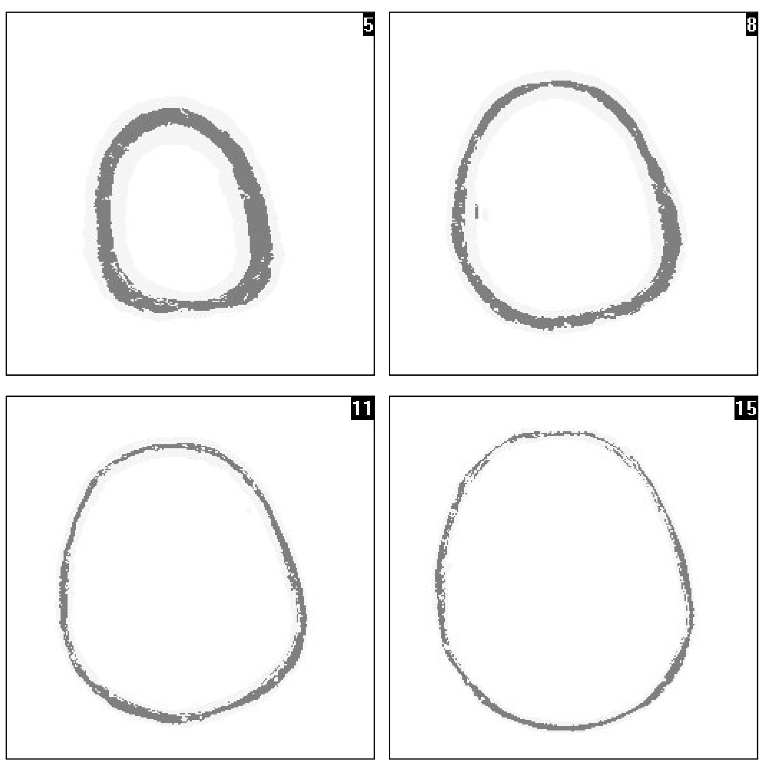


Fig. 3.18. Example of SRG segmentation with cranium gradient.

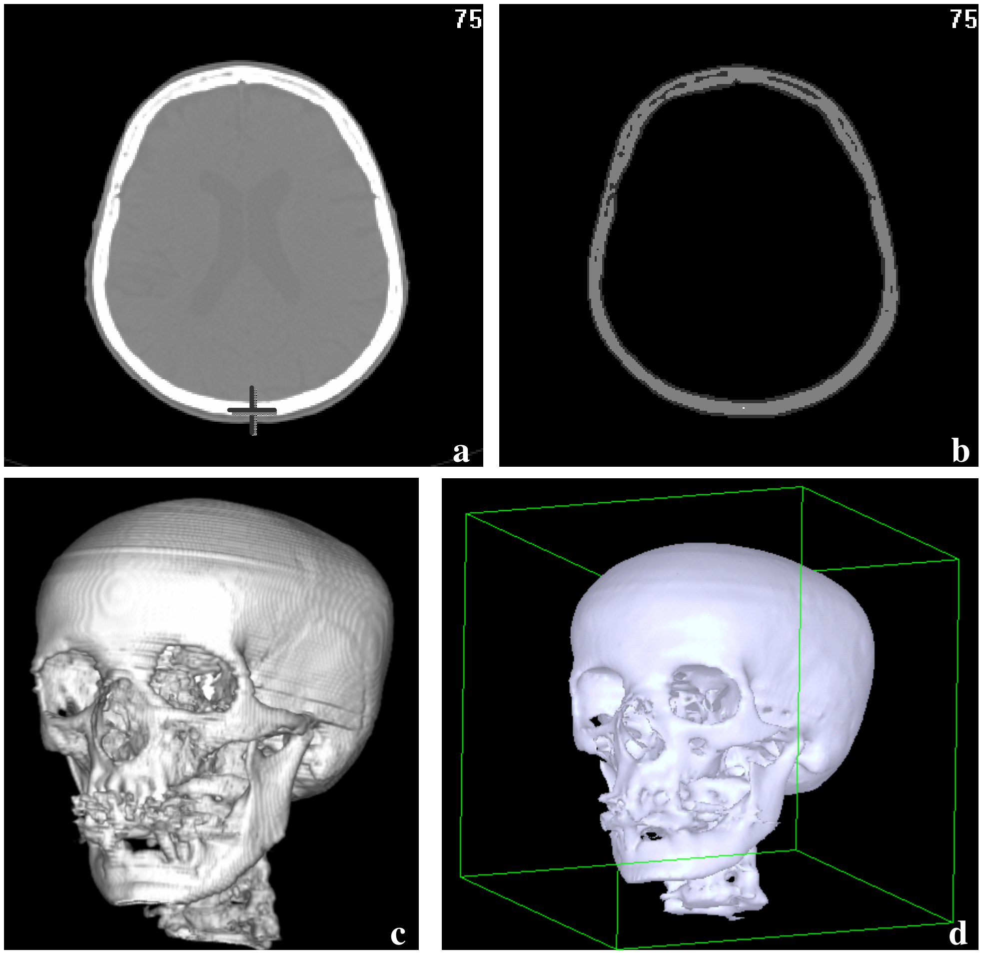


Fig. 3.19. Example of SRG segmentation and volumetric rendering. Original slice data with seed selection (a); slice view (b) and volume rendering of the SRG result (c); extraction of smooth 3D mesh and surface rendering (d).

Chapter 4

Semantic-driven Volume Data Compression with ROI

4.1 Introduction

In the last years, volumetric coding has been receiving increasing attention, thanks to its importance in the medical and scientific field. Medical instrumentation and applications, in particular, make extensive use of volumetric imagery, both for diagnosis and computer assisted surgery. Devices such as CT, MR and PET scanners produce volumetric data as natural output, together with optical probes, such as confocal microscopes, ultrasound probes and others. Despite such wide utilization, and the great storage requirements of volumetric data, efficient volumetric data coding has not been yet fully adopted in the field, and classical lossless techniques are generally used for storage.

Several works have addressed the problem of efficient volumetric data coding from different points of view and purposes. Xiong *et al.* [Xio+03] proposed a modified version of 3D SPIHT and 3D ESCOT. Schelkens *et al.* [Sch+03] gave an overview of several techniques and proposed a new method based on quad-tree and block-based coding. Bilgin *et al.* [Bil+00] introduced a scheme based on 3D zero-tree coding. Kim *et al.* [Kim+99] employed a slice-based subdivision of the volumetric image and tested several integer wavelet kernels together with the 3D SPIHT algorithm. Finally, examples of object-based methods are the works from Ichimura [Ich98] and Menegaz *et al.* [Men+02].

This work proposes a novel scheme for object-based lossy to lossless volumetric data coding. The proposed method is particularly intended for medical data, such as magnetic resonance images, but is well suited for other applications.

The main objective of this research is the design of a coding scheme that takes into account the semantic characteristics of volumetric data. Since such data often represent the objects of interest, *i.e.* a portion of a body scan, together with an extended surrounding region of air, a preprocessing stage is adopted in order to automatically generate a mask for background identification. In fact, while the background should ideally present a uniform value, due to several noise sources and possible reconstruction faults, it often contributes to the signal energy in a significant way. This results in reduced coding performance, especially in the lossless case, when transform coefficients are reconstructed until the least significant bit-plane. Then, mask has the purpose of background suppression for providing an input image as ideal as possible.

Another useful aspect of volumetric data is the frequent presence of a region of predominant significance for the chosen application (Region Of Interest). In general, medical data portrait a region that is larger than the actual volume of interest. A typical example is, for instance, the MR scan of a patient's head in order to acquire the brain information. These considerations support the idea of adopting lossy techniques for the less significant information, while preserving lossless representation for the volume of interest. The generation of the ROI is out of the scope of this work and will not be discussed in the following. An adaptive seeded region growing algorithm has been used to produce the significant ROI. The terms 'object of interest' or 'ROI', 'object' and 'background' will be used to indicate the region of interest, its surrounding relevant area and the irrelevant noisy background respectively.

The coder is derived from 3D-SPIHTp [Gin+04], with the introduction of multiple independent 3D-ROI features and optimization of the coding stage. Despite the need for knowing the ROI geometry, such choice offers several advantages over max-shift methods. First, it allows for independent coding and retrieval of object and background. Then, it is suited for parallel implementation. Finally, ROI geometry information can be easily compressed by vectorizing the ROI mask and transmitting its entropy compressed Freeman code. Absence of artifacts along object borders is guaranteed through the extension of the ROI mask in the wavelet domain. Lossy to lossless coding is guaranteed by the adoption of integer lifting wavelet transform and the 3D-SPIHTp core. Through such scheme, the proposed coder is able to independently code and transmit several regions of the data, each with its own characteristics, *i.e.* complete masking, lossy bitrate or lossless coding. Moreover, since each object-stream is fully scalable and its reconstruction can be carried out bit by bit, interleaving techniques may be adopted in order to provide a composite scalable stream.

4.2. Preprocessing

Two different techniques have been tested for the task of background identification. The first method consists in a preprocessing stage applied to the original volume data. Two automatic threshold levels, th and th_c, are first computed with the Otsu's algorithm [Ots79] for the original image and the contour map respectively. Then, an iterative process is employed in order to identify each separate object. Iterations are controlled by the presence of objects and the compactness of the discovered object. At each step, a foreground mask is computed by deforming the image borders according to th and th_c. The discovered mask is labeled and the vector contour (VC) of the largest discovered object is computed together with its compactness. Once the object is accepted, its shape is deleted from the volume and its VC is further processed for deleting discontinuities. Finally, the VC is assumed as part of the foreground. The processing steps are summarized in Fig. 4.1.

```
Compute th and th_c;
Median filter of original volume;
While(objects) {
        While (compactness ≥ c*) {
                Create mask(th, th_c);
                Objects identification and labeling;
                Create VC for largest object;
                If(compactness ≥ c*) decrease th; }
        Delete object from volume;
        Eliminate discontinuities from VC;
        Assume VC as part of foreground; }
```

Fig. 4.1. Preprocessing scheme.

The second method follows similar principles, but is directly built inside the coder. In fact, instead of considering the original data and applying a preprocessing stage, the largest 3D packet wavelet subbands are considered for background identification. In particular, subbands LH_1, HL_1 and HH_1 of the first z decomposition level are summed in absolute value to obtain the starting volume data. This approach offers several advantages. First, is can be implemented inside the coder instead of adding an external block. Then, the reference volume to be processed for background detection has a size of 1/16 if compared to the original data. Moreover, the scaling of the background mask consists in fewer steps, since the initial mask overlaps by design with the first 3D packet wavelet levels. On the other hand, the background identification task may be complex, since the composition of the first wavelet subbands results in a rather noisy contour image. For such reason, a more accurate preprocessing is used. A wider-neighborhood adaptive median filter has been chosen, so achieving good results (Fig. 4.2). In this case, the scheme in Fig. 4.1 differs for the substitution of the median filter with its adaptive version, and the computation of th_c, which is not performed anymore.

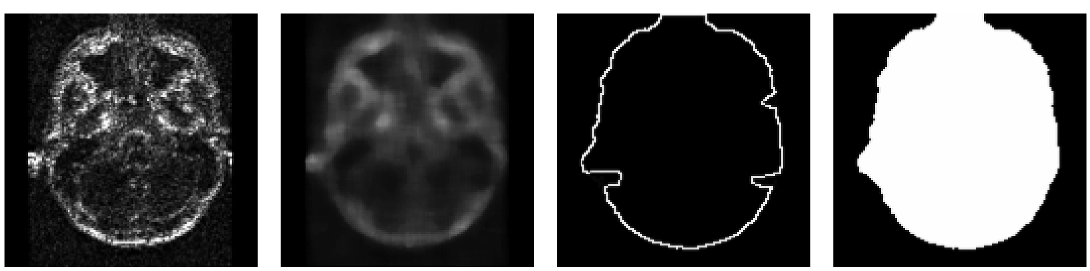


Fig. 4.2. Background mask identification from the wavelet transform

The proposed methods generate similar results, even though the first requires less attention to signal processing, while the latter is more efficient and may be easily embedded into the coder.

The result of the preprocessing stage is shown in Fig. 4.3. Slice 36 of the original and masked *MR_ped_chest* dataset together with the 3D rendering of the mask are shown in the inset. The histograms of the original image (org), object (obj) and background (back) are also represented in Fig.4.4. It can be noticed that details of the chest cavity are preserved, differently from a simple thresholding method, and the object and background histograms overlap in the low-levels region. Fig. 4.5 compares coding performances (discussed in Section 4.5) for a portion of the original and background-suppressed volumetric data, at different lossy bit-rates and in the lossless case. It can be observed that preprocessing gain is small at very low bit-rates, but grows rapidly with increasing bit-rate. Moreover, the lossless result obtained with the preprocessed data is about 2.4 times smaller than that obtained with the original data.

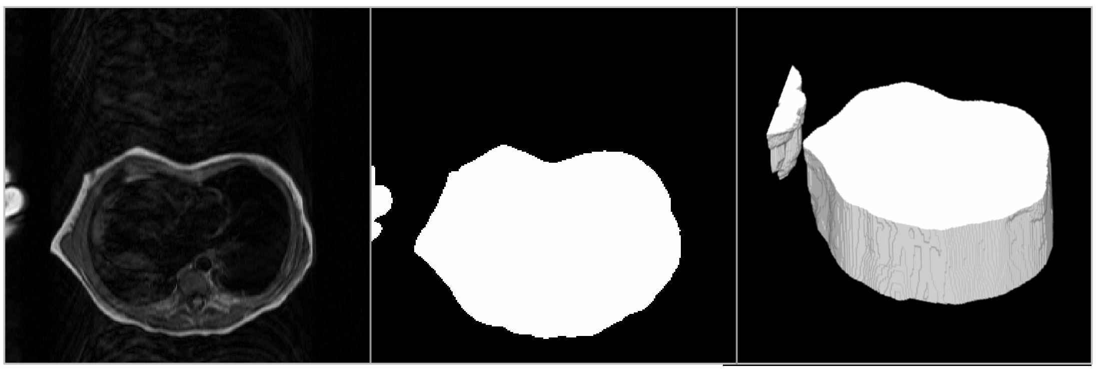


Fig. 4.3. Slice 36 of *MR_ped_chest*; from left: original, mask and 3D mask

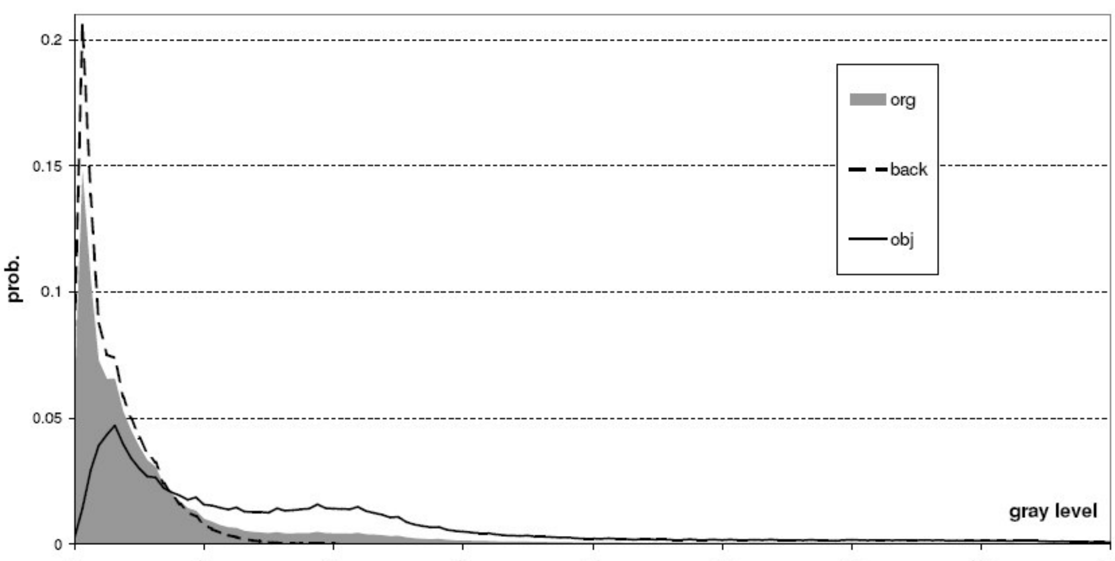


Fig. 4.4. Histogram for *MR_ped_chest*;.

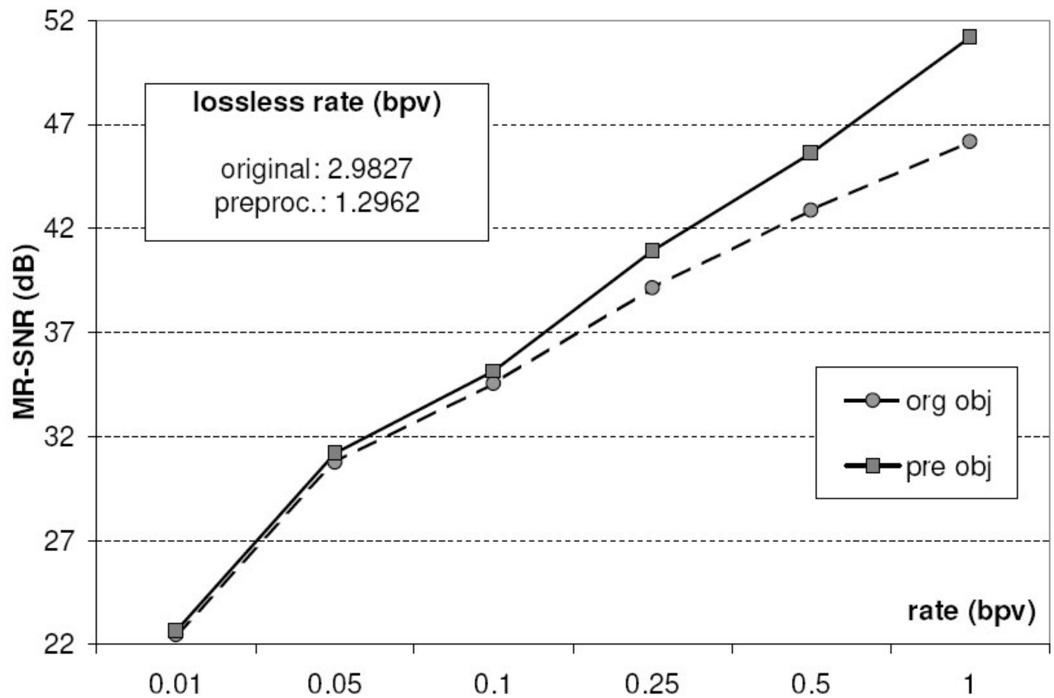


Fig.4. 5. Lossy and lossless coding performance for a portion (32 slices) of the original and preprocessed *MR_ped_chest* dataset without ROI application.

4.3. Coding

The coding core for the proposed approach is based on the 3D packet wavelet SPIHT presented in [Gin+04]. Its object-based generalization requires each significance test to be constrained by the ROI mask. In fact, only the voxels or descendants belonging to the given ROI are to be considered for further processing. This mechanism should result in coding performance increase by limiting the processing to the ROI coefficients only. In order to achieve fully lossy to lossless ROI coding without border artifacts, the ROI mask must be first scaled according to the chosen 3D wavelet decomposition. After scaling, the wavelet domain ROI must be extended according to the adopted wavelet kernels. In Fig. 4.6 the initial background and ROI mask are shown for the *MR_Head* dataset on the left. The scaled mask is shown on the center, with filter extension in gray. Finally, the mask is superimposed to the wavelet transform.

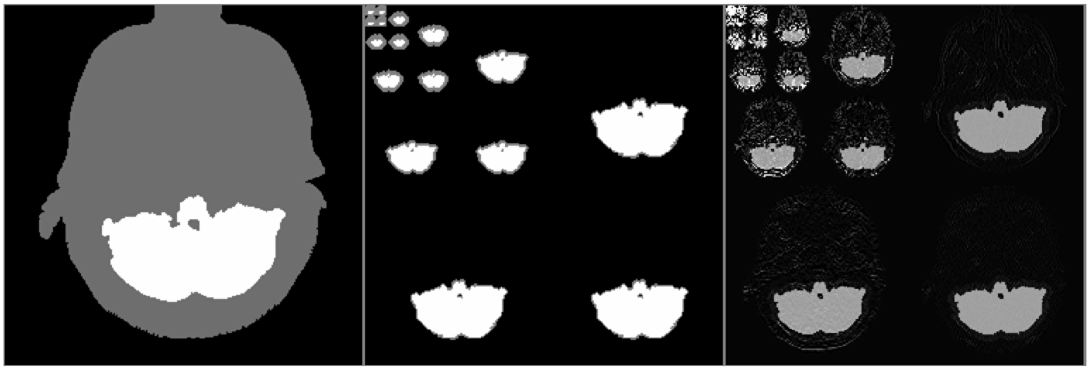


Fig. 4.6. Slice32 of *MR_Head*; from left: background mask with ROI, scaled ROI with filter extension, ROI superimposition to wavelet transform.

4.4. Results

In order to evaluate the performance of the proposed algorithm, three datasets have been considered; their characteristics are reported in Table I together with the pertinent region of interest.

TABLE 4.I. Sample datasets

Name	Size	Bit-depth	ROI
MR_Head	256×256×192	16	brain
MR_Ped_chest	256×256×64	8	heart
MR_Liver	256×256×48	8	liver

Background suppression is applied to each dataset. The object of interest described by the ROI is lossless compressed, while the background is coded at 0.1bpv, 0.5bpv and lossless. In case of lossless coding, the resulting bitrate is reported in bpv. In case of lossy coding, the maximum range SNR (*MR-SNR*) is used as quality metric, defined as:

$$MR - SNR_{dB} = 20\log_{10}\frac{\max_{x\in X}\{x\} - \min_{x\in X}\{x\}}{RMSE\left(X,\hat{X}\right)} \tag{4.1}$$

where X and $\hat{X}$ are the original input without background suppression and co-decoded signals respectively. It must be noticed that *MR-SNR* results are generally lower than the corresponding PSNR, since the input range rarely reaches the maximum extension.

Table 4.II shows the results for each dataset. Column ‘Masked back’ indicates the *MR-SNR* for the masked (suppressed) background. Since such data is intentionally lost, this information is not useful in itself. However, it provides an estimation of the background energy. In fact, a lower *MR-SNR* value indicates higher background energy if compared to

its approximation, that is the null signal given by background suppression. Lossy *MR-SNR* results are relative to the considered sub-volumes, except for the columns titled 'Back'. In this case, the *MR-SNR* is reported for the region only, followed by global result obtained with background suppression, lossless ROI coding and object compression at the given rate.

The last columns represent the bitrate obtained by coding the background-suppressed data as a whole, or lossless coding the ROI and object separately. The percentage increase of the proposed method over the whole data is shown in the last column. The increase is mainly due to ROI extension for precise borders reconstruction.

TABLE 4.II. Coding results

		MR_Head		*MR_ped_chest*		*MR_Liver*	
		Rate (bpv)	*MR-SNR* (dB) region/global	Rate (bpv)	*MR-SNR* (dB) region/global	Rate (bpv)	*MR-SNR* (dB) region/global
Masked back.		-	36.62	-	27.07	-	23.36
ROI (lossless)		0.8825	∞	0.4422	∞	0.6248	∞
Object	0.1bpv	0.1	30.01 / 33.44	0.1	34.92 / 31.05	0.1	29.13 / 26.98
	0.5bpv	0.5	38.18 / 40.37	0.5	46.13 / 31.41	0.5	37.61 / 28.02
	lossless	3.1583	∞ / 45.55	0.9854	∞ / 31.44	2.1499	∞ / 28.22
Whole (bpv)		3.7543		1.2962		2.6443	
Obj+back (bpv)		4.0408		1.4276		2.7747	
% diff		+7.63		+10.14		+4.93	

It may be observed that while for *MR_Head* the global *MR-SNR* increases regularly with the bitrate, for the other two datasets this behavior is fairly less appreciable. This is attributable to the different influence of background suppression. In fact, while for *MR_Head* the background suppression results in a *MR-SNR* of 36.62dB, in the other cases its effect is rather larger, showing a *MR-SNR* of 27 and 23dB. Consequently, in the last two cases the global *MR-SNR* is mainly affected by the background suppression, and shows little variation for increasing object bit-rates. Finally, in Fig.4.7 a visual comparison is shown for *MR_Head* and *MR_Liver*. On the left is the original slice, while the other pictures represent the co-decoded image with background suppression, lossless ROI and object compressed at 0.05, 0.1 and 0.5bpv respectively.

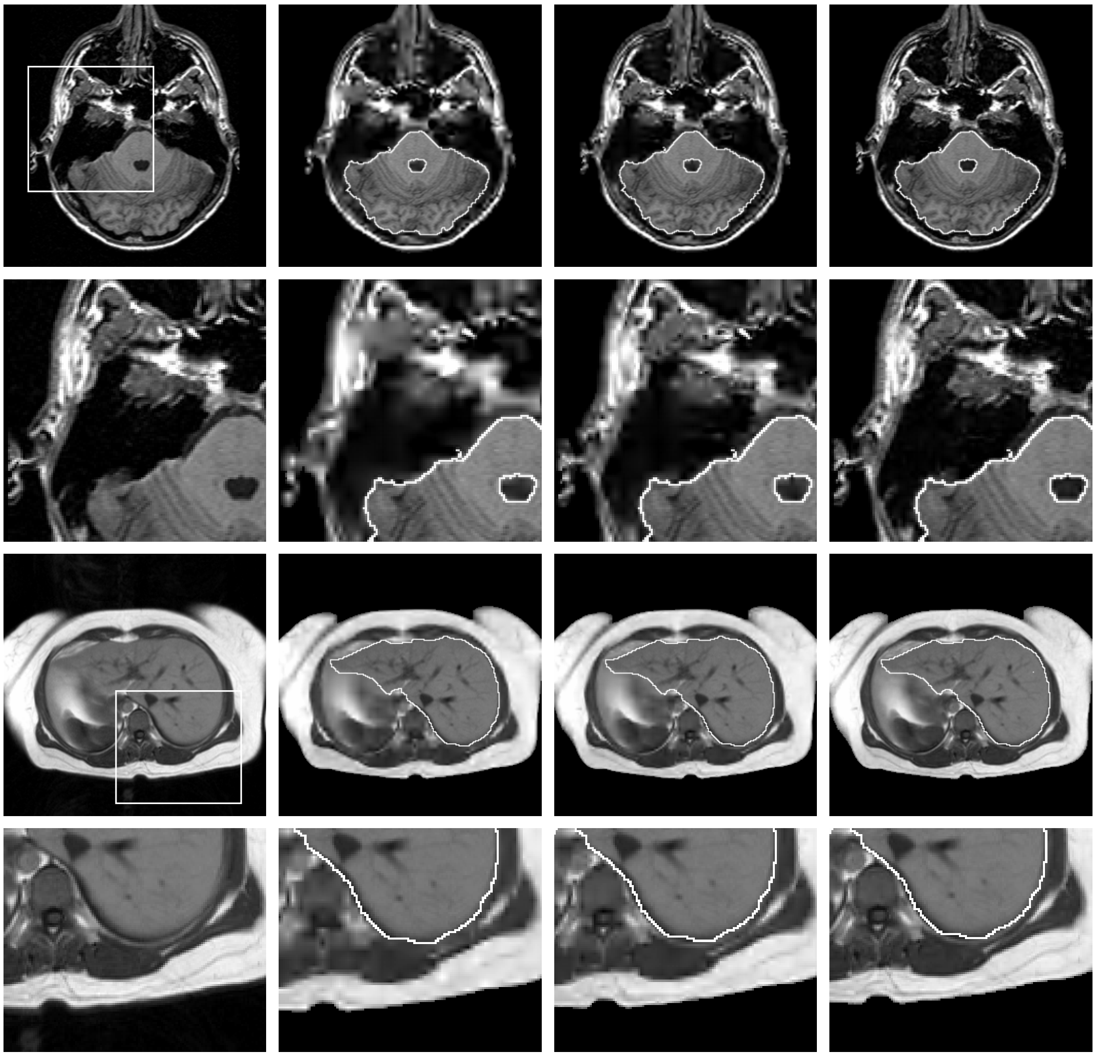


Fig.4.7. Top: slice 52 of *MR_Head*; bottom: slice 47 of *MR_Liver*; from left: original, lossless ROI, object coded at 0.05, 0.1 and 0.5 bpv respectively.

4.5 Conclusions

A new volumetric coding scheme has been introduced. Object-based compression is accomplished through a preprocessing stage for background identification and suppression and ROI-based coding for object of interest prioritization. The proposed method has been extensively tested on three medical datasets. Good results are achieved in terms of rate-distortion and scalability.

Chapter 5

Concealment of JPEG2000 Compressed Images

5.1 Introduction

JPEG2000 [TAUB] is the most recent lossy and lossless image compression technology developed by ISO. With respect to previous standards, and to other compression schemes, JPEG2000 provides a number of remarkable features, which make it optimal for a variety of applications, such as digital photography, home entertainment, Internet, remote sensing and the emerging multimedia mobile communications.

A key point in the evaluation of these applications is the visual quality perceived by end-users. As conventional communication protocols do not usually guarantee error-free transmissions, bit-errors and data losses in received compressed streams may heavily affect the signal quality. To overcome these problems, several solutions have been proposed. Forward error correction schemes (FEC) can be used to protect the most significant parts in the bit-stream. Resilient coding allows for stream decoding in the presence of errors. As an example, layered coding ensures a minimum guaranteed level of quality even in critical situations, by partitioning the picture information into several streams with different transmission protection levels. In [Thi+03], a strategy for allocating source elements into clusters of JPEG2000 packets and finding their optimal code-rates is proposed, which takes into consideration the properties of scalable sources with tree-structured dependency. The source elements are allocated to clusters of packets according to their dependency structure, subject to constraints on packet size and channel codeword length. A similar procedure is presented in [Ban+02], where the use of turbo codes is proposed to implement an unequal error protection scheme.

Differently from error resilience coding, the error concealment approach does not require any additional network service to enforce transmission correctness, but introduces post-processing modules at the decoder side to mask the visual effects of transmission errors. Most of the works proposed in the past have been developed for block-coded images and video sequences. In fact, the former JPEG and the MPEG-1, -2, -4 standards are based on partitioning the image into blocks, which are coded by means of the DCT transform. Errors in the stream result in the loss of all the information related to the damaged block and the next ones up to the next resynchronization marker. Accordingly, the concealment algorithms have been developed assuming that the spatial information related to the affected blocks is completely lost and the surrounding area is entirely available. Figs. 5.1.a and 5.1.d show typical visual effects of bit-errors in JPEG-coded images for the well-known images Bank and Einstein. In [Wan+93], Wang *et al.* propose a reconstruction approach based on maximizing the smoothness between the damaged area and its surroundings. In [Alk+00], the lost coefficients are linearly interpolated from a reduced set of border pixels. Zhu *et al.* propose an algorithm that exploits the smoothness property based on the second-order derivatives [Zhu+98]. In [Suh+97], Suh and Ho propose an algorithm called directional interpolation, which restores edges that are continuous with those present in the neighboring blocks.

Due to the use of a different coding scheme, errors in the JPEG2000 stream produce fairly different visual effects in the decoded image compared to JPEG. As a consequence, concealment algorithms that are applicable to block-coded images are not applicable when using the new standard. To prove this assertion, we briefly outline the new coding scheme. The image samples are firstly transformed into spatial frequency subbands with the aid of the Discrete Wavelet Transform (DWT), usually with five decomposition levels. Each subband is partitioned into small blocks, known as code-blocks, with a typical dimension of 32×32 or 64×64 and each code-block is independently coded. Initial quantization and bit-plane coding are performed on these code-blocks. The bit-plane coder makes three passes over each magnitude bit-plane of the quantized subband sample indices. Errors in the JPEG2000 stream cause the loss of wavelet coefficient bit-planes in a code-block, which results in a spatial damage spreading over a certain area not uniformly. Indeed, the damage extent depends on the local image frequencies. Assuming the use of some default coding parameters (32×32 code-blocks and 5 decomposition levels), damage in a code-block belonging to the lowest decomposition level spreads over an area of 1024×1024 pixels. This area is not completely lost but is only partially damaged at the frequency band corresponding to the affected decomposition level. This phenomenon is illustrated in Figs. 5.1.b and 5.1.e. It follows that the error concealment for block-coded image cannot be applied.

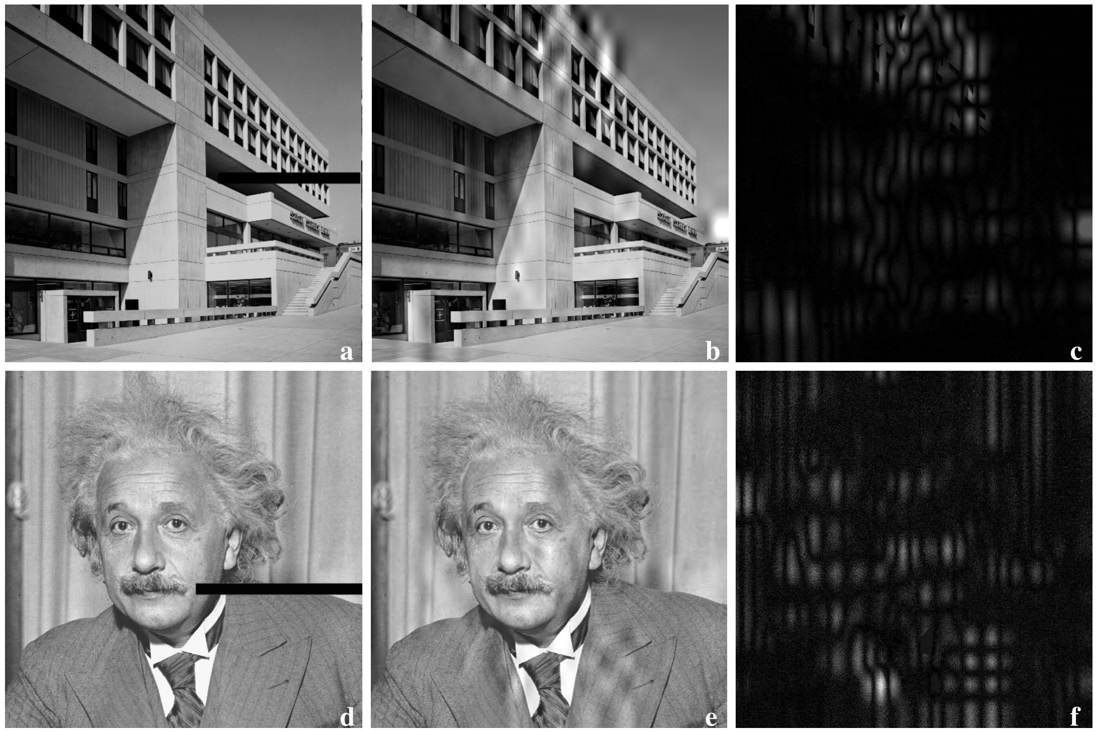


Fig. 5.1. Effects of a bit-error in JPEG (*a*, *d*) and JPEG2000 (error in the first decomposition level, LH subband, 26^{th} bit-plane) (*b*, *e*) coded images for the *Bank* (*top*) and *Einstein* (*bottom*) pictures; in *c* and *f*, the absolute differences between the pictures in *b* and *e* and the original ones are shown.

Damages in the lowest level subband, although more visible than those in other subbands, can be more easily concealed, since a significant amount of spatial correlation is still available to help predicting the lost information. Even if less annoying, damages in high-frequency code-blocks at lower decomposition levels may introduce annoying artifacts. These artifacts cannot be neglected if a satisfactory quality is required in the visual communication. In this case, the concealment operation is complicated by the fact that the smoothness is weak. Only few solutions have been proposed to address this problem, as discussed in Section 5.2.1. An alternative approach is proposed in this research by focusing on the real effects of high-frequency errors and exploiting the theory of projections onto convex sets (POCS) in a weak sense. The first step of this approach consists in the detection of the error through several JPEG2000 features, which allow establishing the bit-plane, code-block number, subband and decomposition level where the error is located. Once the damaged subband has been identified, the error concealment method proceeds with the iterative application of low-pass filtering in the spatial domain and restoration of the uncorrupted wavelet coefficients in the transform domain. The first allows for the removal of the effects caused by errors at high-frequencies in the spatial domain; the second reduces the side-effects of the low-pass filtering by restoring the unaffected wavelet coefficients.

This chapter is organized as follows. In Section 5.2, we provide a brief background about past works, an analysis of bit-error effects on JPEG2000 encoded images, and an overview of the POCS theory. In Section 5.3, the proposed method is described and effects of the applied spatial filtering are discussed so as to improve filtering parameters setting. Results are discussed in Section 5.4. Finally, in Section 5.5 the conclusions are drawn.

5.2. Background

5.2.1. Related Works

In the literature, a few solutions have been proposed to mask errors in JPEG2000 coded images. In [Ran+02], the lost blocks of coefficients are concealed by an interpolation along the edge direction driven by a minimization of the square error at block borders. An alternative approach is proposed in [Atz+03], where a set of wavelet patches is generated from uncorrupted data within the damaged subband; the patch that minimizes a matching measure is used to conceal the error by replacing the corrupted block. The generation of such a set is based on the characteristics of the damaged area, which are predicted by analyzing the properties of the other subbands. Both strategies cannot be applied with the common settings of JPEG2000 coding parameters: five decomposition levels, a unique tile (more tiles increase the overhead and may generate tiling effects) and code-blocks with dimensions equal to 32×32 or 64×64. In particular, these two algorithms are applicable only in case of very small code-blocks or tiles.

In [Hem+97], Hemami and Gray propose a subband-coded image reconstruction for both low-frequency (LL subband) and high-frequency errors (other subbands). LL coefficients are reconstructed by a cubic surface fitting to known coefficients, which is driven by the edge information in the damaged area. Isolated losses or very small blocks of missing coefficients can be concealed. For the concealment of high-frequency errors, that is the framework of our work, Hemami and Gray propose a linear interpolation in the low-frequency direction in the HL or LH subbands to reconstruct only isolated missing coefficients. The considered framework is quite different from the one to be addressed in case of JPEG2000 errors. In fact, even a single bit-error in the compressed stream heavily affects the decoding of the subsequent wavelet coefficients belonging to the damaged block. The dimension of the damaged area depends on the used block size (typical values: 32×32 or 64×64), but it is always quite larger than a block of few coefficients. The extent of the damage depends on the affected bit-plane. In another work [Hem00], Hemami proposes the use of RVLC (Resynchronizing Variable Length Codes) to limit entropy coding error propagations. This error-resilient approach is applied in conjunction with a concealment method, which mostly relies on the algorithm in [Hem+97]. However, the use of RVLC is not compliant with the standard: its adoption was proposed in the past within the standardization committee activities [Yan+99] but this feature has never been included. Also in [Kur+03], a new technique that combines error protection and concealment is proposed. In particular, a method that uses the layer structure and a watermarking technique is experimented. The most significant layer is hidden in the lowest layer of the JPEG2000 bitstream and the hidden data are used for error concealment.

Although developed for block-coded images, the works in [Sun+93] and [Yu+98] are worth of mention at this point, since both make use of the POCS theory as we do in this research. As discussed in Section 5.1, errors in block-coded images spread over all the spatial frequencies related to a certain number of image blocks. Accordingly, the information related to the affected spatial blocks is completely lost while that related to the surrounding area is completely available. Sun and Kwok in [Sun+93] consider projections related to the smoothness of edge continuity and theYu+98] relies on a 9x9 DCT with one pixel overlap between adjacent blocks, which is used to impose a continuity constraint during the concealment. Other two convex sets are defined. They are based on constraints on the available coefficients (those belonging to unaffected blocks) and constraints of pixels having equal intensities.

5.2.2. Effects of Bit-errors in the JPEG2000 Stream

The intent of this section is to describe the resilience features of the JPEG2000 standard and to evaluate the characteristics of spatial artifacts caused by bit-errors. The extents of the artifacts are also analyzed to understand which error type is worth concealing.

Regarding the effects of bit-errors in the JPEG2000 stream, [Bil+03] and [Moc+00] provide valuable case studies. Errors in the packet body result in synchronization loss between the encoder and decoder, which prevents the decoder from correctly reconstructing the bit-plane information conveyed in the damaged packet. The error does not propagate from one block to the others sent in the same packet. This arises from the use of the information enclosed in the packet header. It is worth noting that the damage doesn't affect only the wavelet coefficients that had bit-plane information coded in the affected packet. In fact, due to entropy coding, the subsequent bit-planes for all the coefficients in that block are not correctly decodable too. The decoder is able to detect errors by using byte stuffing mechanisms and the SEGMARK symbol string. The synchronization at packet level is assured by using the SOP marker at the beginning of each packet. Once an error is detected, the decoder can either try to continue the decoding of the affected code-block stream or set the relevant bit-plane to zero.

In [Bil+03], the authors propose also a technique for smart parsing the coding passes sent after the error (for the affected code-block). It relies on external information (for example, provided by the transport layer), which reveals the exact position of the error. If this information is available, the decompression of current and future coding passes can be enhanced.

Errors in the packet header are quite more dangerous. To avoid such situation, the standard provides the PPM marker segment, which can be used to relocate the packet headers from their respective pack-streams to the main header. This option should be always used in case of transmission over error-prone channels and is considered in our study. Alternative solutions to protect header against errors have been proposed. In [Nat+02], Natu and Taubman consider the benefits of protecting packet headers with stronger codes than the corresponding bodies. To the same purpose of extending the error resilience to headers and avoiding the decoding crashes due to error presence, Nicholson *et al.* propose an Error Protection Block (EPB) marker segment containing information about the error protection parameters and data used to protect the headers [Nic+03]. In both cases, to obtain a code-stream compliant with the JPEG2000 Part I [J2K], it is necessary to place the redundant information in such a way that any standard JPEG2000 Part I decoder won't try to interpret it.

In [Lia+00] and [Pou+01], the results of testing error resilience tools are shown. The tests are based on VM 6.0 [VM6] and VM8.6 [VM8]; the simulations have been conducted with noisy channels at different BER (Bit Error Rate). The test results prove that the error resilience tools can provide significant performance improvements with only a small reduction in the data compression performance. However, in these works the correlation between the damaged information type and image quality effects have not been analyzed. Indeed, this is quite important for the development of an error concealment algorithm; therefore, we have performed additional experiments. These have been carried out on standard images (512×512 pixels, 8 bpp grayscale, some of these images are presented in Fig. 5.11), encoded by setting five wavelet decomposition levels, using code-block of size 32×32, and inserting the SOP and SEGMARK markers in the code-stream. To simulate a noisy channel, random bit-errors with a BER in the range 10-3÷10-4 have been introduced using the simulator provided in [JPWL].

As expected, due to the quite different importance of the stream segments, it has been observed that a single bit-error can introduce either almost invisible artifacts or severe damages in the image. In fact, the damage extension is closely related to the affected decomposition level and bit-plane. For example, with the common JPEG2000 settings, a single error at the first decomposition level may result in the corruption of the entire related subbands. Such deterioration severely affects the reconstruction process during the inverse wavelet transform operation. Table 5.I shows the results in terms of the PSNR, in dB, averaged over several test images encoded at bit rates from 1bpp to 0.1bpp and transmitted at the mentioned BER range. The PSNR has been calculated comparing corrupted images with the JPEG2000 co-decoded images. The results refer to errors introduced in the LH, HL, and HH subbands, each consisting of a collection of floating-point coefficients, at different wavelet decomposition levels. The image degradations have been analyzed varying the affected bit-plane and the decomposition level. Note that the higher the bit-plane layer, the most significant the bits are. Since the information in packets belonging to bit-plane $i+1$ is required to decode the bit-plane i and so forth, the data sensitivity to corruption clearly increases as we move from lower to higher quality bit-planes. This suggests that errors in higher bit-planes have a more significant impact on image quality than errors in lower bit-planes. Moreover, the effects of errors are more severe if the corrupted subband is at lower wavelet decomposition levels, that is, if this subband is closer to the LL subband.

TABLE 5.I. PSNR results, averaged over several images, varying the bit-plane and the subband affected by the error.

Decomposition level	*Subband*	**Bit-plane 27**	**Bit-plane 26**	**Bit-plane 25**	**Bit-plane 24**	**Bit-plane 23**	**Bit-plane 22**
First	***LH***	21.6	26.1	32.4	34.0	39.1	42.8
	HL	20.8	25.1	28.3	32.8	39.2	44.2
	HH	24.5	26.7	30.3	35.2	39.2	46.5
Second	***LH***	26.4	28.3	31.4	34.1	38.8	42.6
	HL	25.2	28.9	29.5	33.9	39.6	45.3
	HH	27.7	25.9	32.6	36.1	40.7	44.1
Third	***LH***	32.5	33.0	40.3	42.4	46.0	51.1
	HL	33.1	35.7	37.9	41.5	45.3	52.6
	HH	36.0	38.0	40.6	44.3	47.6	51.5

By observing the visual effects of transmission errors, it comes to light that corrupted images with PSNR over 30÷32dB present only slight artifacts in most of cases. Based on this, we have decided not to consider the masking of these errors. Then, we focused on high bit-planes errors (first three bit-planes) located at the first and second decomposition levels. Figs. 5.1.b and 5.1.e show common spatial effects of errors at the first decomposition level, LH subband, 26th bit-plane for the images; an annoying "wave" artifact is presented over a significant area of the corrupted images.

Table 5.II shows the magnitude difference between correct and corrupted wavelet coefficients of the affected subband in float precision for the error in Fig. 5.1.b. It can be noticed that the number of corrupted coefficients is relatively small in respect to the subband size and their distribution appears random in position, magnitude and sign.

TABLE 5.II. Error in the wavelet domain for the *Bank* image corrupted at the LH subband, 26^{th} bit-plane, first decomposition level (resulting PSNR=23.22dB): this table shows the absolute difference between the corrupted coefficients and the original ones (float representation) for only the affected subband.

2.824936	0	2.139237	0	0	5.078156	0	3.853409	0	0	0	0	0	0	0	0
0	4.621247	5.992646	0	3.16774	0	0	0	0	2.824936	0	0	0	0	0	0
0	0	0	-2.074234	-5.31E+19	0	-8.3E+19	-8.3E+19	0	2.824906	-2.074265	0	0	0	0	0
0	2.024887	0	0	6.678254	0	0	3.510575	0	6.678315	0	0	0	0	0	0
0	0	-2.074234	2.367722	0	0	5.22E+19	1.4E+20	0	-4.491303	0	0	0	0	0	0
-3.902786	0	0	0	4.621247	0	2.824936	0	-2.759903	0	0	0	4.392548	0	0	0
-2.417069	0	0	0	0	0	0	0	0	-4.491242	1.28E+20	3.853409	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-4.491242	0	0	0	0	0
5.22E+19	0	0	0	0	0	0	0	0	0	0	0	-5.94E+19	0	0	0
0	0	0	0	6.7E+19	-5.176972	0	0	0	2.367691	3.16774	9.81E+19	0	0	-1.25E+20	0
2.824906	1.11E+20	-5.405671	5.306916	0	0	0	0	0	0	-3.102738	0	-7.462677	1.79E+10	-3.102738	0
0	0	-3.445603	0	0	0	0	0	0	0	0	0	-2.759934	2.139267	0	0
-5.94E+19	0	2.71E+34	-6.3E+29	6.7E+19	0	0	0	0	0	-1.43E+10	0	0	0	0	0
-5.862641	5.078217	0	-5.176911	1.46E+10	4.621247	6.7E+19	5.992646	-6.54831	0	-2.074265	0	0	0	6.678254	0
-2.11E+29	5.992585	0	-5.405671	0	0	2.139237	0	0	2.482102	0	-2.417099	-6.54831	0	0	0
-2.417099	0	-2.874283	-1.43E+10	0	5.078217	0	0	0	-6.548249	2.367691	0	0	0	0	0

5.2.3. Projections onto Convex Sets

The theory of projections onto convex sets has been used extensively in the field of error concealment and image restoration. Since its introduction by Youla [You78], the method has been extended for a number of applications where a priori information can be used to constrain the size of feasible solutions.

We briefly summarize the theory in the following. Given m closed convex sets, $C_i, i = 1,2,\cdots,m$, in a Hilbert space, and $C_0 \equiv \bigcap_{i=1}^{m} C_i$ nonempty, we call P_i the projection operator, such that:

$$\|\boldsymbol{f} - P_i\boldsymbol{f}\| = \min_{\boldsymbol{g}\in C_i}\|\boldsymbol{f} - \boldsymbol{g}\| \tag{5.1}$$

where $P_i\boldsymbol{f}$ is called the projection of f onto C_i.

Projecting the distorted signal onto C_i results in a new signal for which the distance to the original is reduced or at least not increased. The outcome of the previous operation is projected onto the next set C_{i+1} and so on until the last set is considered. Although the distance with the original signal gradually decreases, even the last projection does not guarantee that the result lies in C_0. However, the iteration

$$\boldsymbol{f}_{k+1} = P_m P_{m-1} \cdots P_1 \boldsymbol{f}_k, \quad k = 0, 1, 2, \cdots \tag{5.2}$$

converges to a point of C_0 for an arbitrary initial point $\boldsymbol{f}_0$.

In general, the design of a POCS-based image recovery algorithm consists in two steps:

1) the definition of the convex constraint sets to be used;
2) the derivation of the projections onto the previously defined sets.

Many sets of constraints have been considered to solve each case. They are based on image properties such as smoothness, when an image/region is required to have slowly varying values; edge continuity, when objects are required to have continuous edges; consistency with known values, when each correctly received sample must not be altered and may control the restoration process in its surroundings.

Through the iteration of such projection operators, it is possible to search for a fixed point that is an acceptable approximation of the original signal. The process is represented in Fig. 5.2.a, where we assume two constrained sets and, consequently, two projection operations have been defined.

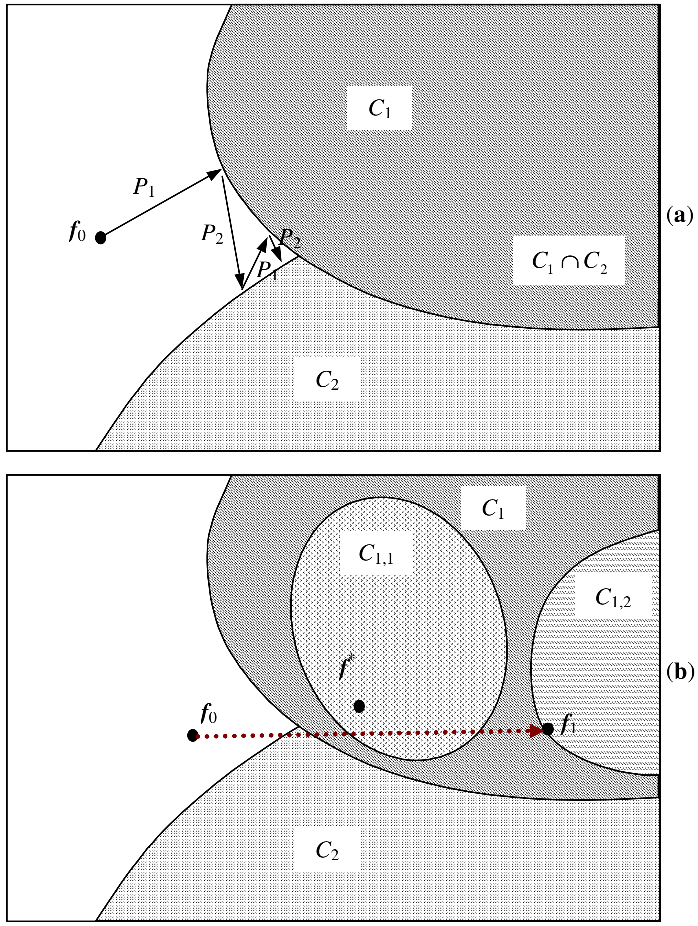


Fig. 5.2. (a) Projections onto convex sets. (b) Possible configuration of the considered convex sets.

5.3. Proposed Method

In Section 5.2.2, we concluded that the error concealment at high-frequency subbands should only address damages in high bit-planes at the first and second decomposition levels, when default coding settings are used. With common image dimensions (512×512÷1024×1024), each subband at these resolution levels is made of only one code-block. Then, when an error is encountered, the information of the affected subband can be severely, if not entirely, altered.

Since the lost coefficients are representative of spatial details at low resolutions (the image signal is first processed with the LP analysis/synthesis filter repeatedly until the involved subband is reached), their loss results in error patterns similar to the effects of inhomogeneous lighting with an undulating behavior that is connected with the use of the wavelet transform. Such annoying “wave” effect can be seen clearly in Fig. 5.1, where the corrupted images Bank and Einstein are shown in b and e and the absolute differences with the original images are displayed in c and f.

To correct these error patterns, a solution is to apply an appropriate low-pass filtering in the spatial domain. The filtering should be able to gradually remove the disturbing "wave" artifact, while preserving image characteristics, such as edge definition and intensity value. In fact, it is difficult to find the optimal filter for this problem, but the median filter is one of the most appropriate [PETR, JAIN]. Among its properties, it has low-pass characteristics and is very efficient in the removal of noise that has a long-tailed distribution (*e.g.* Laplacian distribution). Not only does it smooth noise in homogeneous image regions, but it also produces regions of constant or nearly constant intensity while preserving edge sharpness.

As a drawback, while the filtering operation is able to partially conceal the effect of wavelet coefficient alteration, it inevitably introduces a certain amount of errors since it is implemented uniformly across the image. As a consequence, the spatial filtering itself is unable to conceal the errors properly, but can be used to obtain an approximation of the original image. To partially solve these drawbacks, the filter can be applied iteratively to improve its smoothing effects; additionally, the side-effect damages on the correctly received subbands can be removed by reintroducing the correct coefficients.

These considerations represent the basis for the proposed concealment algorithm, described in the following sub-section. After the analysis of the results, an extension of the algorithm is proposed to improve its performance; the analysis and the extension are reported in sub-sections 5.3.2 and 5.3.3, respectively.

5.3.1. Basic Algorithm

Based on the previous analysis, we developed an iterative algorithm for error concealment that relies on the theory of POCS in a weak sense.

Although the following description will refer to wavelet transform coefficients, it needs to be said that the algorithm itself works directly with the JPEG2000 bitstream. Several routines have been implemented to extract and insert the coefficients from and into the stream.

The first step of the algorithm consists in zeroing the corrupted coefficients. The stream resulting from this operation constitutes the input for the following processing and represents the starting point for our iterative method. This choice is justified by the consideration that, given the unpredictability of the error distribution, the use of the unmodified corrupted stream as starting point may result in the instability of the algorithm. On the other hand, zeroing the corrupted coefficients has experimentally proven to be a proper seed.

The second step is to formulate the desired properties in terms of convex constraints. To characterize such properties, the following constraints and projections are considered:

1) The class C_1 of images with smooth regions delimited by well-defined contours. Given the generality of such definition, this class can be seen as container for several sub-classes, each with different projection operators. Among these, we have chosen the median filtering, whose convergence to C_1 can't be proven theoretically. However, this is an expected behavior that will be discussed in the following. The projection operator P_1 onto the convex set C_1 is then:

$$P_1 f = \Theta_M(f), \tag{5.3}$$

where $\Theta_M(\mathbf{x})$ denotes the median filtering operator applied to image $\mathbf{x}$ with a square window of size M.

Some problems may arise from the extent of the C_1 set. In particular, the $\Theta_M(\mathbf{x})$ operator can converge to a sub-class in C_1 not containing the uncorrupted image. This problem is illustrated in Fig. 5.2.b where f_0 is the seed, $C_{1,1}$ is the subset containing the correct solution, f^*, and $C_{1,2}$ is another subset containing the convergence point f_1. In this example, the chosen operator leads correctly to the defined set, but the ending point is far from the uncorrupted image.

2) The class of signals that have fixed values for the some wavelet coefficients. This set (C_2) contains all signal vectors f in the n-dimensional real space R^n, with some areas or entire subbands of transform coefficients equal to known values. It can be expressed as:

$$C_2 = \{ f \in R^n : [\mathrm{T}f]_i = z_i, i \in \boldsymbol{I} \}, \tag{5.4}$$

where T is a transform operator, z_i are known constants, and $\boldsymbol{I}$ is a set of wavelet coefficients. The projection operator P_2 onto the convex set C_2 is then:

$$[\mathrm{T}P_2 f]_i = \begin{cases} z_i & i \in \boldsymbol{I} \\ [\mathrm{T}f]_i & otherwise \end{cases}. \tag{5.5}$$

The projection operator P_2 can be used to substitute the corrupted coefficients with those computed through a step of decoding, median filtering and encoding, while preserving the correctly received coefficients.

These two constraints are used in the proposed iterative algorithm, described in Fig. 5.3, where each projection operator is incorporated in a functional macro block. After the zeroing of corrupted coefficients, the wavelet coefficients are inversely transformed. The resulting image is then processed with the median filter and wavelet-transformed. Following, the new coefficients at the corrupted subband are replaced in the original bitstream. The process is iterated so that the signal is forced to satisfy the two convex constraints described above:

$$f_{i+1} = P_1 P_2 f_i \, , \tag{5.6}$$

where P_1 and P_2 represent the projection operators onto sets C_1 and C_2, respectively. Although the scheme presented in Fig. 5.3 is the correct and most efficient representation of the proposed method, JPEG2000 co-decoding has been operatively used in place of wavelet direct and inverse transform. In fact, since the experimental research is mainly aimed at evaluating the method feasibility and performance at this stage, computational complexity considerations have been sacrificed for coding and implementation simplicity.

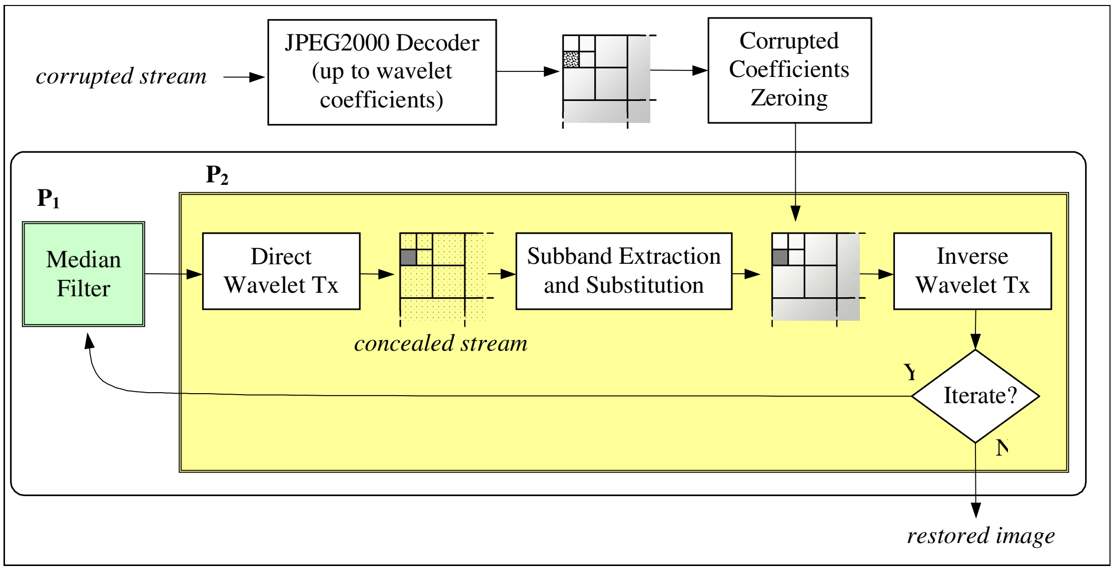


Fig. 5.3. Scheme of the proposed error concealment algorithm.

5.3.2. Discussion

To evaluate the performance of this algorithm, we have selected some representative samples of corrupted streams among those analyzed in Section 5.2.2. In particular, we have selected one sample for each test image shown in Fig. 5.11, characterized by errors at the first decomposition level and bit-plane levels in the range 27÷25. The median filter combined with iterative transform and subband substitution have then been applied to these corrupted images. This algorithm has experimentally proven to be effective for error concealment of wavelet coded images. This is confirmed by the PSNR quality evaluation shown in Fig. 5.4.a. An average improvement of almost 5dB has been obtained with minimum and maximum values of 1.05 and 8.99dB, respectively.

However, an issue arises from the generality of such method. In particular, the algorithm is strongly dependant on the filter size (*M*) and the number of iterations (*N*); additionally, it shows a different behavior with different image types (highly textured, natural, etc.). This phenomenon is highlighted in Fig. 5.4.b, where the combinations of *M* and *N* for the best results are represented in a scatter plot for each test image. These have been obtained computing the PSNR of the filtered images respect to the original one varying *M* and *N*.

Indeed, the encouraging results in Fig. 5.4.a have been obtained with these best combinations, which are quite different for each image. To reach such results, it is essential to define a procedure for selecting the appropriate *M* and *N* values for each image in a real error concealment framework, where original images are obviously unavailable.

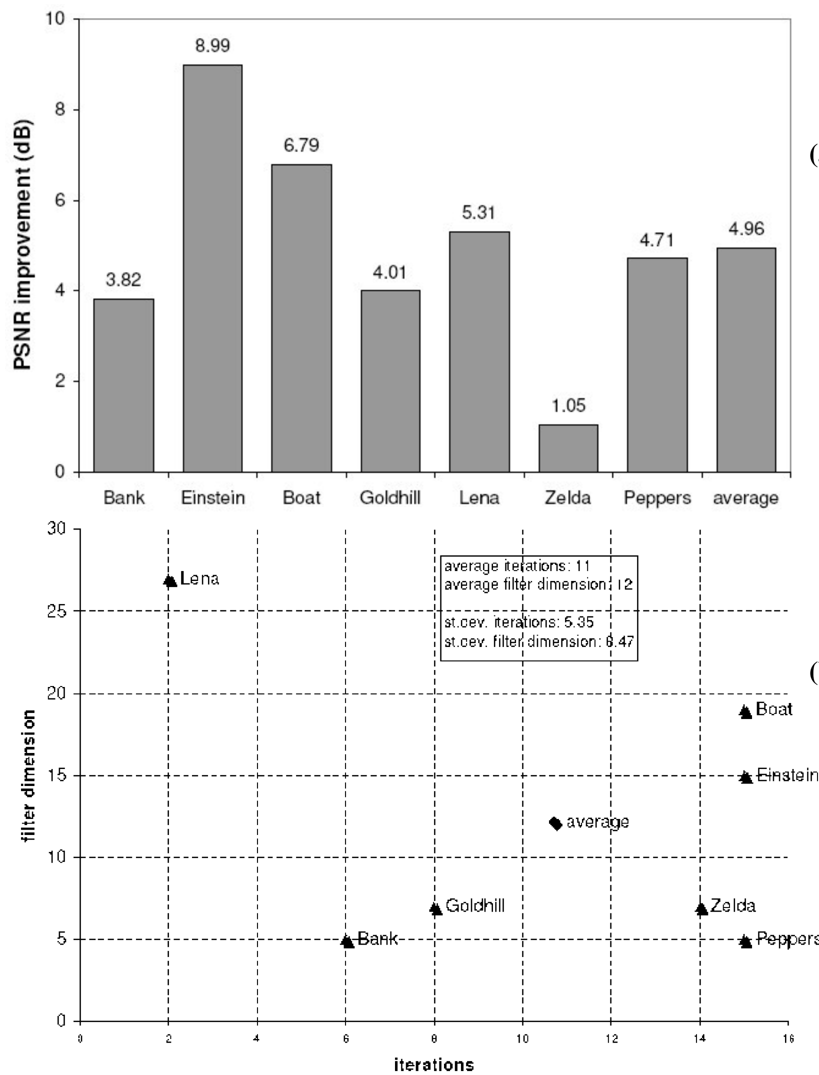


Fig. 5.4. (a) PSNR improvement for each test image over the corresponding corrupted images. (b) Best results distribution scatter plot.

A possible solution relies on the use of a couple of fixed values to be used for every image, which can be experimentally computed by maximizing the filtering results over a considerable heterogeneous set of images off-line. However, the scattering of the points in Fig. 5.4.b is not in favor of this solution. To better analyze its effectiveness, we observed the behavior of the filter varying *M* and *N* for each different images. Due to space problems, in Figs. 5.5 and 5.6 we present the results for only two of our test images (*Bank* and *Einstein*); however, it is enough to carry out our analysis. For each filter size, 30 iterations have been considered and the resulting PSNR computed in comparison with the uncorrupted co-decoded image. The two straight lines marked by “corr” and “zero” represent the PSNR obtained for the corrupted image and that obtained after damaged subband zeroing, respectively. These have been introduced to better evaluate the effects of varying the number of iterations and the window size. In both cases, significant improvements can be obtained with the right combination of *M* and *N*. However, the same couple of values have quite different effects on the two images. Accordingly, the use of fixed values for every image considerably reduces the effectiveness of the algorithm when used in a realistic context. When fixed values are applied to the entire set of test images, we obtained an average improvement of almost 1dB lower than those obtained with the optimal values presented at the beginning of this section.

This analysis calls for a procedure able to adapt *M* and *N* to the characteristics of the image to be filtered. At this point, it is worth considering that desired error masking effects are often reached with big filters, but, on the other hand, their use in areas with fine details brings to edge features deterioration. Such considerations lead to exploit an adaptive method. Two different approaches can be followed: either to adjust the filter size once for each image (adaptivity based on global image considerations) or to set the filter size locally in the image (adaptivity based on local image considerations). Since each image comprises regions with different characteristics, the latter seems the most adequate. Consequently, the adaptivity of the new approach should be based on local considerations on the image energy in terms of edge strength.

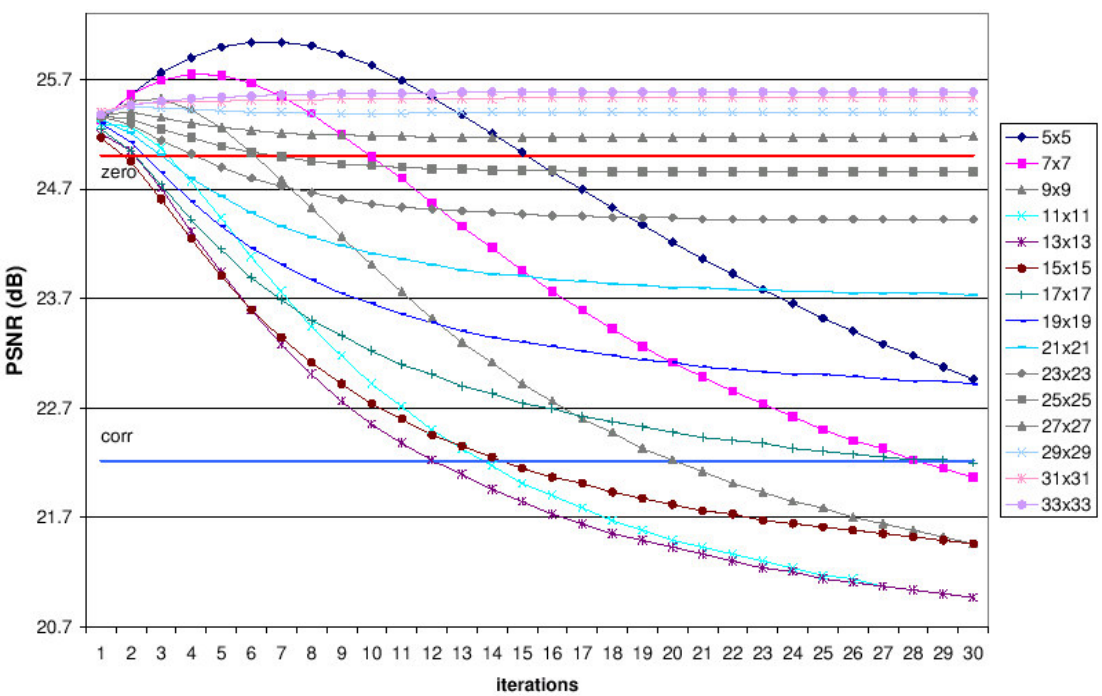


Fig. 5.5. Results of the basic algorithm applied to *Bank*.

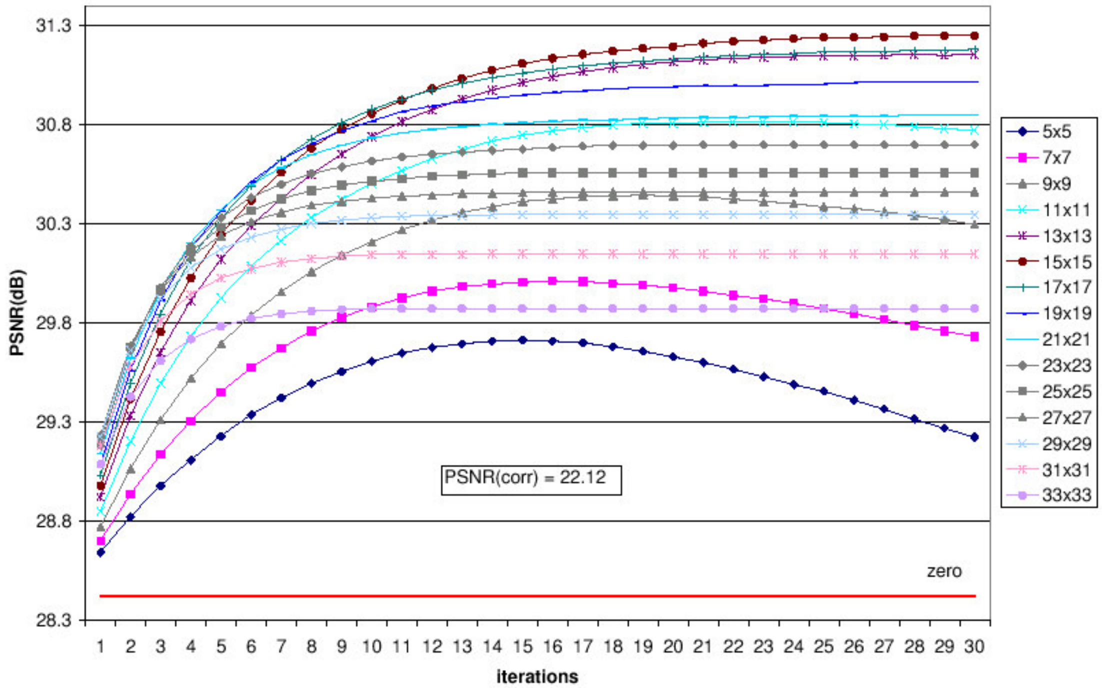


Fig. 5.6. Results of the basic algorithm applied to *Einstein*.

To evaluate the positive and negative effects of the median filter and show that the optimal mask size actually depends on the local image characteristics, two synthetic images have been produced: *circles*, representing LH errors over a smooth region to be concealed, and *regions*, containing objects with well-defined edges to be preserved (Fig. 5.7). The median filter is expected to be beneficial in removing the frequency artifacts from the *circles* image, but it is also expected to corrupt the *regions* image. To evaluate such behavior, the two images have been median-filtered with 3 different mask sizes (M=5, 15, 33). For effect comparison, the PSNR has been computed as shown in Fig. 5.7 by the tag *comparison*. In particular, for *circles* the comparison is meant to show the positive effect of median filtering, and is done by computing the difference between the PSNR of filtered and original images and the PSNR of corrupted and original images. In *regions*, the comparison consists in the PSNR computed between the original and filtered images, thus showing the effect of details degradation. The experiment is reported in Fig. 5.8, where the positive and negative effects are well visible. As expected, while the application of the median filter tends to correct the defects of image *circles*, it severely affects the quality of image *regions*. In fact, since the median filter substitutes the median value in a window of $M \times M$ pixels, it is able to correct (especially if applied iteratively) slowly varying gradients, such as the artifacts caused by wavelet coefficients loss. However, median filtering of edge regions, especially with big masks, results in detail loss and contour deterioration. Such behavior, although exaggerated in our test, is expected to happen with natural images too and can be of major relevance for our method. To improve the performance of the filtering operation, it is then necessary to adaptively balance the effects of artifacts reduction and edge deterioration. For such reasons, an adaptive filtering is strongly desirable. Through such method, the capability of correction of slowly varying gradients should be preserved, while edge and detail deterioration should be avoided. Even if not directly acting on the number of optimal iterations to be applied to each image, the use of such adaptive approach is also expected to make the algorithm less sensitive to the number of iterations. This behavior is analyzed in the following.

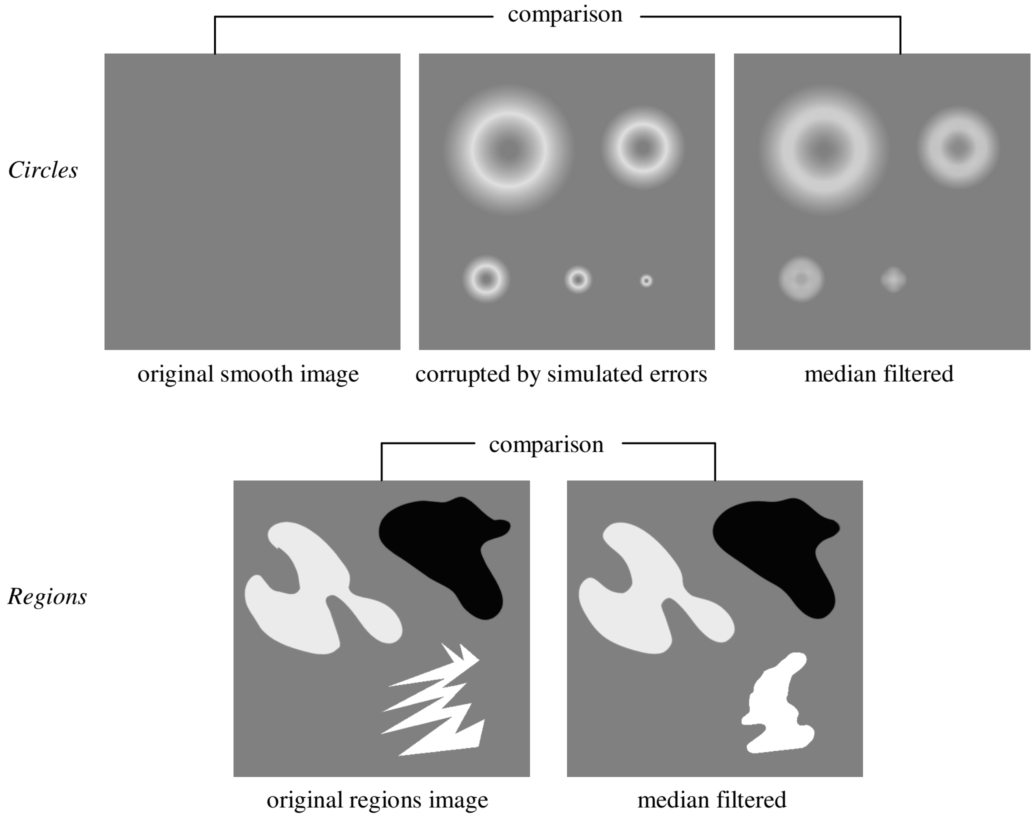


Fig. 5.7. Synthetic test images (*circles* and *regions*) used to analyze the effects of varying the filtering window size: the *comparison* tag indicates which images are used to evaluate the filtering results.

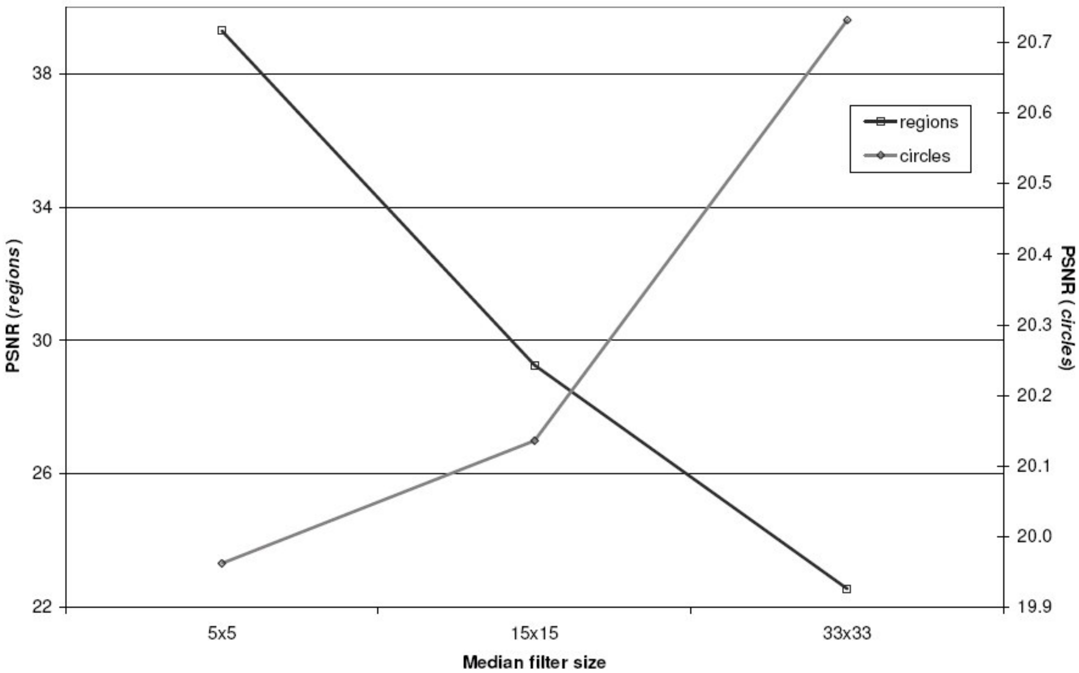


Fig. 5.8. Effects of the median filter on the two synthetic images (*regions* and *circles*).

5.3.3. Adaptive Algorithm

We have considered the limitations of conventional filters and introduced the idea of applying an adaptive filtering to obtain better results and a more regular behavior.

The proposed method is based on a size-adaptive median filter. Our goal is to build an automatic filter, which chooses a small mask in the proximity of edges and details, while opting for a large mask in flat or slowly varying regions. To evaluate the characteristics of each pixel surroundings, we can use the edge map given by the application of the Sobel filter.

The new approach is substantially equivalent to the one described in Fig. 5.3, except for the filtering engine. Instead of applying the simple median filter, we have developed the energy-adaptive median filter sketched in Fig. 5.9.

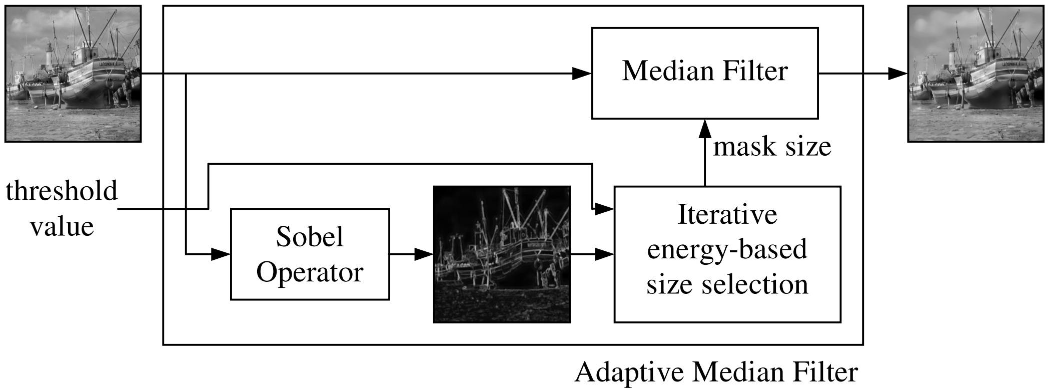


Fig. 5.9. Scheme of the adaptive median filter.

We first consider the Sobel operator, obtained through the convolution of the image $I_{x,y}$ with the masks:

$$S^h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad S^v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{5.7}$$

and the computation of the gradient magnitude at coordinate i, j:

$$s_{i,j} = \sqrt{\left(S^h \otimes I_{x-i,y-j}\right)^2 + \left(S^v \otimes I_{x-i,y-j}\right)^2} \tag{5.8}$$

to obtain the edge map from the original image. Given $s_{i,j}$, pixels resulting from the edge detection operation, we define the edge energy over the mask of size M as:

$$E_{i,j}^M = \frac{\sum_{r=i-M/2}^{i+M/2} \sum_{t=j-M/2}^{j+M/2} s_{r,t}}{M^2}, \quad M \text{ odd}. \tag{5.9}$$

The mask size for median filtering each pixel surrounding is then chosen iteratively. Starting from the biggest size (M×M pixels), we search for the size M as follows:

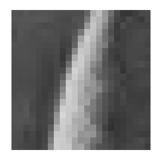
if $E_{i,j}^{M} \le th_e$ then choose size M and exit;

else decrease M of *2* and repeat.

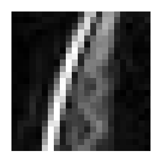
Where the is a fixed threshold level. In case the energy is always higher than the desired threshold, the median filter is applied in a surrounding of 3×3 pixels. The threshold value has been selected experimentally. In particular, a first approximation is to set the threshold value as a fraction of the energy term. Since the Sobel mask is rescaled to integer values in the range 0÷255 and the energy term is normalized in respect to the mask dimension, its value may vary between 0 (smooth region) and 255 (all edges). At first, the threshold has been set at 10% of such range (25.5). In order to evaluate this choice, several experiments have been performed. In fact, a single iteration of the filtering operation has been applied to all images in our experiment set with variable threshold values. The average value of the best threshold estimates has been selected as operative threshold. In particular, such value has been changed from the predicted 25.5 to 22.5.

An example of our method is illustrated in Fig. 5.10. A relatively small region containing edge features from image Lena is displayed in a, while in b its edge map is represented; finally, in c the chosen mask sizes are displayed. Note that the mask size is bigger where the corresponding pixel neighborhood is more homogenous. The efficacy of adaptively selecting the filter mask size is shown in the following section.

25	25	25	23	21	19	17	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	5	9
25	25	25	23	21	17	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9
25	25	23	21	19	17	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	7	9
25	25	23	21	19	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	5	7	9
25	25	21	19	17	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	5	7	9	11
25	23	21	19	17	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	5	7	9	11
25	23	21	17	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	5	7	9	11
25	21	19	17	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	5	7	9	11
23	21	19	17	15	13	11	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	7	9	11
23	21	19	17	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	7	9	13
21	19	17	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	7	11	13
21	19	17	15	13	11	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	7	11	13
21	19	17	15	13	11	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9	11	13
21	19	17	15	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9	11	13
21	17	15	13	11	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9	11	13
19	17	15	13	11	9	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9	11	13
19	17	15	13	11	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	7	9	11	13
19	17	15	13	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9	11	13
19	17	15	11	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9	11	13
17	15	13	11	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9	11	13
17	15	13	11	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9	11	13
17	15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	7	9	11	13
17	15	13	11	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	7	9	11	13
17	15	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	7	7	11	13
15	13	11	9	7	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	7	7	11	13
15	13	11	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	7	11	13
15	13	11	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	7	11	11
15	13	11	9	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	7	11	11
15	13	9	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	9	11



(a)



(b)

(c)

Fig. 5.10. Results of the adaptive selection of the filter mask size for a detail of *Lena*'s hat; (a) original image, (b) edge map, (c) chosen mask sizes.

5.4. Experimental Results

This section presents the additional experiments performed using the same samples of corrupted streams of Section 5.3.2. This allows for an evaluation of the improvement obtained with the adaptive filtering procedure compared to the basic version of the concealment technique. As already stated, corrupted images have been obtained from the original pictures in Fig. 5.11, coded using the SOP and SEGMARK markers, five wavelet decomposition levels, single tile, 32×32 code-block size, and corrupted in the first or second decomposition level at bit-planes in the range 27÷25. The PSNR has been used as quality metric and has been calculated comparing concealed or corrupted images with the transmitted images, that is the JPEG 2000 co-decoded images.

Fig. 5.11. Original test images.

The overall results of the basic algorithm are shown in Fig. 5.12. For each test image, the bars represent the PSNR results for different image versions: the corrupted (corr); corrupted subband zero substitution (zero); concealment with best combination of filter size and iteration number (best); and concealment using the fixed values for these two parameters obtained by selecting the average best values over the entire set of images (fixed). Fig. 5.4.a provides the parameters values used for best version (different for each image) and the fixed version (the average values). Note that the concealment is effective in all cases and the average parameters combination works in all cases except for Bank, in which case the concealment brings to further image degradation. This behavior is extremely undesirable, since it invalidates the option of fixing a set of parameters. Some details for the previous images (Bank and Einstein) are provided in Fig. 5.13 in case of best combination of algorithm parameters. For each image, the result of the basic algorithm (right side) is compared with the corrupted image. The quality improvement is particularly significant for the details of Bank, where the artifacts are almost completely removed. On the other hand, the improvement of Einstein is less visible but still important, also considering that the corrupted image presents less artifacts than in the previous case (see the artifacts on the jacket collar).

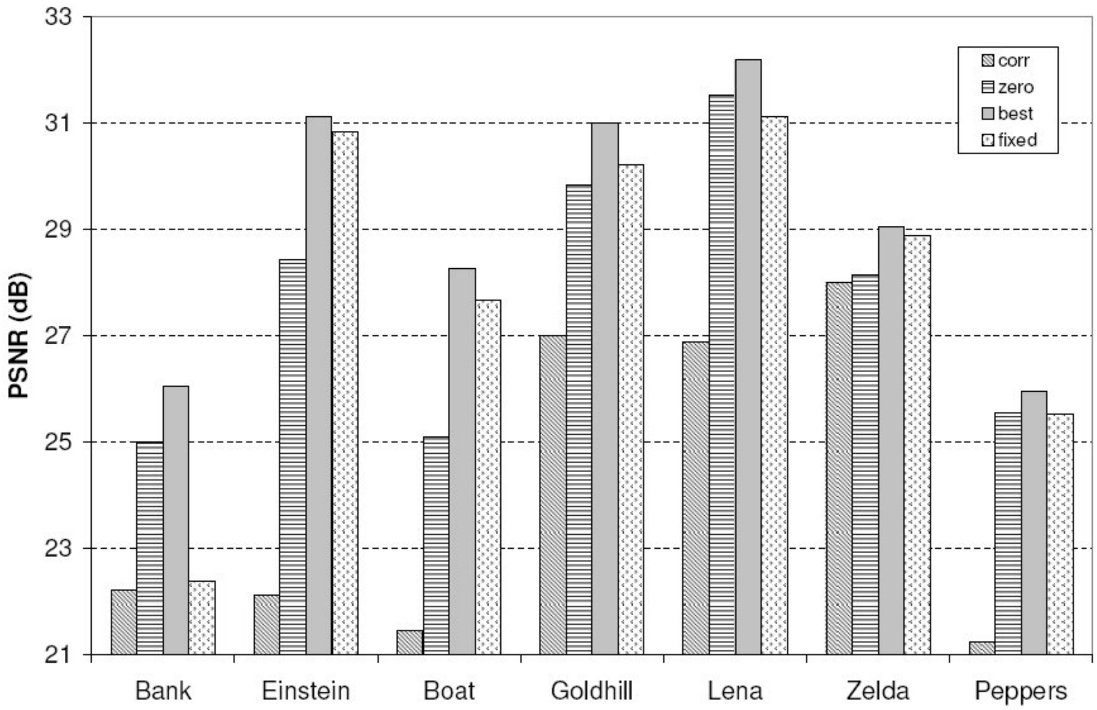


Fig. 5.12. Results of the basic algorithm: corrupted image (corr), zeroing of the damaged subband (zero), concealment with the best and fixed window size and number of iterations.

(a) **(b)**

Fig. 5.13. Details showing the improvement with the basic algorithm with the optimal combination of concealment parameters; (a) *Bank*, (b) *Einstein*, left=corrupted, right=concealed.

Concerning the adaptive algorithm, Fig. 5.14 shows the results for the images Bank and Einstein versus the number of iterations. This time the curve shapes are similar, differently from what observed in Figs. 5.5 and 5.6. An improvement of more than 9dB over the corrupted images has been reached in both cases after 8 and 18 iterations for Einstein and Bank respectively. Moreover, it can be noted that good results are obtained for both images in a common range (19÷27).

To subjectively evaluate the results, some details for images Lena, Boat, and Peppers are shown in Fig. 5.15. The leftmost image is the corrupted version; in the middle is the result of the concealment with the basic algorithm; the rightmost is the result of the adaptive algorithm. In all cases, the adaptive method produces better results than the basic one. In particular, in Fig. 5.15.a, the detail of Lena's shoulder is more clearly defined in the rightmost picture. In Fig. 5.15.b, distortion artifacts are better removed from the Boat's keel with the adaptive method. In Fig. 5.15.c, the black space between the two peppers at the bottom is again better restored with the adaptive algorithm.

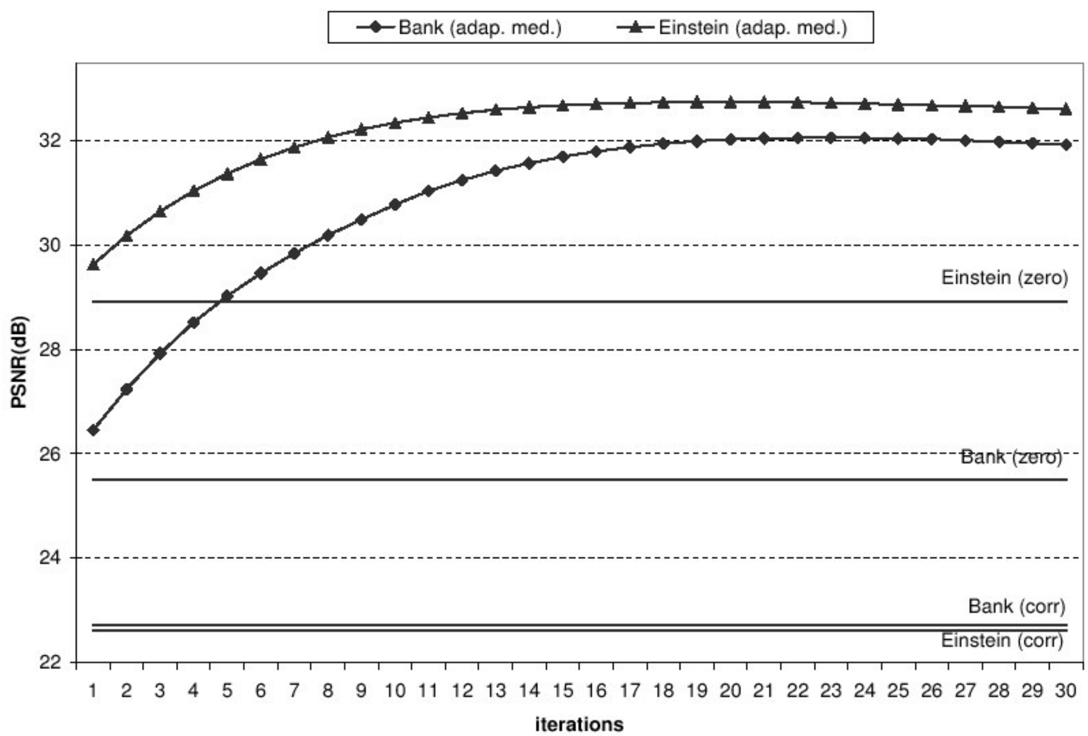
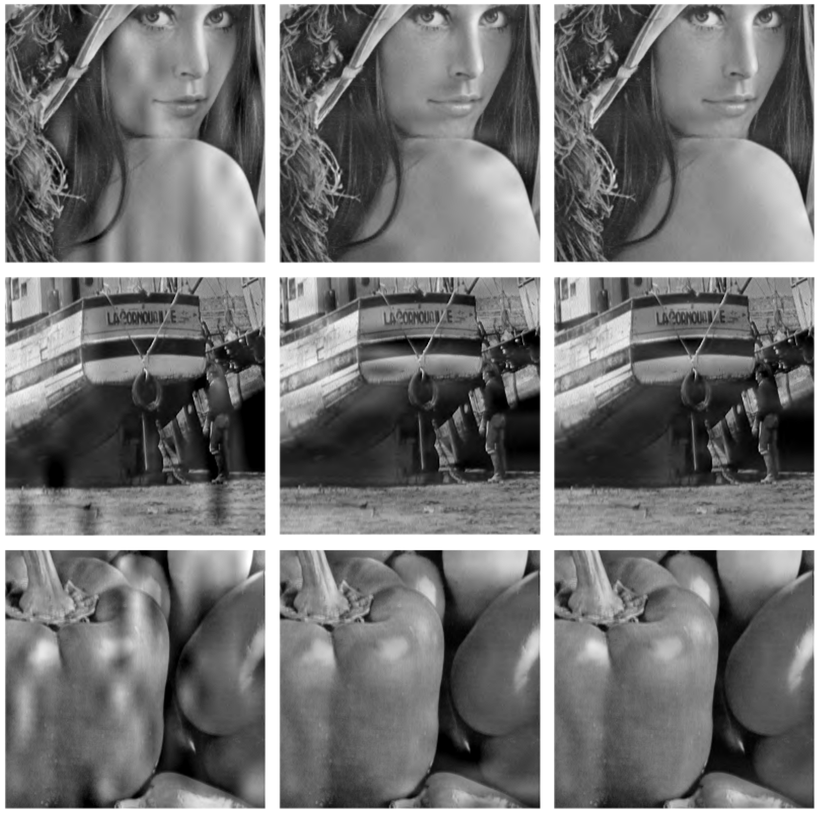


Fig. 5.14. Results of the adaptive algorithm applied to *Bank* and *Einstein*.



(a)

(b)

(c)

Fig. 5.15. Details showing the improvement obtained with the adaptive algorithm; (a) *Lena*, (b) *Boat*, (c) *Peppers*; left=corrupted, center=concealed with basic algorithm, right=concealed with adaptive algorithm.

The overall results obtained with the adaptive algorithm are shown in Fig. 5.16. The first four bars represent the PSNR results for the corrupted version, corrupted subband zero substitution, optimal and fixed iteration number, respectively. The "optimal" results have been obtained by selecting the best results varying the number of iterations for each image, while the "fixed" one are obtained by using the same value for all the images. While an average improvement of almost 4.5dB has been reached, it can be noted that sometimes the "fixed" results are one dB or more lower than the "optimal" ones. Therefore, other stopping criteria have been tested. Among these, the most successful criterion is based on computing the difference between the images resulting at the end of iterations i and $i+1$. The adopted approach relies on the observation that the iterative algorithm converges toward a point within the two convex sets with progresses that decrease increasing the number of iteration. We had to verify if a certain correlation is present between the quality curve (in terms of PSNR between the original image and image at iteration i) and the difference curve (in terms of MSE between images at step i and $i+1$). Fig. 5.17 provides the results of this analysis for the Bank image; similar results are observed for the other images. As expected, the amount of improvement decreases increasing number of iterations. Close to the maximum improvement, the advance becomes smaller. However, passed the maximum improvement point, additional iterations result in a decrease of the image quality. This is due to the phenomenon illustrated in Fig. 5.2.b, where the final point is farer from the original image respect to some intermediate points. The optimal MSE threshold that allows for stopping iterations close to the maximum improvement point has been experimentally determined. In particular, a value of 0.1 has been obtained (Fig.5.17). With this criterion, a specific number of iterations can be computed for each image. All the experiments for the adaptive version of the algorithm have been repeated with the new criterion, obtaining higher PSNR values compared to the 'fixed' results, as shown in Fig. 5.16 (MSE -based bars). The last bar in this figure represents the result of the basic method with the average best combination of parameters.

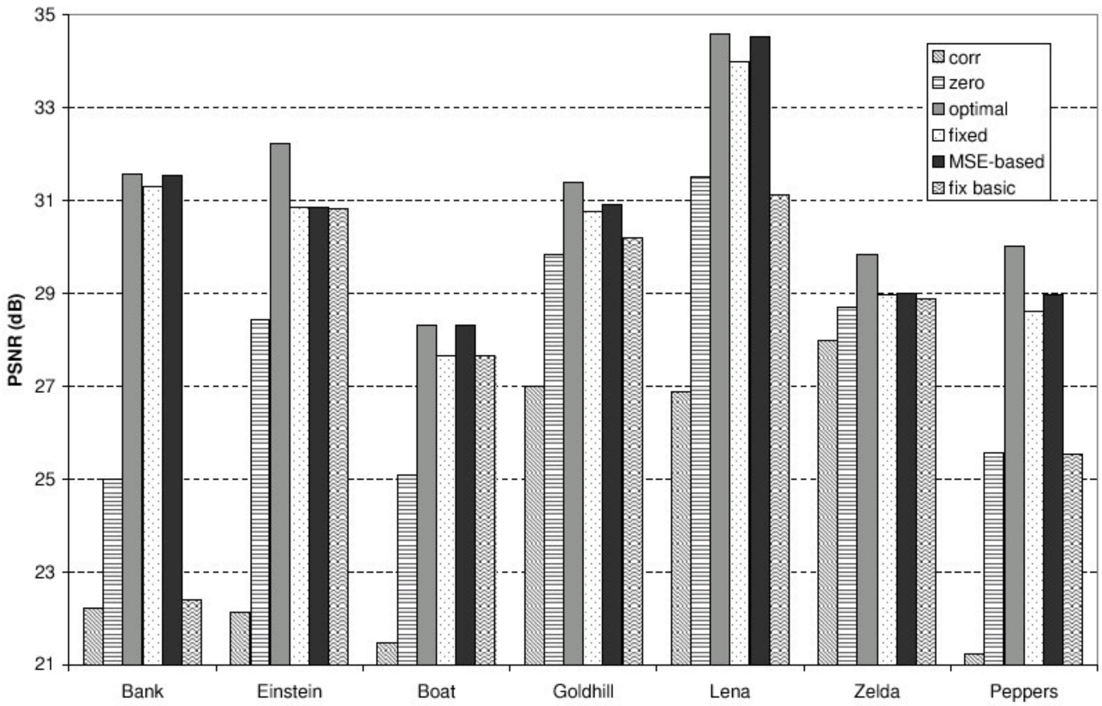


Fig. 5.16. Results of the adaptive algorithm: corrupted image (corr), zeroing of the damaged subband (zero), concealment with the optimal and fixed number of iterations (fixed), concealment when using the MSE-based stopping criteria (MSE-based), and concealment with the basic algorithm with fixed window size and number of iterations (fix basic).

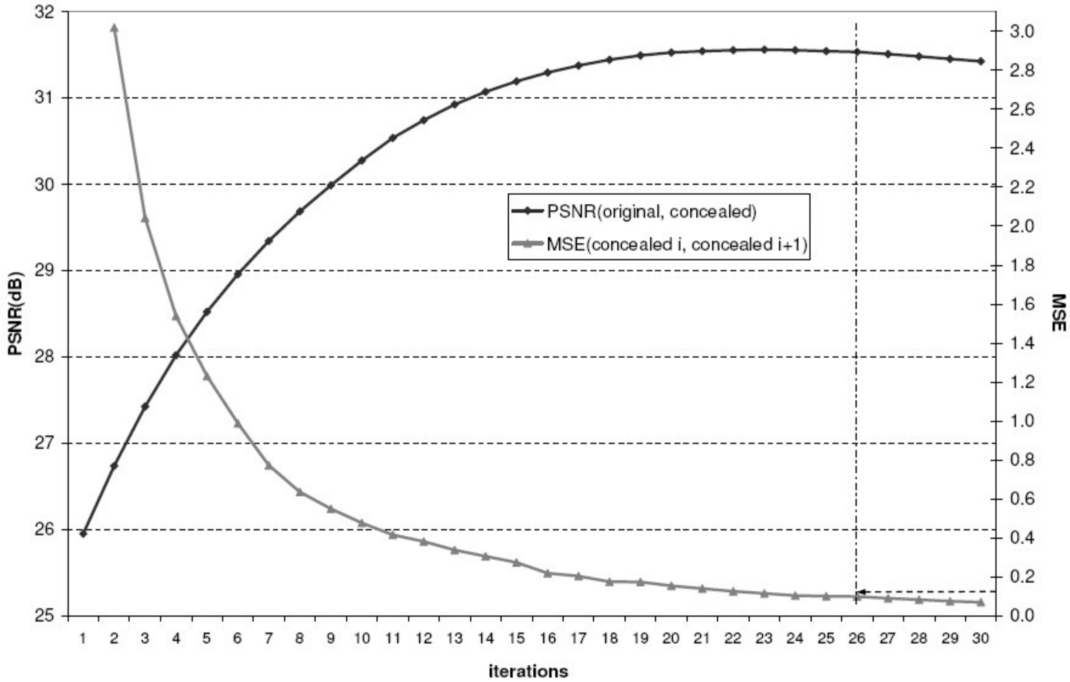


Fig. 5.17. Quality curve (in terms of PSNR between the original image and the resulting image at iteration i) and difference curve (in terms of MSE between the resulting images at step i and $i+1$) for image *Bank*.

To complete the performance analysis of the proposed algorithm, it is important to note that since the first step of our algorithm consists in zeroing all the coefficients of the corrupted subband, the level of the affected bit-plane doesn't influence the resulting image quality after applying the concealment algorithm. However, the extension of the error varies significantly together with the benefits introduced by the concealment. Then, the suitability of our algorithm is connected to the quality degradation of corrupted image, which, in turn, depends on what bit-plane is affected by errors.

To illustrate the performance of our algorithm in case of various affected bit-planes, Table 5.III shows the PSNR for corrupted images and after subband zeroing for various affected bit-planes. In case of zeroing, all the bit-planes are set to zero from the corrupted one to the least significant one. All the results are referred to the LH subband at the first decomposition level. By comparing the results in this table with those shown in Fig. 5.16, it is clear that the performance of our error concealment technique is particularly satisfactory when considering errors in the most significant bit-planes. In case of corruption of the lower bit-planes, the concealment algorithm is ineffective. However, such corrupted images present almost invisible artifacts.

TABLE 5.III. Resulting PSNR for the test images after error corruption and zeroing in the LH subband of the first decomposition level at different bit-planes (the asterisk means that the corresponding bit-plane was not present).

Image		**Bit-plane 27**	**Bit-plane 26**	**Bit-plane 25**	**Bit-plane 24**
Bank	Corrupted	*	22.22	26.68	30.28
	Zeroing	*	25.01	28.53	31.15
Lena	Corrupted	*	26.88	32.68	33.62
	Zeroing	*	31.52	33.02	34.78
Boat	Corrupted	21.46	25.52	27.36	33.16
	Zeroing	25.09	26.01	29.56	35.43
Einstein	Corrupted	22.12	29.61	33.27	34.67
	Zeroing	28.42	30.02	33.89	35.23
Goldhill	Corrupted	*	*	26.99	31.54
	Zeroing	*	*	29.83	32.67
Zelda	Corrupted	*	28.00	31.48	33.21
	Zeroing	*	28.69	32.59	34.12
Peppers	Corrupted	21.23	27.14	28.06	32.09
	Zeroing	25.55	27.89	29.67	33.15

5.5. Conclusions

In this chapter, we investigated the problem of JPEG2000 error concealment at high-frequency in lower decomposition levels. A method based on the theory of projections onto convex sets has been developed by defining two convex sets that take into account the spatial image characteristics and transformed coefficients consistency. As to the first set, a median filtering with a fixed mask size is considered for the implementation of the projection operator. The projection onto the second set is performed by restoring the correctly received wavelet coefficients in the not-corrupted subbands. Experimental results proved a significant average quality improvement with a high variance, which highlighted the dependence of the algorithm on image characteristics. After experimental studies, a size-adaptive median filter was developed to enhance and normalize the performance of our method. Final results show good objective improvements in terms of PSNR with an average gain of about 6.5dB over the corrupted images and 2.9dB over the zeroed images. On the basis of subjective comparisons, it is possible to observe that the visual artifacts deriving from transform coefficients corruption are mostly removed and a good level of perceptual quality is generally restored.

REFERENCES

[Ada+94] R. Adams and L. Bischof, "Seeded Region Growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641-647, Jun. 1994.

[Aia+99] B. Aiazzi, L. Alparone, S. Baronti, F. Lotti, "Virtually-lossless compression of medical images through classified prediction and context-based arithmetic coding," *Visual Communications and Image Processing '99*, Proc. SPIE, Vol. 3653, pp. 1033-1040, San Jose, CA, 1999.

[Alk+00] Z. Alkachouch and M.G. Bellanger, "Fast DCT-based spatial domain interpolation of blocks in images," *IEEE Trans. on Image Processing*, vol. 9, no. 4, pp. 729–732, Apr. 2000.

[Atz+03] L. Atzori, D.D Giusto, A. Raccis, "High-frequency error concealment in wavelet-based image coding," *IEE Electronics Letters*, Vol. 39, No.2, pp. 203-205, Jan. 2003.

[AVI] Airbore Visible/Infrared Imaging Spectrometer - http://aviris.jpl.nasa.gov

[Baj+01] C. Bajaj, I. Ihm, S. Park, "Visualization-Specific Compression of Large Volume Data," *Proc. of 9th Pacific Conf. on Computer Graphics and Appl.*, pp. 212-222, Tokyo, Japan, Oct. 2001.

[Ban+02] B.A. Banister, B. Belzer, T.R. Fisher, "Robust Image transmission using Jpeg2000 and turbo-codes," *IEEE Signal Processing Lett.*, vol. 9, no. 4, pp. 117-119, Apr. 2002.

[Bil+00] A. Bilgin, G. Zweig, M.W. Marcellin, "Three-dimensional image compression with integer wavelet transforms," *Applied Optics*, Vol. 39, pp. 1799-1814, Apr. 2000.

[Bil+03] A. Bilgin, Z. Wu, M.W. Marcellin, "Decompression of corrupt Jpeg2000 codestreams," *IEEE Data Compression Conference*, Snowbird, Utah (USA), 2003.

[CARA] D. Caramella and C. Bartolozzi, 3D Image Processing, Springer, 2002.

[Cho+] S. Cho, D. Kim, W.A. Pearlman, "Lossless Compression of Volumetric Medical Images with Improved 3-D SPIHT Algorithm," *Journal of Digital Imaging*, in press.

[Coh+93] L.D. Cohen and I. Cohen, "Finite-element method for active contour models and balloons for 2D and 3D images," IEE*E Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1131-1147, Nov. 1993.

[COUR] R. Courant and D. Hilbert, Methods of Mathematical Physics, vol. 1, Interscience Publishers Inc., New York, 1955.

[Dau+98] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Anal. Appl.*, Vol. 4, No. 3, pp. 245-267, 1998.

[DICOM] Digital Imaging and Communications in Medicine Standard – http://medical.nema.org/.

[Dro+01] M. Droske, B. Meyer, M. Rumpf, C. Schaller, "An Adaptive Level Set Method for Medical Image Segmentation," *Information Processing in Medical Imaging* (IPMI01), pp. 416-422, 2001.

[Fan+02] Y. Fan, T. Jiang, D.J. Evans, "Volumetric Segmentation of Brain Images Using Parallel Genetic Algorithms," *IEEE Trans. on Medical Imaging*, vol. 21, no. 8, pp 904-909, Aug. 2002.

[Gam+04] J.C. Gamio, S. Belongie, S. Majumdar, "Normalized Cuts in 3D for Spinal MRI Segmentation," *IEEE Trans. on Medical Imaging*, vol. 23, no. 1, pp. 36-44, Jan. 2004.

[Gin+02] B. van Ginneken, A.F. Frangi, J.J. Staal, B.M. ter Haar Romeny, M.A. Viergever, "Active Shape Model Segmentation with Optimal Features," *IEEE Trans. on Medical Imaging*, vol. 21, no. 8, pp. 924-33 Aug. 2002.

[Gin+04] G. Ginesu, D.D. Giusto, W.A. Pearlman, "Lossy to Lossless SPIHT-based Volumetric Image Compression", *IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (ICASSP04), Montreal, Canada, May 21-24, 2004.

[Gok+01] S.B. Gokturk, C. Tomasi, B. Girod, C. Beaulieu, "Medical Image Compression Based on Region of Interest, with Application to Colon CT Images," *Proc. 23rd Int. Conf. of the IEEE on Medicine and Biomedical Engineering*, Istanbul, Turkey, Oct. 2001.

[Ham98] T. Hamid, "DICOM Requirements for JPEG2000," ISO/IEC JTC1/SC 29/WG 1, Report N944, 1998.

[Hem+00] S.S. Hemami, "Robust image transmission using resynchronizing variable length codes and error concealment," *IEEE Journal on Selected Areas in Communications*, vol. 18. no. 6, pp. 927- 939, Jun. 2000.

[Hem+97] S.S. Hemami and R.M. Gray, "Subband coded image reconstruction for lossy packet networks," *IEEE Trans. on Image Processing*, vol. 6, no. 4, pp. 523-539, Apr. 1997.

[Ho+02] S. Ho, Y. Kim, E. Bullitt, G. Gerig, "Medical image segmentation by 3-D level set evolution," Technical Report UNC Chapel Hill, Mar. 2002.

[Ich98] N. Ichimura, "Content-Based Coding of Volume Data Using Finite Mixture Model, Octree and Shape Adaptive DCT," Bulletin of the Electrotechnical Laboratory, Vol. 62, No.7, 1998.

[Ihm+98] I. Ihm and S. Park, "Wavelet-Based 3D Compression Scheme for Very Large Volume Data," *Proc. of Graphics Interface '98*, pp. 107-116, Vancouver, Canada, Jun. 1998.

[Isl+99] A. Islam and W.A. Pearlman, "An embedded and efficient low-complexity hierarchical image coder," Proc. Of SPIE Conf. Visual Comm. and Image Proc., S. Jose, Jan. 1999.

[J2K] ISO/IEC JTC1/SC29/WG1 N15444-1 – "JPEG 2000 Part 1 – Core Coding System".

[J2Kmed] ISO/IEC JTC1/SC29/WG1 N2782 – "JPEG 2000 for Medical Imaging Applications".

[JAIN] A.K. Jain, Fundamentals of digital image processing, Thomas Kailath, Series Editor, 1989.

[JP3D] ISO/IEC JTC1/SC29/WG1 N3022 – "JP3D Requirements 5.0".

[JPWL] ISO/IEC JTC1/SC29/WG1 N2845 – "JPWL simulation enviroment".

[Kas+87] M. Kass, A. Witkin, D. Terzopoulos, "Snake: active contour model," Int. *Journal on Computer Vision*, vol. 1, pp. 321-331, 1987.

[Kha+02] I. Kharitonenko, X. Zhang, S. Twelves, "A Wavelet Transform with Point-Symmetric Extension at Tile Boundaries," *IEEE Trans. on Image Processing*, Vol. 11, No. 12, Dec. 2002.

[Kim+00] B. Kim, Z. Xiong, W.A. Pearlman, "Low Bit-Rate Scalable Video Coding with 3-D Set Partitioning in Hierarchical Trees (3-D SPIHT)," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 10, No. 8, Dec. 2000.

[Kim+99] Y. Kim and W.A. Pearlman, "Lossless Volumetric Medical Image Compression," *Proc. of SPIE Conf. on Applications of Digital Image Proc. XXII*, Vol. 3808, pp. 305-312, Jul. 1999.

[Kud+92] G.R. Kuduvalli and R.M. Rangayyan, "Performance analysis of reversible image compression techniques for high-resolution digital teleradiology," *IEEE Trans. on Medical Imaging*, Vol. 11, No. 3, pp. 430-445, Sept. 1992.

[Kuh+97] G. Kühne, C. Poliwoda, C. Reinhart, T. Günther, J. Hesser, R. Männer, "Interactive Segmentation and Visualization of Volume Data Sets," *IEEE Proc. Visualization*, 1997.

[Kur+03] M. Kurosaki, K. Munadi, K. Nishikawa, H. Kiya, "Error protection for JPEG2000 encoded images and its evaluation over OFDM channel," Proc. 2003 *IEEE International Symposium on Circuits and Systems*, Bangkok, Thailand, 2003.

[Lia+00] J. Liang, B. Luc, A. Lie, J. Wus, F. Kossentini, " Error resilience ad-hoc sub-group report for the Tokyo Meeting," ISO/IEC JTC 1/SC 29/WG 1 N1606, Mar. 2000.

[Lin+01] Z. Lin, J. Jin, H. Talbot, "Unseeded region growing for 3D image segmentation," *Selected papers from Pan-Sydney Workshop on Visual Information Processing*, Sydney, Australia, 2001.

[Mas+04] F. Massidda, G. Ginesu, D.D. Giusto, "Valutazione della Qualità per Immagini Mediche Compresse basata sul Sistema Visivo Umano," *XII Congresso Nazionale SIPF*, Alghero (Italy), Oct. 2004.

[Men+02] G. Menegaz and J. Thiran, "Lossy to Lossless Object-Based Coding of 3-D MRI Data," *IEEE Trans. on Image Processing*, Vol. 11, No. 9, pp. 1053-1061, Sept. 2002.

[Moc+00] I. Moccagatta, S. Soudagar, J. Liang, H. Chen, "Error-resilient coding in JPEG-2000 and MPEG-4," *IEEE Journalon Selected Areas in Communications*, vol. 18, no. 6, pp. 899-914, Jun. 2000.

[Nat+02] A. Natu and D. Taubman, "Unequal error protection of JPEG2000 code-stream in wireless channels," *in Proc. IEEE Global Telecommunications Conference* (GLOBECOM), Taipei, Taiwan, 2002.

[Ney90] D.R. Ney, E.K. Fishman, D. Magid, R.A. Drebin, , "Volumetric rendering of computed tomography data: principles and techniques," *Computer Graphics and Applications, IEEE*, Vol. 10, No. 2, pp. 24-32, Mar. 1990.

[Ngu+01] K.G. Nguyen and D. Saupe, "Rapid High Quality Compression of Volume Data for Visualization," *Eurographics 2001*, Vol. 20, No. 3, Manchester, UK, 2001.

[Nic+03] D. Nicholson, C. Lamy, C. Poulliat, X. Naturel, "Result of core experiments on header error protection (JPWL C01)," ISO/IEC JTC 1/SC 29/WG 1 N 2935, May 2003.

[Ots79] N. Otsu, "A Threshold Selection Method from Gray Level Histograms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.

[PETR] M. Petru and P. Bosdogianni, Image Processing: The Fundamentals, John Wiley & Sons LTD, 2000.

[Pou+01] C. Poulliat and D. Nicholson, "Impact and efficiency of error resilience tools for mobile applications," ISO/IEC JTC 1/SC 29/WG 1 N2176, Jun. 2001.

[Ran+02] S.D. Rane, J. Remus, G. Sapiro, "Wavelet-domain reconstruction of lost blocks in wireless image transmission and packet-switched networks," *Proc. 2002 Int. Conf. on Image Processing* (ICIP2002), New York, 2002.

[Ris91] E.A. Riskin, "Optimum bit allocation via the generalized BFOS algorithm," *IEEE Trans. on Information Theory*, vol. 37, pp. 400-402, Mar. 1991.

[Rod01] F.F. Rodler and R. Pagh, "Fast Random Access to Wavelet Compressed Volumetric Data Using Hashing," BRICS RS-01-34, Aug. 2001.

[Rod99] F.F. Rodler, "Wavelet Based 3D Compression for Very Large Volume Data Supporting Fast Random Access," BRICS RS-99-34, Oct. 1999.

[Sai+J96] A. Said and W.A. Pearlman, "A New, Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, No. 3, pp. 243-250, Jun. 1996.

[Sai+S96] A. Said and W.A. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Trans. on Image Processing*, Vol. 5, No. 9, pp. 1303-1310, Sept. 1996.

[Sch+00] P. Schelkens, X. Giro, J. Barbarien, A. Munteanu, J. Cornelis, "Compression of Medical Volumetric Data," ISO/IEC JTC 1/SC 29/WG 1, Report N1712, Jun. 2000.

[Sch+03] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, J. Cornelis, "Wavelet Coding of Volumetric Medical Datasets," *IEEE Trans. on Medical Imaging*, Vol. 22, No. 3, pp. 441-458, Mar. 2003.

[SUET] P. Suetens, Fundamentals of Medical Imaging, Cambridge University Press, 2002.

[Suh+97] J.-W. Suh and Y.-S. Ho, "Error concealment based on directional interp olation," *IEEE Trans. on Consumer Electronics*, vol. 43, no. 3, pp. 295-302, Aug. 1997.

[Sum+95] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Trans. on Image Processing*, vol. 4, no. 4, pp. 470-477, Apr. 1995.

[Swe96] W. Sweldens, "The Lifting Scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, Vol. 3, No. 2, pp. 186-200, 1996.

[TAUB] D.S. Taubman and M.W. Marcellin, JPEG2000: Image compression fundamentals, standards and practice, Norwell, MA: Kluwer Academic Publishers, 2002.

[Tek+96] H. Tek and B.B. Kimia, "Volumetric segmentation of medical images by three dimensional bubbles," *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 305-322, Nov. 1996.

[Tes+96] A.G. Tescher, J.T. Reagan, J.A. Saghri, "Near lossless transform coding of multispectral images,", *1996 Int. Geoscience and Remote Sensing Symposium* (IGARSS '96), Vol. 2, pp. 1020 – 1022, May 1996.

[Thi+03] J. Thie and D. Taubman, "Opti mal erasure protection assignment for scalable compressed images with tree-structured dependencies," in Proc. *IEEE Int. Conf. on Image Processing* (ICIP2003), Barcellona, Spain, 2003.

[Tom+98] C. Tomasi, R. Manduchi, "Bilateral Filtering for Gray and Color Images," *6th Int. Conf. on Computer Vision*, pp. 839-846. New Delhi, India, 1998.

[Uen+03] I. Ueno and W.A. Pearlman "Region of interest coding in volumetric images with shape-adaptive wavelet transform," *SPIE/IS&T Electronic Imaging 2003*, Proc. SPIE, vol. 5022, Jan. 2003.

[VM6] C. Christopoulos, "JPEG2000 Verification Model 6.0 (Technical description)," ISO/IEC JTC1/SC29/WG1, N1575, Jan. 2000.

[VM8] C. Christopoulos, "JPEG2000 Verification Model 8.6 (Technical description)," ISO/IEC JTC1/SC29/WG1, N1894, Dec. 2000.

[Wan+93] Y. Wang, Q.F. Zhu, L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. on Communications*, vol. 41, no. 10, pp. 1544-1551, Oct. 1993.

[WATT] A.H. Watt, 3D Computer Graphics, Addison Wesley, 1999.

[Wu+03] D. Wu, D.M. Tan, H.R. Wu, "Visually lossless adaptive compression of medical images," *Proc. of the 2003 Joint Conf. of the 4th Inter. Conf. on Information, Communications and Signal Processing*, Vol. 1, pp. 458 – 463, Dec. 2003.

[Wuo+95] S. Wong, L. Zaremba, D. Gooden, H.K. Huang "Radiologic image compression - a review," *Proceedings of the IEEE*, Vol. 83, No. 2, pp. 194 - 219, Feb. 1995.

[Xio+03] Z. Xiong, X. Wu, S. Cheng, J. Hua, "Lossy-to-Lossless Compression of Medical Volumetric Data Using Three-dimensional Integer Wavelet Transforms," *IEEE Trans. on Medical Imaging*, Vol. 22, No. 3, pp. 459-470, Mar. 2003.

[Xu+D98] C. Xu and J.L. Prince, "Generalized gradient vector flow external force for active contour," *Signal Processing - An International Journal*, vol. 71, no. 2, pp. 131-139, Dec. 1998.

[Xu+M98] C. Xu and J.L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *IEEE Trans. on Image Processing*, vol. 7, no. 3, pp. 359-369, Mar. 1998.

[Yan+99] T.-C.I. Yang, S. Kumar, and M. Shen, "Use of RVLC in EBCOT," ISO/IEC JTC1/SC29/WG1 N1253, Mar. 1999.

[You78] D.C. Youla, "Generalized image restoration by the method of alternating orthogonal projections," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 25, pp. 694-702, Sept. 1978.

[Yu+98] G.S. Yu, M.M.K. Liu, M.W. Marcellin, "POCS-based error concealment for packet video using multiframe overlap information," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 422–434, Aug. 1998.

[Zhu+98] W. Zhu, Y. Wang, Q.-F. Zhu, "Second-order derivative-based smoothness measure for error concealment in DCT-based codecs," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 6, pp. 713-718, Oct. 1998.

Publications

Journals

- G. Ginesu, D.D. Giusto, V. Märgner, P. Meinlschmidt, "Detection of Foreign Bodies in Food by Thermal Imagery," *IEEE Trans. on Industrial Electronics*, Vol.51, No.2. April, 2004.
- L. Atzori, G. Ginesu, A. Raccis, "JPEG2000-coded Image Error Concealment Exploiting Convex Sets Projections," *IEEE Trans. on Image Processing*, final acceptance for publication.
- F. Massidda, G. Ginesu, D.D. Giusto, "Quality Assessment of Compressed Medical Images based on Human Visual System," *Journal of Psychophysiology*, Hogrefe&Huber, in press.

Conferences

- G. Ginesu, D.D. Giusto, V. Märgner, P. Meinlschmidt, "Detection of Foreign Bodies in Food by Thermal Imagery," *IAPR Workshop on Machine Vision Applications*, Nara, Japan, 2002.
- G. Ginesu, D.D. Giusto, W.A. Pearlman, "Lossy to Lossless SPIHT-based Volumetric Image Compression," *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP04), Montreal, Canada, May 21-24, 2004.
- L. Atzori, G. Ginesu, A. Raccis, D.D. Giusto "A Novel Iterative Approach for Jpeg2000 Error Concealment," *IEEE International Workshop on Multimedia Signal Processing* (MMSP04), Siena, Italy, September 29 - October 1, 2004.
- F. Massidda, G. Ginesu, D.D. Giusto, "Valutazione della Qualità per Immagini Mediche Compresse basata sul Sistema Visivo Umano," *XII Congresso Nazionale SIPF*, Alghero, Italy, October 8-11, 2004.
- G. Ginesu, D.D. Giusto, "Semantic-Driven Lossy to Lossless Coding of Volumetric Data," *Picture Coding Symposium* (PCT04), San Francisco, USA, December 15-17, 2004.
- G. Ginesu, D.D. Giusto, "3D Region of Interest Segmentation of Volumetric Medical Data with Snakes," *2nd Cairo International Biomedical Engineering Conference* (CIBEC04), Cairo, Egypt, December 27-29, 2004.