Universitat
Autònoma
de Barcelona

Departament d'Enginyeria de la
Informació i de les Comunicacions

# MODEL-BASED JPEG2000
# RATE CONTROL METHODS

SUBMITTED TO UNIVERSITAT AUTÒNOMA DE BARCELONA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

by Francesc Aulí Llinàs
Bellaterra, October 2006

Directed by
Dr. Joan Serra Sagristà

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Bellaterra, October 2006

Dr. Joan Serra Sagristà
(Adviser)

*Committee:*

Dr. Rosa M. Figueras Ventura (New York University)
Dr. Michael W. Marcellin (University of Arizona)
Dr. Josep Rifà Coma (Universitat Autònoma de Barcelona)
Dr. Philippe Salembier (Universitat Politècnica de Catalunya)
Dr. David S. Taubman (The University of New South Wales)

# Abstract

This work is focused on the quality scalability of the JPEG2000 image compression standard. Quality scalability is an important feature that allows the truncation of the code-stream at different bit-rates without penalizing the coding performance. Quality scalability is also fundamental in interactive image transmissions to allow the delivery of Windows of Interest (WOI) at increasing qualities.

JPEG2000 achieves quality scalability through the rate control method used in the encoding process, which embeds quality layers to the code-stream. In some scenarios, this architecture might raise two drawbacks: on the one hand, when the coding process finishes, the number and bit-rates of quality layers are fixed, causing a lack of quality scalability to code-streams encoded with a single or few quality layers. On the other hand, the rate control method constructs quality layers considering the rate-distortion optimization of the complete image, and this might not allocate the quality layers adequately for the delivery of a WOI at increasing qualities.

This thesis introduces three rate control methods that supply quality scalability for WOIs, or for the complete image, even if the code-stream contains a single or few quality layers. The first method is based on a simple Coding Passes Interleaving (CPI) that models the rate-distortion through a classical approach. An accurate analysis of CPI motivates the second rate control method, which introduces simple modifications to CPI based on a Reverse subband scanning Order and coding passes Concatenation (ROC). The third method benefits from the rate-distortion models of CPI and ROC, developing an approach based on a novel Characterization of the Rate-Distortion slope (CoRD) that estimates the

rate-distortion of the code-blocks within a subband.

Experimental results suggest that CPI and ROC are able to supply quality scalability to code-streams, even if they contain a single or few quality layers, achieving a coding performance almost equivalent to the one obtained with the use of quality layers. However, the results of CPI are unbalanced among bit-rates, and ROC presents an irregular coding performance for some corpus of images. CoRD outperforms CPI and ROC achieving well-balanced and regular results and, in addition, it obtains a slightly better coding performance than the one achieved with the use of quality layers. The computational complexity of CPI, ROC and CoRD is negligible in practice, making them suitable to control interactive image transmissions.

# Agraïments

M'agrada pensar que la realització d'aquesta tesi ha estat com el creixement d'un arbre. En les primeres etapes, la llavor ha hagut de germinar, crear profundes arrels i adquirir aliment, energia i il·lusió. El primerenc brot l'ha hagut d'escalfar el sol, de refrescar la pluja i d'empènyer la brisa, convertint-lo primer en un verd i fràgil branquilló, després en un alt i escardalenc tronc i després en l'arbre que, finalment, ha donat el fruit que en aquests moments teniu a les vostres mans. No hi ha faltat el fred, el vent ni les inclemències del temps, però les arrels, profundament clavades a terra, l'han mantingut ferm i sempre intentant créixer amunt.

La llavor germina, les arrels aprofundeixen, les fulles transpiren i un fruit, primer petit, tímid i, encara molt tendre, apareix en una de les branques. No és només el resultat de l'esforç d'un. El reg de cada dia, l'abono, la poda, el desbrossament de males herbes i les atencions diàries volen una gran dedicació. En aquesta tasca ha estat el meu director, en Joan, el qui ha portat el pes. Amb ell vàrem escollir el lloc on crèiem que hi havia terra fèrtil per plantar-hi la llavor original, i m'agrada pensar que no ens vàrem equivocar. Ha estat ell qui ha vetllat perquè l'arbre creixés en la direcció adequada, enfortint-lo i fent-lo madurar perquè, finalment, donés fruit. Per tot, moltes gràcies.

El procés de fotosíntesi transforma el raig de sol en matèria viva i un nou brot fa espurnejar noves fulles. Infinitat d'actors actuen en molts diferents moments *but, throughout the development of this thesis there have been some worth moments that I like to remember. In the first steps of this research Prof. Michael Marcellin and Prof. David Taubman encouraged me to continue studying. I heartily thank them for their interest in this modest research and for traveling to Barcelona for the public defense of this thesis.* M'agrada pensar en el seu contacte com aquella pluja que refresca la terra després de la sequera, mitigant la calor i revivint aquell brot amb ganes de créixer.

El fruit creix, madura, cau i arriba l'hora de recollir-lo per avaluar-lo. És el tribunal d'aquesta

dissertació qui s'encarrega de fer-ne el tast, comprovant-te la dolçor i fent-ne l'informe final. Sóc conscient que és una tasca llarga i que desitjo no esdevingui tediosa, i per tot això també vull agrair a tots els membres del mateix la lectura i aprovació, si s'escau, d'aquest fruit. En especial, també agraeixo al Dr. Philippe Salembier la seva ràpida resposta en la realització de l'informe de qualitat.

Un arbre no pot ni tan sols despuntar sense un dels ingredients que fan d'aquest planeta un formigueig d'éssers vius de tota classe: l'aire. És el departament d'Enginyeria de la Informació i de les Comunicacions l'aire que he respirat cada dia. Vull agrair la tasca que l'equip directiu, amb en Josep Rifà al capdavant, està fent, amb rigor i seriositat digne d'exemple. M'agrada pensar que podré seguir respirant aquest aire durant molt de temps.

Les envestides del temps, amb gelades, pedregades i fins i tot inundacions, són indubtablement moments difícils que tots hem de combatre, resistir i, sempre, intentar vèncer. Els arbres solitaris estan indefensos i dèbils davant les adversitats. És sempre millor rodejar-se de bons companys de viatge, formant un bosc, alt i frondós, amb el qual tenir bona companyonia. M'agrada pensar que, tant en el grup de compressió d'imatges com en el departament, fem una bona pinya. En especial, agraeixo a la Cristina el bon ambient i amistat que ha crescut en les incomptables hores i hores de despatx que hem compartit, les vesprades de conversa i tabola amb el Joaquín i el Guille, a tots els membres de grup GICI per compartir codis i coneixements i a l'Alaitz, del CREAF, per facilitar-nos amb tota diligència les imatges del corpus de teledetecció.

Tota llavor prové d'un fet intranscendent i sense el qual aquest planeta perdria tot sentit. El solem anomenar amor. És d'amor d'on tots provenim i com tots transcendim. I és amb amor amb el qual tot cobra sentit. Amb tota seguretat, aquest és l'amor dels meus pares. Són ells qui es mereixen l'agraïment amb el més gran amor i afecte. El seu exemple diari sobre com enfocar la vida, la seva infinita paciència i la seva inacabable dedicació a nosaltres ha estat i és, per mi, la guia diària i el mirall amb el qui més vegades intento reflectir-me. La llavor original, va ser la seva. M'agrada pensar que els agrada el fruit d'aquella llavor. Amb ells, la meva germana Teresa, el meu cunyat Joan i el meu nebot Bernat són el recolzament vital que em sosté.

I, finalment, el més gran element: el sol. El sol converteix aquest planeta en terra fèrtil i plena de vida. És l'astre rei el que ens il·lumina, ens escalfa i ens guia. Gràcies Mayte, per ser el meu sol.

Aquesta tesi ha gaudit del suport del Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya, del Ministeri d'Educació i Ciència del govern espanyol i del Fons Social Europeu, amb les ajudes 2006FIC-00189, TIC2003-08604-C04-01 i SGR2005-00319.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

The use of images is becoming every day more popular in our society: remote sensing centers have currently more and more available images of the earth surface, the medical community is already used to the diagnosis with images, and in our every-day lives digital cameras with high resolution capabilities are becoming usual. It is well known that images need high storage capacities, and that compression can help to reduce the huge storage requirements needed in some centers.

The type of information within an image is highly redundant. An efficient encoding of the original samples can reduce in several degrees the amount of information needed to store the image. Besides, depending on the scenario, the encoding process can allow a loss of information, which helps to enhance the compression efficiency. When the image is encoded through a process which produces some loss of the original information, the compression process is called lossy. When no loss of information is allowed, the compression is referred to as lossless. Near-lossless compression stands for an encoding process which allows a slightly loss of information, usually achieving higher compression ratios than lossless compression without penalizing the visual quality of the decoded image.

Apart from compression, the manipulation of images currently requires other advanced features. Some of these features are the availability to transmit images interactively over the network, to support error resilience, or even to supply capabilities of watermarking and fingerprinting. Encoding systems must take these needs into account to provide a flexible framework that allows an efficient management.

Image compression techniques have been developed during almost twenty years. The most popular technique used in image compression until 1991 was the Discrete Cosine Transform [3], adopted in the popular JPEG standard [39, 73]. Since 1992, the Discrete Wavelet Transform [6] has been applied in most coding systems, being EZW [83] and SPIHT [77] the first coding systems which benefited from this technique.

In this scenario, the Joint Photographic Experts Group [47] decided in 1997 to develop a new image compression standard that supplies advanced features with the use of the most advanced techniques in image compression. The core coding system of the standard (Part 1) was published in December 2000 [42].

This thesis began in January 2003 with the study of the state-of-the-art of still image compression techniques. Comparing among the coding systems found in the literature, JPEG2000 was one of the most outstanding standards, supplying advanced features. From January 2004 the Part 1 of JPEG2000 was studied and implemented. The development of a new JPEG2000 implementation was aimed to create a framework suitable to easily test new proposals within the standard. Besides, the implementation helped to understand all the particularities within the core coding system. The suite of applications BOI [36] is the result of this development and it is nowadays being integrated in a GIS application from the Center for Ecological Research and Forestry Applications (Spain) [15].

Throughout the JPEG2000 study and BOI implementation, several ideas have been tested aimed to enhance some of the features of the standard. One of those ideas was the implementation of a rate control method devised to fulfill some special requirements. From June 2005 we decided to perform a more in-depth research of the rate control in JPEG2000, which has become the main topic of this thesis.

We present three rate control methods. One of the most important features of them is that they do not use any distortion measure based on the original image, which is the main difference compared to the methods found in the literature. The developed rate control methods are model-based, estimating the rate-distortion of the encoded image. In order to obtain highly competitive results, the last rate control method introduced in this thesis uses a novel rate-distortion characterization of the fractional bit plane coding of JPEG2000.

## 1.2 Fundamental concepts

A planar image is a two-dimensional matrix of sample values $x[n_1, n_2]$ with size $N_1, N_2$ in the vertical and horizontal directions respectively. The samples have a valid range of values, which is referred to as bit-depth, setting the number of bits needed to represent each sample value. More precisely, if an image has a bit-depth of $B$, the samples values can express $2^B$ different values of intensities in the range

$$x[n_1, n_2] \in \{0, 1, 2, ..., 2^B - 1\} \text{ with } 0 \leq n_1 < N_1, 0 \leq n_2 < N_2 \, .$$

One of the most common bit-depth for digital images is $B = 8$, and the range of values may vary depending on the integer representation of the sample. When the image has a single two-dimensional matrix of values, or also referred to as a single *component*, it commonly represents a gray-scaled image. Colour images are usually represented as the Red, Green and Blue (RGB) primary colours in three separate components, although the representation may vary depending on the format. In general, an image can be compounded by $C$ components, with $x_c[n_1, n_2], 0 \leq c < C$.

When an image with a single component is not compressed, it needs $N_1 \cdot N_2 \cdot B$ bits to represent the sample values. The encoding process generates an encoded representation of the image to a string of bits which is called encoded bit-stream, or *code-stream*, denoted as $\mathcal{X}$. The purpose of the compression is to minimize the length of the code-stream in order to achieve high compression factors. The compression factor is defined as

$$\text{compression factor} = \frac{N_1 \cdot N_2 \cdot B}{length(\mathcal{X})}$$

where $length(\mathcal{X})$ denotes the number of bits of the code-stream $\mathcal{X}$. The inverse of the compression factor is the *compression ratio*. Another measure of compression commonly used is the bit-rate, referred to as $R$ and expressed in *bits per sample* (bps). The bit-rate represents the number of bits used in $\mathcal{X}$ to represent each sample of the image, and it is calculated as

$$R = \frac{length(\mathcal{X})}{N_1 \cdot N_2} .$$

When using lossy compression, the most common measure to compare the recovered image with the original one is the *Mean Squared Error* (MSE), which is defined as

$$MSE = \frac{1}{N_1 \cdot N_2} \cdot \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])^2$$

where $\hat{x}$ denotes the samples decoded from the code-stream $\mathcal{X}$. In image compression, the MSE is commonly expressed with the reciprocal measure *Peak Signal to Noise Ratio* (PSNR), in *decibels* (dB), which is calculated according to

$$PSNR = 10 \cdot log_{10} \cdot \frac{(2^B - 1)^2}{MSE} .$$

The samples of an image are usually encoded in a bit-plane by bit-plane fashion. If the binary representation of a sample is denoted as $x_b = b_{B-1}b_{B-2}...b_1b_0$, the bit-plane $p$ is the binary representation of the bit $b_p$ of all samples. The minimum number of bits needed to represent a sample value is called the number of *magnitude bit-planes* of that sample, and it is referred to as $K$, with $0 \leq x < 2^K$. The first bit different from $0$ of the binary representation of a sample is $b_{K-1}$ and it is called the *most significant bit*. When the encoding is performed in a bit-plane by bit-plane fashion, it is said that the sample $x$ becomes significant at bit-plane $K - 1$. The bits $b_{K-2}...b_0$ are called *refinement bits*.

A code-stream can be embedded, progressive, scalable, or any combination of these three properties. An *embedded* code-stream $\mathcal{X}$ means that, if $R^j, R^{j+1}$ denote any two bit-rates of $\mathcal{X}$ satisfying $R^j < R^{j+1}$, the code-stream of bit-rate $R^j$ is a prefix of the code-stream of bit-rate $R^{j+1}$. This implies that to decode the code-stream at bit-rate $R^{j+1}$, all bits until the bit-rate $R^j$ must also be decoded. A code-stream is *progressive* for an specified measure, say for example in terms of distortion, if there exists a set of bit-rates $\{R^1, R^2, ..., R^n\}$ with $R^1 < R^2 < ... < R^n$ at which the distortion of the decoded image is $D^1, D^2, ..., D^n$ satisfying $D^1 > D^2 > ... > D^n$. Within a *scalable* code-stream there is a set of identifiable layers $L^1, L^2, ..., L^n$ of bit-rate $R^1, R^2 - R^1, ..., R^n - R^{n-1}$ such that the code-stream of bit-rate $R^j$ is a prefix of the code-stream of bit-rate $R^{j+1}$, implying that layer $L^j$ must be decoded before layer $L^{j+1}$.

A code-stream can be scalable by *quality* (or distortion), by *resolution*, by *spatial location* or by *component*. Each type of scalability means that the layers within the code-stream identify different qualities/resolutions/spatial locations/components of the image. Besides, any combination of these scalabilities is possible: a scalable code-stream by quality and resolution means that, within the code-stream, layers of quality and resolution are identifiable. Note also that a smart organization of the layers gives different combinations of progression, for instance progression by quality and resolution means that quality layers are ordered in increasing quality and within each quality layer there are a set of identifiable resolution layers ordered in increasing resolution.

We call *coding system* to the complete set of algorithms, techniques and formats needed to encode and decode an image, specified in a compression standard or in a technical document. The *coding process* is the process followed by a coding system in order to encode or decode an image. *Encoding* means that the input of the coding process is an image, producing an output code-stream. *Decoding* is the process where a code-stream is taken as input and produces the image it represents as output.

*Coding performance* denotes the efficiency of a coding system in terms of rate-distortion. For instance, a coding system $A$ has a better coding performance than $B$ at bit-rate $R$, if $A$ is able to generate a code-stream which recovers the image with less distortion than $B$, at the same bit-rate $R$. The *computational complexity* of a coding process means the number of operations executed to perform the encoding or decoding of an image.

Some mathematical notation used in this thesis is: $|x|$ denotes the absolute value of $x$; $sign(x)$ gives 1 when $x \geq 0$, 0 otherwise; the floor operation is represented as $\lfloor x \rfloor$;

$x \ mod \ y$ denotes the integer modulus between $x$ and $y$; and the binary and hexadecimal representations of a value are denoted as $\mathrm{bbbb}_b$ and $\mathrm{HHHH}_h$ respectively.

## 1.3 Thesis organization

This thesis allows a lineal reading: Chapter 2 presents the main concepts and techniques of the JPEG2000 core coding system, Chapter 3 introduces the rate-distortion optimization problem and reviews the state-of-the-art of JPEG2000 rate control methods. Section 4.1 in Chapter 4 introduces the motivations followed to develop our original rate control methods, setting the main requirements. The remaining of Chapter 4 introduces a simple rate control method based on a Coding Passes Interleaving (CPI). An accurate analysis of CPI motivates the rate control method named Reverse subband scanning Order and coding passes Concatenation (ROC), introduced in the following Chapter 5. Chapter 6 presents a rate control method completely different from CPI and ROC, based on a novel Characterization of the Rate-Distortion slope (CoRD). Chapter 7 contains a brief summary of this work and our conclusions.

A JPEG2000 expert would probably skip Chapter 2 as well as Sections 3.1.1, 3.1.2 of Chapter 3. A reader familiar with rate control in JPEG2000 would probably skip the whole Chapter 3, although we suggest the reading of the last experimental Section 3.2.3, that describes the experiments performed in the following chapters.

Along the development of the rate control methods and in the experimental sections, we use the corpus of images of the ISO/IEC standard 12640-1 [40], which is compounded by the eight natural images depicted in Figure 1.1. The most extensive experimental section is in Chapter 6, where a great variety of experiments are presented to asses and compare various features of the developed rate control methods. In this section we also use two more image corpus, compounded of medical and remote sensing images respectively.

All the experimental tests have a label specifying the image features, coding parameters and implementations versions in order to allow the repetition of the experiments. The JPEG2000 implementations used in this thesis are BOI [36] and Kakadu [96]. The latter

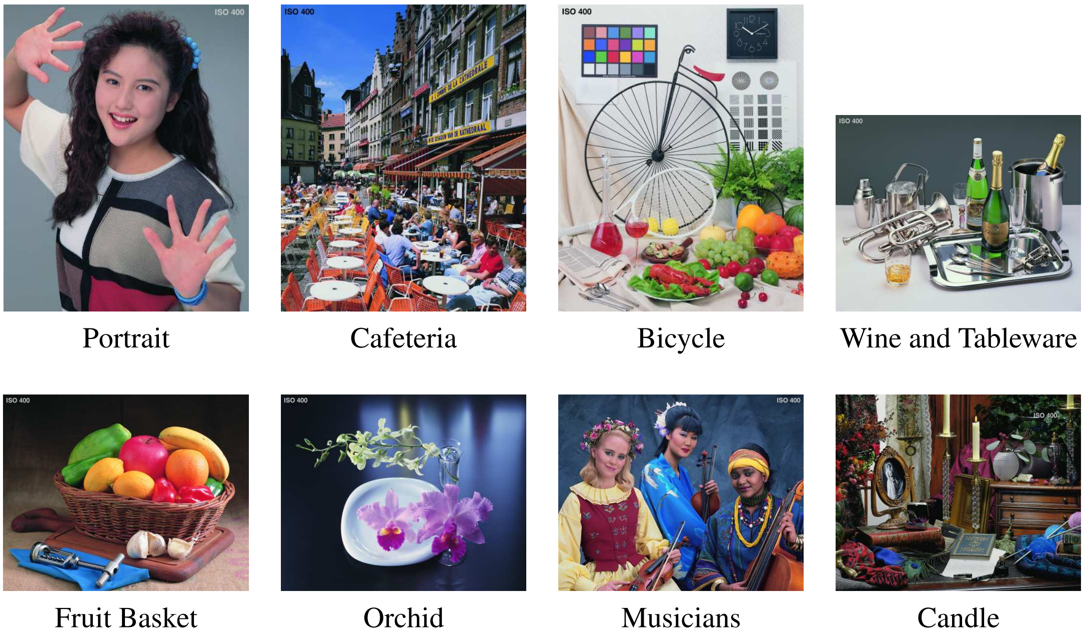



Figure 1.1: *Images of the corpus ISO 12640-1.*

is one of the most optimized implementations of the standard, developed by Dr. David Taubman, one of the major contributors of JPEG2000.

## 1.4 Thesis contributions

The main contributions of this thesis are the rate control methods CPI, ROC and CoRD. CPI was presented at the Data Compression Conference in March 2006 [12] and ROC has been recently accepted for publication to the IEEE Signal Processing Letters journal [11]. From our point of view, CoRD is the most outstanding contribution of this work, and it is foreseen to be submitted to a journal after the public defense of this dissertation.

The identification of the JPEG2000 drawbacks, which motivated the development of the presented rate control methods (Section 4.1), and the extensive review and comparison among the JPEG2000 rate control methods (Section 3.1) are also issues of interest.

We consider the suite of applications BOI, first introduced in [9], another important contribution, although the hard work spent on its development is not reflected in this document. Besides, the techniques tested within the core coding system [10], and the review of the state-of-the-art coding systems [82], performed at the beginning of this research, are also contributions of this thesis.

# Chapter 2

# The JPEG2000 standard

This chapter presents the fundamental concepts and techniques within the core coding system of JPEG2000, which are widely used throughout the development of this thesis. The main source of this chapter, and an excellent guide of this research, has been the JPEG2000 book [100], by D. Taubman and M. Marcellin.

## 2.1 Introduction

The international standard JPEG2000 has been developed by the Joint Photographic Experts Group [47] (JPEG) to address different aspects related with image compression, transmission and manipulation. Although the standard was initially structured in six parts, in 2001 six more parts were proposed. The main purposes of each part are summarized in Table 2.1.

As of October 2006, Parts 1, 2, 3, 4, 5, 6, 9 and 12 are international standards, Part 7 has been abandoned, Parts 8 and 11 are in the last stages of standardization, and Part 10 is still under development. JPEG2000 Part 1 [42] was the first part published as International standard in December 2000, and it defines the technologies defined in the core coding system, which are also used in the remaining parts. The research presented in this thesis is mainly focused in JPEG2000 Part 1, referred to as simply JPEG2000 from now on, although some of the developed algorithms could also be used in JPEG2000 Part 9 [46].

Table 2.1: *Description of the twelve parts of the JPEG2000 standard.*

| |
|---|
| • **JPEG2000-1 Core coding system**: description of the minimal decoder and a simple code-stream syntax. This part has a limited number of options in order to facilitate the interchange among applications. |
| • **JPEG2000-2 Extensions**: extensions of the core coding system, providing advanced coding features which can be used to enhance the coding performance or to manipulate unusual data types. This part also provides an enhanced file format. |
| • **JPEG2000-3 Motion JPEG2000**: extensions of the core coding system devised to support the manipulation of image sequences. |
| • **JPEG2000-4 Conformance testing**: information for the compliance and conformance among JPEG2000 implementations. |
| • **JPEG2000-5 Reference software**: two implementations of the core coding system: JJ2000 [29], developed in Java, and JasPer [1], developed in C. |
| • **JPEG2000-6 Compound image file format**: additional file format for tailored and compound documents. |
| • **JPEG2000-7**: this part has been abandoned. |
| • **JPEG2000-8 Secure JPEG2000**: description of a code-stream syntax for interpreting secure image data and a normative process for registering JPSEC tools. |
| • **JPEG2000-9 Interactivity tools, APIs and protocols**: description of the transmission protocol JPIP, devised to interactively transmit JPEG2000 images. |
| • **JPEG2000-10 Volumetric JPEG2000**: coding of volumetric data, providing enhanced coding features for floating point data. |
| • **JPEG2000-11 Wireless JPEG2000**: description of error protection techniques for JPEG2000 code-streams aimed to detect and correct errors produced during the code-stream transmission. |
| • **JPEG2000-12 ISO base media file format**: definition of the ISO file media file, providing an extensible format which facilitates interchange, management and editing. Common with [45]. |

Although the purpose of this document is not to review the JPEG2000 process of standardization[1], in this section we briefly describe the main features of the standard to better understand its fundamental characteristics. The development of the standard began with a call for technical contributions, issued in March 1997 [48], with a list of the desired features that the core coding system of the standard should contain, summarized in Table 2.2.

The selection and evaluation of the submitted algorithms lasted for more than two years;

[1]A wide explanation of the JPEG2000 standardization process can be found in [100, Chapter 9].

Table 2.2: *Desired features for the JPEG2000 standard [100, Chapter 9.2].*

| |
|---|
| • *Superior low bit-rate performance*: high coding performance for all bit-rates, specially at low bit-rates and with respect to the previous standard JPEG [39, 73]. |
| • *Continuous-tone and bi-level compression*: independent coding of different image components, each with 1 to 16 bit depth, using one unified architecture. |
| • *Progressive transmission by pixel accuracy and resolution*: progression, in order to increase the image quality or resolution when transmitting the code-stream over communication links. |
| • *Lossless and lossy compression*: both types of compression within one unified architecture. |
| • *Random code-stream access and processing*: access and decoding of image areas without needing to decode the whole code-stream. |
| • *Robustness to bit-errors*: ability to manage bit errors introduced in the code-stream. |
| • *Sequential build-up capability*: the coding architecture should not need to buffer the entire image to produce the code-stream . |

finally a slightly modified version of the Embedded Block Coding with Optimized Truncation [93] (EBCOT) was adopted as the core coding system. All the desired features set out in the first Call for Technical Contributions were achieved using an integrated coding architecture and a flexible code-stream syntax. Moreover, some of the initial desired features were extended and a few new ones were added. The most important features of the core coding system are related in Table 2.3.

JPEG2000 has become a powerful standard that provides even more features than those initially planned. The technologies supported by the standard have been described in many different papers. To mention only some of them, the JPEG2000 features and the most important techniques used in it are reviewed in [88, 87]; an in-depth overview of the core coding system is described in [76, 101]; the modified version of EBCOT used in JPEG2000 is presented in [97]; and, more recently, a review of the current state of the standard has been written in [52]. The excellent book [100] by Taubman and Marcellin provides the most detailed description of the standard, accompanied by some theory of image coding.

Table 2.3: *Features provided by JPEG2000 Part 1 [100, Chapter 9.2].*

• *Compress once, decompress many ways*: one of the most important features of JPEG2000 is the possibility to decode a selected image area at different qualities and/or resolutions without needing to process the whole code-stream.
• *Compressed domain image processing/editing*: simple operations such as rotation, mirroring or flipping, resizing and quality reduction can be performed without needing to re-encode the image.
• *Progression*: JPEG2000 defines progressive transmission by quality, resolution, spatial location and component. These progressions can be mixed, producing a more powerful progression orders. The re-organization of a code-stream to another progression order does not compel to re-encode the image.
• *Low bit-depth imagery*: binary valued components can be coded by adjusting the coding parameters, although resolution and quality scalability is not available. Images containing a limited number of colours can also be encoded as a single component thanks the definition of look-up-tables.
• *Region of interest coding*: arbitrary image regions can be prioritized by encoding them first, without modifying the coding pipeline.

## 2.2 The core coding system

### 2.2.1 Overview

The core coding system of JPEG2000 is constituted by four main stages: sample data transformations, sample data coding, code-stream re-organization, and rate control. The three first stages are considered as the coding pipeline, whereas the rate control may involve different techniques affecting some operations carried out in these stages. Figure 2.1 depicts the stages and operations of the whole core coding system.

*Sample data transformations* is the first stage. It is devised to prepare the image samples for the next coding stage by compacting the energy of the image to achieve better coding performance. Besides, this stage also modifies the range of the sample values to simplify certain implementation issues. When using floating point processing in the operations of colour and wavelet transform carried out in this stage, the coding performance is highly increased, although then lossless compression is not supported. To allow lossless and lossy compression without sacrificing coding performance, both integer and floating point data are supported in those operations, depicted in Figure 2.1 with two different coding paths (reversible/irreversible). The region of interest coding is also carried out in this stage by a simple operation which prioritizes the coefficients of the selected regions in such a way that the following coding stage encodes those regions first. A more detailed explanation of these operations is found in Section 2.2.2.

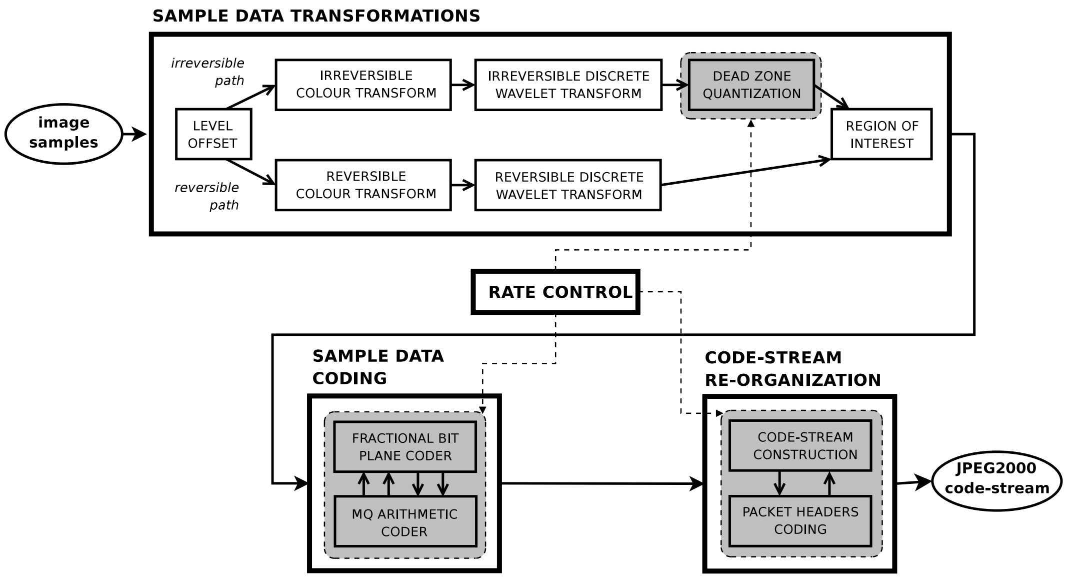



Figure 2.1: *JPEG2000 core coding system: main stages and operations.*

Before coding, the image is partitioned in a hierarchical structure. This operation is arguably not considered as a stage of the coding process because no operations related with coding is performed, however it plays an important role in the coding system and is discussed in Section 2.2.3. The smallest structures provided by these partitions are called code-blocks, which are small blocks of coefficients that are independently encoded by the *sample data coding* stage. This coding stage uses a fractional bit plane coder and an arithmetic coder, and produces an embedded code-stream for each code-block. The process carried out in this stage is also called Tier-1 coding and is reviewed in Section 2.2.4. The construction of the final JPEG2000 code-stream is performed by the *code-stream re-organization* stage, explained in Section 2.2.5, which implies a re-organization of the generated code-streams and a coding of some more information related to code-blocks. The coding of this information is also called Tier-2 coding.

The main function of the *rate control* stage (or also called rate-distortion optimization stage depending on the techniques used in it) is to manage the bit-rate and/or the distortion of the final code-stream produced by the coding system. When the user specifies a desired bit-rate, the rate control optimizes the quality of the final code-stream (minimizes the distortion); when a desired quality is specified, the rate control optimizes the bit-rate of the code-stream (minimizing its length). Besides, the rate control manages the construction of quality layers, which are important structures of the code-stream devised to provide quality scalability, a fundamental feature of the standard. The JPEG2000 standard defines a guideline for the rate control based on the formulation proposed in EBCOT [93], however this stage can be implemented in different ways by modifying the quantization, the sample data coding or the code-stream re-organization stages. This is why more than twenty different rate control methods have appeared since 2002. The analysis of these rate control methods is presented in Chapter 3.

JPEG2000 uses many different techniques, however, in our opinion three of them stand out as the keys of the coding system. The first one is the *image division in code-blocks and their independent coding*, which allows random-access and independent processing and it provides spatial scalability. Thanks to code-blocks, the code-stream can be interactively transmitted using the JPIP transmission protocol, defined in JPEG2000 Part 9. The main difficulty when dividing the image in code-blocks is how to encode efficiently these code-blocks. This difficulty is overcome by the optimized coding of the *bit plane and arithmetic coder*, which constitute the second key of the standard. The sample data coding is highly efficient due to the discrimination of different coefficient contexts, used by the arithmetic coder to widely reduce the amount of bits generated. This allows the achievement of state-of-the-art coding performance. The third key is the *code-stream organization and syntax*, which defines an efficient and flexible organization in containers, providing advanced features such as different progression orders. It is worth noting that these techniques have been developed, or adapted, specifically for JPEG2000.

### 2.2.2 Sample data transformations

#### Level offset

The first operation of the sample data transformations stage is the level offset. When the image samples of an image component, referred to as $x[n]$, are non-negative with a bit-depth of $B$ bits, an offset of $-2^{B-1}$ is added to obtain a sample representation in the range $-2^{B-1} \leq x[n] < 2^{B-1}$. The main purpose of this operation is to assure that the coefficients obtained by the discrete wavelet transform belonging to the LL subband have a symmetric distribution around 0, which avoids numerical overflow and facilitates the coefficient context specification. Level offset is carried out optionally in each image component and can be easily implemented when the image is loaded. Part 2 of the standard [44] allows a generalized level offset that supports user-specified subtraction values for each image component.

**Remark 2.2.1** *The level offset operation has negligible effects on the coding performance. For instance, the code-stream obtained when encoding without level offset the Candle image at 5 bps is only 0.003% larger than the one obtained when encoding with level offset.*

#### Colour transform

When the three first components of the image contain the red, blue and green samples of a colour image and they have identical size and bit-depth, a colour transform can be optionally carried out to convert the RGB data into a different representation. For lossy compression, RGB is converted to the $YC_bC_r$ representation, where $Y$ stands for the image luminance and $C_bC_r$ for the blue and red chrominance respectively. This is the irreversible colour transform (ICT), and is defined as

$$\begin{pmatrix} x_Y[n] \\ x_{C_b}[n] \\ x_{C_r}[n] \end{pmatrix} = \begin{pmatrix} 0.2999 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{pmatrix} \begin{pmatrix} x_R[n] \\ x_G[n] \\ x_B[n] \end{pmatrix}$$

while the inverse operation is given by

$$\begin{pmatrix} x_R[n] \\ x_G[n] \\ x_B[n] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.402 \\ 1 & -0.344136 & -0.714136 \\ 1 & 1.772 & 0 \end{pmatrix} \begin{pmatrix} x_Y[n] \\ x_{C_b}[n] \\ x_{C_r}[n] \end{pmatrix} .$$

Lossless compression uses the reversible colour transform (RCT), which approximates the ICT by decorrelating image colour. RCT is defined as

$$x_{Y'}[n] = \lfloor \frac{x_R[n] + 2x_G[n] + x_B[n]}{4} \rfloor,$$

$$x_{D_b}[n] = x_B[n] - x_G[n], \quad x_{D_r}[n] = x_R[n] - x_G[n]$$

while the inverse operation is given by

$$x_G[n] = x_{Y'}[n] - \lfloor \frac{x_{Db}[n] + x_{Dr}[n]}{4} \rfloor,$$

$$x_B[n] = x_{D_b}[n] - x_G[n], \quad x_R[n] = x_{D_r}[n] - x_G[n] .$$

When using the RCT, implementations should take into account that samples of $D_b$ and $D_r$ components have one more bit-depth than the original RGB components. The colour transform can be easily implemented when loading the image and its use is recommended in order to achieve better coding performance.

**Remark 2.2.2** *Although it depends on the image, the colour transform highly improves the efficiency of the sample data coding stage. For lossless compression, the code-stream bit-rate can decrease from 7% to more than 15%, and for lossy compression, from 20% to more than 35%.*

### Discrete wavelet transform

The discrete wavelet transform (DWT) decorrelates the image samples, generating a compact representation of the image that can be encoded efficiently without producing blocking artifacts at high compression factors. Besides, the DWT produces a multi-resolution representation of the image in increasing resolution levels, fitting perfectly with the original JPEG2000 desired features.

The one dimensional (1D) DWT can be understood as a successive application to an original sequence of samples, referred to as $x[n]$, of a pair of low-pass and high-pass filters, called analysis filter-bank and referred to as $(h_L, h_H)$ respectively. After the application of the filter-bank, a sub-sampling operation is carried out, discarding every second sample and producing the subband sequences $y_L[n]$ and $y_H[n]$ for the low-pass and high-pass filters respectively. It is worth noting that the filter-bank is designed to allow a perfect reconstruction of the original signal, when there is not any quantization error, using another pair of low-pass and high-pass filters, called synthesis filter bank and referred to as $(g_L, g_H)$ respectively. Filter-banks which allow perfect reconstruction are called *bi-orthogonal*, an important property for image applications.

The low-pass filter eliminates high frequencies of the image, obtaining a blurred version of the original samples, whereas the high-filter preserves high frequencies, as edges and details, eliminating low frequencies. The final sequence $y[n]$ contains the filtered samples output from the DWT, which are referred to as wavelet coefficients (or just coefficients) and has the same length of the original $x[n]$, obtained by interleaving $y_L[n]$ and $y_H[n]$ as

$$y[2n] = y_L[n] \quad y[2n+1] = y_H[n] \; .$$

The extension of the 1D DWT to the two dimensions of an image component considers the application of the filter-bank and sub-sampling to each row of the image and then to each column. This operation produces four different subbands, which contain the Low-Low (LL), High-Low (HL), Low-High (LH) and High-High (HH) frequencies in horizontal and vertical respectively. The LL subband can be decomposed successively into four smaller subbands, increasing the number of DWT levels and obtaining the usual dyadic image decomposition in increasing resolution levels, first attributed to Mallat [61]. Figure 2.2 depicts two common representation of the dyadic decomposition of an image in three DWT levels. Each label $SS_d$ indicates the subband type by $SS$ and the decomposition level by $d$.

**Remark 2.2.3** *In general, it is recommended to use 5 or 6 DWT levels. For example, at 0.1 bps the Orchid image is recovered at 37.54 dB when using 5 DWT, 36.59 dB when using 3 DWT (lossy compression).*

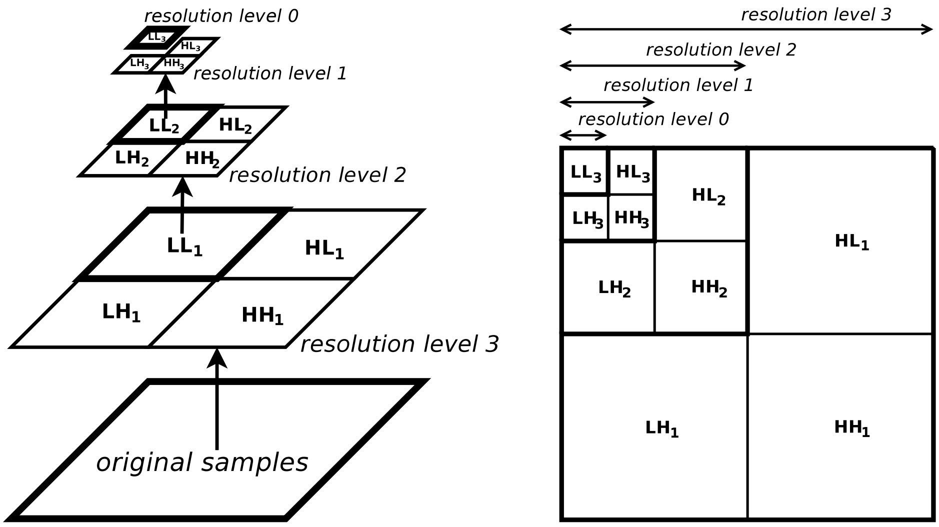



Figure 2.2: *Two common representations of the dyadic decomposition.*

Although JPEG2000 Part 2 supports the specification of the filter-bank, only two filter-banks are allowed in Part 1. The Daubechies 9/7 filter-bank, introduced in [6], is used for lossy compression; lossless compression uses the Le Gall 5/3 filter-bank, first proposed in [34]. The coefficients output from the DWT are usually multiplied by a normalization factor to maintain the range of the coefficients between certain bounds. Actually, this operation of normalization can be performed multiplying the filter-bank by the normalization factor, saving some unnecessary operations. The normalized filter-banks used by JPEG2000 are given in Table 2.4. Both filter-banks are bi-orthogonal.

Table 2.4: *Normalized coefficients of the 9/7 and 5/3 filter-banks of JPEG2000 [76].*

**9/7 filter-bank**

| z | $h_L(z)$ | $g_L(z)$ | z | $h_H(z)$ | z | $g_H(z)$ |
|---|---|---|---|---|---|---|
| 0 | +0.6029490182 | +1.1150870525 | $-1$ | +1.1150870525 | 1 | +0.6029490182 |
| $\pm1$ | +0.2668641184 | -0.5912717631 | $-2,0$ | +0.5912717631 | $+0,2$ | -0.2668641184 |
| $\pm2$ | -0.0782232665 | -0.0575435262 | $-3,1$ | -0.0575435262 | $-1,3$ | -0.0782232665 |
| $\pm3$ | -0.0168641184 | +0.0912717631 | $-4,2$ | -0.0912717631 | $-2,4$ | +0.0168641184 |
| $\pm4$ | +0.0267487574 | | | | $-3,5$ | +0.0267487574 |

**5/3 filter-bank**

| z | $h_L(z)$ | $g_L(z)$ | z | $h_H(z)$ | z | $g_H(z)$ |
|---|---|---|---|---|---|---|
| 0 | 3/4 | 1 | $-1$ | 1 | 1 | 6/8 |
| $\pm1$ | 1/4 | 1/2 | $-2,0$ | 1/2 | $0,2$ | -2/8 |
| $\pm2$ | -1/8 | | | | $-1,3$ | -1/8 |

Even using normalized filter-banks, some unusual combinations of samples could still violate the range boundaries. JPEG2000 manages this situation by defining a number of guard bits, explained in the following quantization operation.

*Orthonormality* is another important property of transformations, and is fulfilled when the sum of the squares of the original samples is equal to the sum of the squares of the coefficients produced by the transformation, expressed as

$$\sum_n x[n]^2 = \sum_n y[n]^2 \ .$$

When a transformation is orthonormal, the impact of a coefficient quantization is easily quantified, providing an important mechanism to estimate the MSE of the reconstructed image. This property is fundamental for the rate control but, unfortunately, the DWT is not orthonormal. A method which, under certain assumptions, approximates the MSE of the reconstructed image by defining the L2norm for each DWT subband is proposed in [106]. The L2norm is the MSE contribution of the subband to the reconstructed image, and it can be used by the quantization stage to calculate efficient step sizes for each subband, or by the rate control, to estimate the distortion contribution of each quantized coefficient to the overall MSE.

The first implementations of the DWT were not efficient in terms of computational complexity and, in addition, they needed to buffer the entire image in memory. These drawbacks have been overcome with a computationally efficient method [90, 25] called lifting scheme and with a line-based processing method [23] that widely reduces the memory consumption. JPEG2000 applications are encouraged to implement both methods.

### Quantization

The quantization is the operation that maps the coefficients provided by the DWT to an index, which will be encoded. Since lossless compression needs a reversible process, quantization is not used when compressing in lossless mode and the process is reduced to a ranging operation, explained below. Lossy compression uses an uniform scalar dead-zone quantization that defines a step size parameter $\triangle_b$ for each subband $b$ in order to produce a signed integer representation, referred to as $q_b$, of the wavelet coefficients $y_b[n]$. This operation is defined as

$$q_b[n] = sign(y_b[n]) \cdot \left\lfloor \frac{|y_b[n]|}{\triangle_b} \right\rfloor .$$

As Figure 2.3 depicts, all the intervals defined by the dead-zone quantization have the same $\triangle_b$ size, except from the interval $(-\triangle_b, \triangle_b)$, which has a size of $2\triangle_b$ and is called the "deadzone".

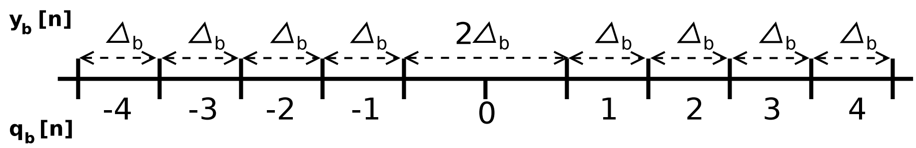



Figure 2.3: *Dead-zone quantizer with a step size of* $\triangle_b$.

The step size $\triangle_b$ of each subband is usually determined taking into account properties of the human visual system or the L2norm of each subband. In JPEG2000, $\triangle_b$ is represented with a mantissa $\mu_b$ of 11 bits and an exponent $\varepsilon_b$ of 5 bits according to the expression

$$\triangle_b = 2^{R_b - \varepsilon_b} \cdot (1 + \frac{\mu_b}{2^{11}})$$

where $R_b$ represents the maximum number of bits that coefficients in subband $b$ can have, called the nominal dynamic range. Although better coding performance may be achieved using different step sizes for each subband, the standard also supports a derived quantization, which derives $\mu_b$ and $\varepsilon_b$ of each subband from the step size assigned to the $LL$ subband ($\triangle_{LL}$) according to

$$(\varepsilon_b, \mu_b) = (\varepsilon_{LL} - L + n_b, \mu_{LL})$$

where $L$ denotes the number of DWT levels, $n_b$ the DWT level of the subband $b$, and $\varepsilon_{LL}$, $\mu_{LL}$ the parameters of $\triangle_{LL}$.

**Remark 2.2.4** *Coding performance is slightly penalized by the use of derived quantization. At 0.5 bps, the Fruit Basket image is recovered only 0.02 dB worse when using derived quantization than when using the L2norms to quantize each subband (lossy compression).*

Since the bounds of the nominal dynamic range can be violated by the coefficients produced by the DWT, JPEG2000 performs a process of ranging specifying a number of guard bits $G$, from 0 to 7, which determines the real bounds of the magnitude of wavelet coefficients conforming to

$$\left\lfloor \frac{|y_b[n]|}{\triangle_b} \right\rfloor < 2^{M_b}$$

where $M_b = \varepsilon_b + G - 1$ referring to the number of magnitude bit-planes of subband $b$.

Part 2 of the standard supports a generalized uniform scalar dead-zone quantization, which allows a deadzone interval of variable width, as well as a trellis coded quantization. A review on how these quantization methods work is described in [63].

**Region of interest**

JPEG2000 defines one type of Region Of Interest (ROI) coding called *Maxshift*. However, thanks to the EBCOT paradigm, an *implicit ROI coding* can also be performed by just adjusting some parameters of rate control. Implicit ROI coding causes that the more ROI coefficients a code-block has, the more prioritized the code-block is by the rate control. An advantage of this method is that it allows the ROI modification by just re-sequencing code-blocks in the final code-stream. However, ROIs are restricted to code-block boundaries. A more in-depth explanation of this implicit ROI coding can be found in [100, Chapter 10.6]. Rate control mechanisms are also used to define code-blocks ROIs and to organize them into different prioritization levels in [79].

The Maxshift method defined in the standard allows the prioritization of arbitrary spatial regions by modifying the wavelet coefficients of these ROIs, referred to as $q[r]$. This modification is carried out up-shifting by a factor of $s$ the ROI coefficients, that is

$$q'[r] = q[r] \cdot 2^s \; .$$

The selection of $s$ is as follows: $s = max(M_b)$, ensuring that ROI coefficients are encoded before background coefficients. This also causes that once the code-stream is constructed, the ROI can not be modified. Note that the wavelet coefficients belonging to the ROI may vary depending on the filter-bank due to the length of the low-pass and high-pass filters, therefore it is important to select them adequately. When decoding, these coefficients are easily identified and down-shifted.

**Remark 2.2.5** *The sample data coding efficiency is penalized when ROIs are encoded. For example, when encoding the Musicians image with a ROI of 25% the size of the image, the resulting code-stream is about 6% larger than when encoding without any ROI. However, it is only need to transmitt 1.2 bps to recover the ROI perfectly (lossy compression).*

JPEG2000 Part 2 supports another type of ROI called *generalized scaling method*, which allows different scaling factors to different ROIs, overcoming some of the drawbacks of Maxshift. Further discussion of Maxshift can be found in [8], and a comparison between Maxshift and implicit ROI is presented in [66]. Besides, in the last four years, novel ROI coding methods have appeared supplying new advanced features. To mention only some of them, in [104, 56, 109, 55] the up-shifting method of ROI coefficients is slightly modified in order to encode ROI and background simultaneously, and in [86] a method which uses the generalized scaling method to improve the visual ROI quality is presented.

### 2.2.3 Sample data partitions

JPEG2000 defines a sophisticated partitioning system that provides some of the advanced features of the standard. The first partition is the *tile*, which defines rectangular regions of the image with the same size each (except when crossing image boundaries). Each component within the tile is called *tile-component*. The main purpose of defining tiles is that each tile-component can be coded independently and that each tile generates an independent code-stream. The tile partition allows a sequential processing of large images, and supplies an efficient mechanism for memory constrained applications. However, the use of tiles causes blocking artifacts at high compression factors, while sequential processing can also be obtained by using line-based DWT processing.

The number of DWT levels, referred to as $L$, can be specified for each tile-component, creating $L + 1$ distinct image *resolution levels*, each one twice the size of the previous one when starting from the LL subband. Recall from Section 2.2.2 that each resolution level have four *subbands*, apart from the lowest resolution level that only contains the LL subband. This successive partition in resolution levels is fundamental to provide progressive transmission by resolution.

**Remark 2.2.6** *The use of tiles is not recommended because they cause blocking artifacts at high compression factors. See, for instance, a region of the Bicycle image recovered at 0.15 bps when it is encoded with and without tiles in Figure 2.4. For interactive image transmission, tiles are also not recommended [94].*

*Tiles of size 256x256 (23.870 dB)*

*No tiles (24.287 dB)*

Figure 2.4: *Visual comparison of coding with and without tiles.*

An important property of the EBCOT paradigm is the partition of each subband in small blocks of coefficients called code-blocks, which are encoded independently. Before the code-block partition, JPEG2000 defines the division of each resolution level in rectangular regions called *precincts*. The precinct size can be specified at each resolution level and it partitions each subband of that resolution level in spatial regions of the same size. The same spatial region of the subbands belonging to that resolution level is referred to as a precinct. The purpose of defining precincts is to facilitate the access to spatial regions of the image. Precinct size determines the minimum accessible region size when decoding image areas of that resolution level. This is caused due to the organization of the code-stream, further discussed in Section 2.2.5.

Finally, each precinct is partitioned in *code-blocks* of same size for the whole tile-component (except when crossing subband or precinct boundaries). Note that the size of the code-block determines the minimum size of precincts, and that the minimum number of code-blocks that a precinct must contain is one. The partition of precincts and code-blocks supplies two advanced features: random code-stream access, and processing and edition in the compressed domain.

**Remark 2.2.7** *When code-blocks dimensions are small, the efficiency of the sample data coding stage is penalized. The code-stream bit-rate of the Cafeteria image fully encoded with code-blocks of size 16x16 is 4% larger than when using code-blocks of size 64x64. At*

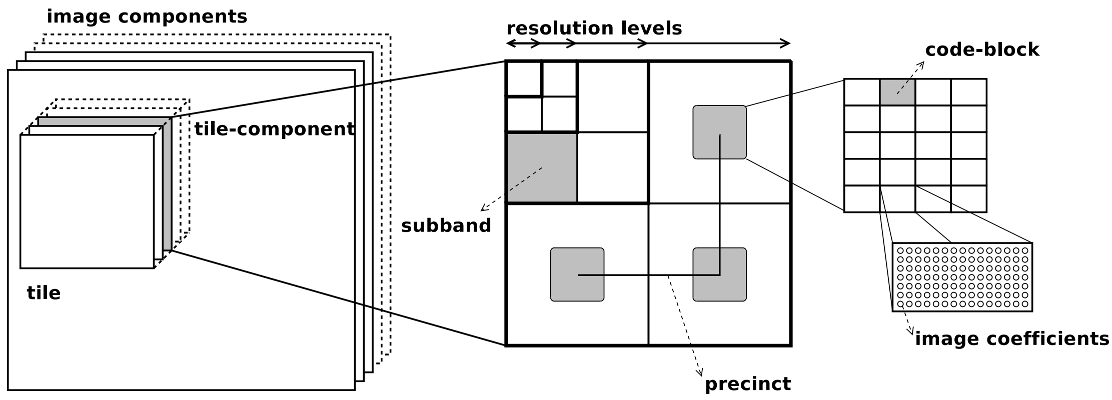



Figure 2.5: *The JPEG2000 partitioning system.*

*1 bps the Cafeteria image is recovered at 32.59 dB when using 64x64 code-blocks; at 31.98 dB when using 16x16 code-blocks (lossy compression).*
*Small precinct sizes usually slightly penalize the coding performance. The code-stream of the Cafeteria image fully encoded using the smallest precinct size is 1% larger than when the precinct size is the largest one. At 1 bps, the PSNR difference between the images recovered using these code-streams is 0.17 dB (lossy compression).*

Figure 2.5 depicts the partitions defined by JPEG2000. Because different components may have different dimensions and they could have different sub-sampling (or spacing) factors, JPEG2000 defines a single reference grid called *canvas*. Every sample of every tile-component is mapped to a location of the canvas, allowing a consistent representation of the image components. Together the mapping rules and the definition of the component partitions are usually referred to as the *canvas coordinate system*.

### 2.2.4 Sample data coding

The sample data coding stage produces an embedded code-stream that can be truncated at different points for each code-block independently. The coding is carried out using a fractional bit plane coder based on EBCOT [93] and an arithmetic coder called MQ-coder, a modified version of the Q-coder [89] developed by IBM and also adopted in the JBIG standard [43]. The main idea behind the coding performed in this stage is to benefit from the spatial redundancy of the coefficients within the code-block, creating favorable conditions for the MQ-coder to code as efficient as possible the data provided by the bit plane coder. These favorable conditions are achieved by using contextual information of the coefficients, which is utilized in the MQ-coder to precisely adjust the probabilities of the incoming symbols. The adequate context selection used in JPEG2000 is a fundamental key in the sample data coding stage because it makes work the MQ-coder efficiently. This issue has been reviewed recently in [57], concluding that the JPEG2000 context selection is already highly efficient.

A detailed explanation of the techniques involved in the bit plane coder and in the MQ-coder can be found in [97], and they are beyond the purpose of this document. Here we only present a rough description of the operations performed in this stage.

### The fractional bit plane coder

Let $K_i$ be a sufficient number of bit-planes to represent all the coefficients of the code-block $\mathcal{B}_i$. The fractional bit plane coder of JPEG2000 encodes each coefficient of $\mathcal{B}_i$ from the highest bit-plane $p = K_i - 1$ to the lowest bit-plane $p = 0$. Each bit-plane $p$ of code-block $\mathcal{B}_i$ is encoded in three sub-bit-plane coding passes: significance propagation pass $\mathcal{P}_i^{(p,2)}$, magnitude refinement pass $\mathcal{P}_i^{(p,1)}$, and cleanup pass $\mathcal{P}_i^{(p,0)}$. The aim of this sub-bit-plane coding is to encode first that information which supplies the greatest reductions of the image distortion. Besides, the sub-bit-plane coding produces an embedded code-stream with a large collection of potential truncation points (each one at the end of each coding pass), which are used by the rate control.

$\mathcal{P}_i^{(p,2)}$ visits the neighbours of already significant coefficients, which have a high probability of becoming significant, $\mathcal{P}_i^{(p,1)}$ codes the magnitude refinement of already significant coefficients, and $\mathcal{P}_i^{(p,0)}$ visits the remaining coefficients not being visited in $\mathcal{P}_i^{(p,2)}$. The scanning order used by these three coding passes is stripe-based, meaning that the coefficients are organized in rows containing four coefficients each one. The scanning in each stripe is from top to bottom and from left to right, as Figure 2.6 depicts. Obviously, $\mathcal{P}_i^{(p,2)}$ and $\mathcal{P}_i^{(p,1)}$ are empty coding passes at the highest bit-plane, i.e. $p = K_i - 1$, therefore the coding begins with $\mathcal{P}_i^{(K_i-1,0)}$.

Because at the highest bit-planes most coefficients are insignificant, the bit plane coder defines a run mode devised to reduce the number of emitted bits. This mode is only used in the cleanup pass $\mathcal{P}_i^{(p,0)}$ and emits a single symbol when the four coefficients of one column of the stripe are insignificant and do not have any significant neighbour.

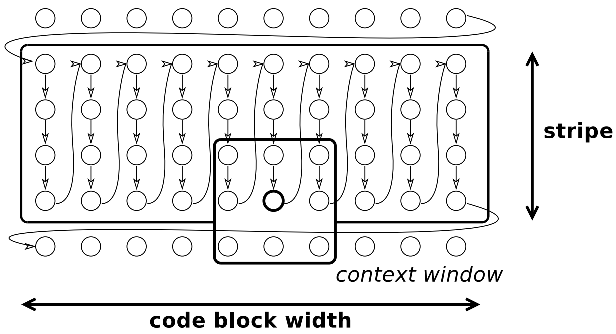



Figure 2.6: *Stripe-based scanning order defined by JPEG2000 [100, Chapter 12.2.3].*

JPEG2000 defines nineteen different contexts. The significance coding (i.e. coding passes $\mathcal{P}_i^{(p,2)}$ and $\mathcal{P}_i^{(p,0)}$) defines the context depending on the significance state of the neighbour coefficients (depicted in the context window of Figure 2.6). The main idea behind this definition is that the current coefficient is more likely to be significant in the current bit-plane if its neighbours have been significant in the current or higher bit-planes. Although $2^8 = 256$ different contexts could be defined, JPEG2000 groups some configurations in nine contexts. The sign coding takes five more contexts, also selected depending on the neighbourhood, and the run mode uses two more contexts. Finally, for the magnitude refinement encoding only three contexts are defined, selected depending on the number of already emitted refinement bits.

Some authors have presented modifications of the bit plane coder in order to achieve better coding performance. A re-arranging of code-block columns is performed before starting the sample data coding in [60], the significance coding is modified using a smart scanning order in [72] and the cleanup pass is modified to reduce the number of emitted symbols in [2].

### The MQ-coder

The main idea behind arithmetic coding is to map an input sequence of symbols $x_n$ to a single code-word by recursive interval partitioning based on the probabilities of the input symbols. The MQ-coder adopted by JPEG2000 uses several techniques to supply advanced features while providing high efficiency. The MQ-coder is *binary* because the input symbols, referred to as $x_n$, are bits (i.e. $x_n \in \{0, 1\}$), causing that there is a single probability $\Sigma$ of the most probable symbol MPS $s, s \in \{0, 1\}$. The MQ-coder is *adaptive* because $\Sigma$ is successively adapted depending on the encoded symbols and $s$ may change its value under certain conditions. Therefore, implementations need to maintain the tuple $\Sigma, s$ in order to determine which is the current state of the coder. For implementation reasons, $\Sigma$ is not computed as a floating point number; instead there are forty-six different fixed probabilities.

The MQ-coder does not have a single state. Each context defined by the bit plane coder, and referred to as $k$, has its own tuple $\Sigma_k, s_k$. Since the symbols encoded by each context represents different kinds of information, the probabilities of input symbols are better approximate taking into account their context. This is how the context is used by the MQ-coder in order to achieve a high coding efficiency. Obviously, the input symbols must be tuples of symbol and context (i.e. $x_n, k_n$). The use of this technique makes the MQ-coder *contextual*.

The interval partitioning of arithmetic coding is efficiently represented by two registers. However, the encoding process may cause an overflow condition which compels to modify the registers by shifting a carry bit. This is a drawback for practical implementations because long registers need to be maintained before emitting the final code-word. MQ-coder is *byte-oriented*, meaning that once 8 bits are encoded, one byte is flushed. Overflow situations are overcome thanks to a technique called *bit-stuffing*, which detects and prevents carry bits adding one redundant bit before flushing the output byte. This does not penalizes the coding performance (only about 0.05%) and, besides, ensures that the range of a byte following an $\mathrm{FF}_h$ byte does not exceed $8\mathrm{F}_h$. The remaining range $\mathrm{FF90}_h$ through $\mathrm{FFFF}_h$ is used by code-stream markers. This assures that the code-word produced by the MQ-coder does not contain some byte combinations used in error resilience methods to detect transmission errors. Besides, some modifications to the coding techniques of the MQ-coder can supply enhanced methods of error resilience, as presented in [35].

The termination procedure of the MQ-coder should take into account that when the decoder runs out of bytes, a $\mathrm{FF}_h$ reading is simulated in order to keep decoding. This may be used to achieve the optimal termination described in [100, Chapter 12.3.2], by eliminating the last unnecessary bytes of the code-word. Recently, five optimized architectures of the MQ-coder that use concurrency techniques to process multiple symbols simultaneously have been proposed in [28].

#### Other considerations

JPEG2000 supports six minor variations in some operations carried out in the sample data coding stage, described in Table 2.5. These variations can be implemented by slightly modifying the bit plane coder and the MQ-coder and they are devised to provide more capabilities to implementations related to parallelization, computational complexity reduction and error resilience. RESET, RESTART and CAUSAL allow parallelization of coding processes. BYPASS supplies a reduction on the computational complexity of the coding without sacrificing coding performance, and ERTERM and SEGMARK provide simple error resilience mechanisms.

**Remark 2.2.8** *Although coding variations supply some advantages to applications, all of them penalize the coding performance. Table 2.6 compares the different coding variations.*

Table 2.5: *JPEG2000 coding variations.*

| |
|---|
| • *RESET*: this variation resets the tuples $\Sigma_k, s_k$ at the beginning of each coding pass. |
| • *RESTART*: this variation resets the internal registers of the MQ-coder at the beginning of each coding pass and terminates the code-word at the end of each coding pass. The use of this variation compels to encode the bit-rate of each coding pass segment. |
| • *CAUSAL*: when using this variation, the context window of the bit plane coder (used to calculate the coefficient context) does not cross stripe boundaries. |
| • *BYPASS*: when the current bit-plane $p$ has a low value, the context probabilities of coding passes $\mathcal{P}_i^{(p,2)}$, $\mathcal{P}_i^{(p,1)}$ are usually not well approximated, widely decreasing the efficiency of the MQ-coder. With this variation MQ-coder is not used when $p < K_i - 4$ at coding passes $\mathcal{P}_i^{(p,2)}$, $\mathcal{P}_i^{(p,1)}$, terminating the segment of the code-word at each pass. |
| • *ERTERM*: the use of this variation surrenders the freedom of the coder of terminating the code-word in a fashion way, and a predictable termination must be used. |
| • *SEGMARK*: this variation encodes the four symbols $1010_b$ at the end of each bit-plane. |

### 2.2.5 Code-stream organization

JPEG2000 defines a flexible code-stream organization, mostly derived from the partitions discussed in Section 2.2.3 and supported by a rich syntax. This organization encapsulates the code-streams produced by the sample data coding stage using different containers to supply different degrees of scalability and progression efficiently. The main containers of the code-stream are called *tile-streams*, *tile-parts* and *packets*. The tile-stream contains all the data corresponding to one image tile (including all its components), and a JPEG2000 code-stream may contain one or more tile-streams. Each tile-stream is comprised by one or more tile-parts, which contain collections of packets, the smallest identifiable container within the whole code-stream.

The sample data coding stage produces a code-stream for every code-block $\mathcal{B}_i$ that can be truncated at different points, referred to as $\mathcal{B}_i^s$. A packet $\mathcal{T}_P$ is a container that encapsulates some code-stream segments of code-blocks belonging to the same precinct $P$. Because the precinct corresponds to an spatial location of one resolution level of one component of one tile, we could define the packet as a quality increment of that spatial

Table 2.6: *Coding performance penalization of the JPEG2000 coding variations.*

| | *0.25 bps* | *0.5 bps* | *1 bps* | *2 bps* | *max (55 dB)* |
|---|---|---|---|---|---|
| *normal* | 23.73 | 27.36 | 32.60 | 39.59 | 5.07 bps |
| *RESET* | 23.68 | 27.30 | 32.51 | 39.48 | 5.10 bps |
| *RESTART* | 23.64 | 27.26 | 32.47 | 39.43 | 5.12 bps |
| *CAUSAL* | 23.71 | 27.32 | 32.54 | 39.54 | 5.08 bps |
| *BYPASS* | 23.72 | 27.34 | 32.53 | 39.43 | 5.08 bps |
| *ERTERM* | 23.73 | 27.36 | 32.60 | 39.59 | 5.07 bps |
| *SEGMARK* | 23.71 | 27.34 | 32.57 | 39.56 | 5.08 bps |

LEGEND: *Each cell is the PSNR (in dB). The last column shows the bit-rate of the code-stream when the complete encoding is performed.*
RESULTS: *Cafeteria image of the corpus ISO 12640-1, gray scaled, size 2048x2560.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, expounded quantization, 64x64 code-blocks, no precincts, 1 quality layer. Lossy compression. Targeted bit-rates.*
IMPLEMENTATION: *Kakadu v4.5*

location. To allow the encapsulation of the whole code-stream of one code-block, precinct $P$ could have different packets, referred to as $\mathcal{T}_P^l$, with $0 \leq s \leq l$.

This organization allows the identification of layers, defined as the collection of all packets $\mathcal{T}_P^l$ with equal $l$ in that tile-part. When the code-stream segments are selected appropriately, each layer increases successively the quality of the image, and then they are usually called quality layers. Note that if we sort packets $\mathcal{T}_P^l$ in the increasing order $l = [0, 1, 2, ...]$, the first packets with $l = 0$ enclose the first layer, $l = 1$ the second one, and so on, generating a quality progressive code-stream. The packets in each layer can also be ordered in some specific way, repeating the same operation successively. The recursively way in which the packets have been ordered within the tile-part is called the progression order, and JPEG2000 defines 5 different types, briefly explained in Table 2.7. Figure 2.7 depicts the whole code-stream organization.

The selection of the code-stream segments, the number of quality layers and the bit-rate and/or quality of each layer is delegated to the rate control.

Table 2.7: *Progression orders defined by JPEG2000.*

| |
|---|
| • *LRCP*: code-stream primarily progressive by quality. In each quality layer, the lowest resolution level of all components is transmitted first, followed by the second resolution level of all components, and so on. |
| • *RLCP*: transmission of each resolution level progressively. In each resolution level the quality of the all components is increased progressively. |
| • *RPCL*: also transmission of each resolution level progressively, but the order in each resolution is by position, i.e. the first spatial location defined by precincts is fully transmitted first for all components, then the second spatial location, etc. |
| • *PCRL*: code-stream primarily progressive by position, meaning that the first location defined by the precincts is fully transmitted first, then the second one, etc. |
| • *CPRL*: transmission of each image component progressively. In each component the progression is by position. |

LEGEND: *Each progression order is defined as $X_1X_2X_3X_4$ with $X_n = \{L, R, C, P\}$ being Layer, Resolution, Component or Position respectively. $X_1$ defines the first sorting directive (or primary progression), $X_2$ the second one and so on.*

In order to decode correctly the code-stream, each packet must contain, for each code-block, the number of the included coding passes, the segment bit-rate and, when it is the first apparition, the number of bit-planes $K_i$. This information is included in a packet header and, since a code-stream may contain several quality layers with several packets each one, a coding process based on tag trees is used. The tag tree coding uses the redundancy among code-blocks belonging to the same precinct to reduce the amount of data that has to be transmitted. The code-stream has other headers containing information about image dimensions, number of components, etc., and they can be combined in different ways in order to provide more flexibility.

The organization of the code-stream in containers allows wide possibilities of random access and re-organization. First, tile-streams provide access to large image areas and packets to smallest spatial locations within tiles, supplying an efficient mechanism for interactively transmit large images and allowing the modification of image dimensions. Second,

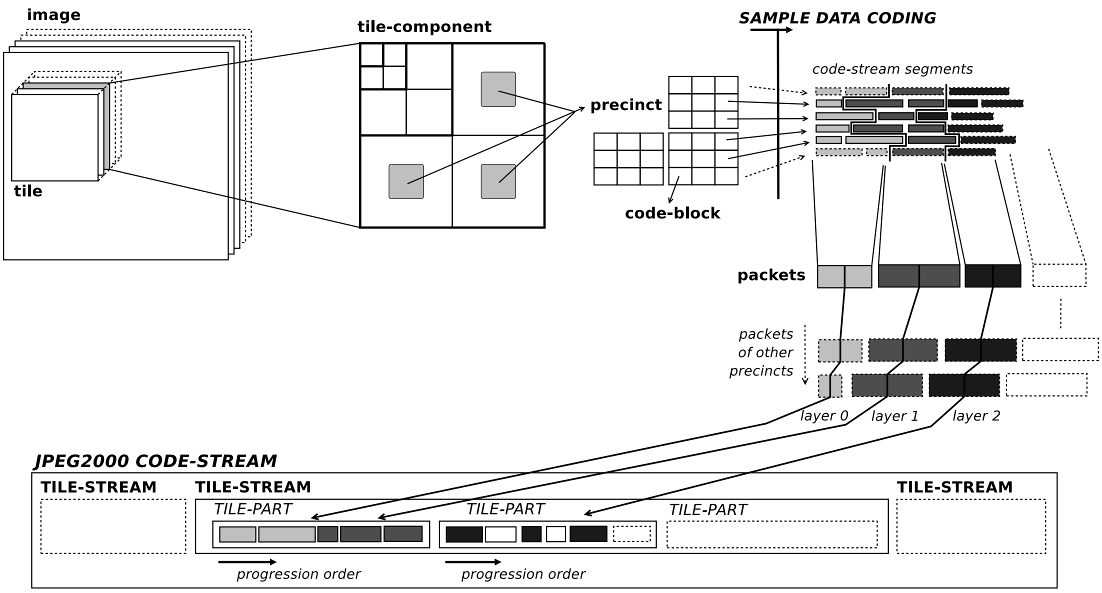



Figure 2.7: *JPEG2000 code-stream organization.*

since each tile-part may use a different progression order, the construction of the final code-stream could use different progression orders at bit-rate ranges. Third, a quality progression among all image tiles is also possible thanks to the identification of quality layers within each tile-part. When these tile-part quality layers are re-organized appropriately, a tile quality progression could be available. Fourth, a complete re-organization of the code-stream, involving precinct dimensions changes, progression orders or even quality of different image location, could be done without decoding any code-block.

It is worth noting that the code-stream organization could allow many other possibilities that applications could exploit. However, the re-organization can not consider the modification of tile dimensions and the modification of quality layers. Although tiles are usually not recommended, quality layers are fundamental.

The complete details of the code-stream organization and syntax are described in the standard [42] and in [38]. Besides, a channel protection technique using the organization and syntax of the code-stream is presented in [78].

### 2.2.6 Decoding procedures

The decoding stages of JPEG2000 are almost the same as the encoding ones. The main difference between coder and decoder is that the coder needs a rate control method. However, there are some points that the decoder should take into account, summarized in the following items:

- *Beyond code-words lengths*: as explained in Section 2.2.4, when the MQ-decoder exhausts the code-word, readings of $\mathrm{FF}_h$ bytes must be simulated in order to finish the decoding.

- *Bias reconstruction parameter*: the decoding of one coefficient bit corresponding to its first significant bit-plane, referred to as $p$, indicates that the coefficient value lies in the range $[2^p, 2^{p+1})$. One might expect that the best approximation of the value should be the center of this range, i.e. $2^p + 2^{p-1}$, and successively for its magnitude refinement bits. This is the same as to set the reconstruction parameter $\gamma = \frac{1}{2}$ in the dequantization process, which reconstructs the decoded coefficients $q^p[n]$ at bit-plane $p$ as

$$y'[n] = \begin{cases} (|q^p[n]| + \gamma) \cdot sign(q^p[n]) \cdot 2^p \triangle_b & \text{if } q^p[n] \neq 0 \\ 0 & \text{if } q^p[n] = 0 \end{cases} .$$

  However, experience indicates that improved results are achieved when the reconstruction parameter is biased towards 0; $\gamma = \frac{3}{8}$ is a good choice. Obviously the optimal value could be calculated on the statistics of each subband, but the decoder usually does not have access to these statistics. Note that when using lossless compression this operation is not needed since the quantization operation is not performed.

- *Range check*: if the dequantization process uses the center range approximation, it could produce some special combinations for which, when inverting the DWT, the resulting samples violate the bit ranges of the original image. The decoder should take into account these exceptions correcting the coefficients when necessary.

- *Rounding before saving*: the decoding pipeline outputs an image with a floating point representation (in lossy mode). The conversion to an integer representation in order to save or display the image should be preceded by a rounding operation.

An interesting method which improves the quality of the image at the decoder by exploiting quantization and geometric information is explained in [54], although the computational complexity of the decoder is increased. On the other hand, a study on how to improve the visual quality using mechanisms supported by JPEG2000 (requiring some special parametrization of both coder and decoder) is presented in [117].

## 2.3 Analysis

### 2.3.1 Computational complexity

The computational complexity is an important issue for JPEG2000. Applications are to be able to encode or decode images rapidly and, in some cases, this could become a decisive issue when deciding whether to use the JPEG2000 standard or not. Actually, in the last meetings selecting the core coding system, SPECK coding system [71] competed against EBCOT because, although SPECK has worse coding performance than EBCOT, it is more efficient computationally. The EBCOT paradigm supplies a tremendous flexibility and a high coding performance but it has a high computational complexity compared to other coders due to the use of three coding passes for each bit-plane plus the use of the arithmetic coder. To analyze the computational complexity is important in order to optimize some of the stages of the core coding system.

When studying the JPEG2000 coding pipeline, it is clear that the sample data transformation stage is quite simple and would imply almost the same operations for any wavelet-based coding system and, therefore, the same computational complexity whatever the bit plane coder was. Besides, the code-stream re-organization is even less complex, implying just a few operations for the packet headers construction. Obviously the sample data coding is the most complex stage overall the coding process, whereas the rate control is not complex and it could help to diminish the computational complexity of the coding stage as showed in Sections 3.1.4, 6.4.5.

In order to evaluate the computational complexity of JPEG2000, we present a simple comparison among the different stages of the coding pipeline. It is worth noting that this computational complexity evaluation is a difficult task since each implementation may use different methods and optimization techniques. In our evaluation we use BOI, which implements a few optimization techniques. In general, the obtained results coincide with other studies presented in the literature, such as [95] and [21]. Table 2.8 shows the computational load of the different stages of the JPEG2000 coding pipeline when encoding several images using lossy compression.

For lossless compression, the computational complexity of the DWT is slightly less than for lossy compression, the quantization stage is not applied, and the computational complexity of the remaining stages is practically the same for both compression types. On the other hand, notice that no rate control stage percentage is provided in Table 2.8 because BOI uses the rate control method introduced in Chapter 6, which has a negligible computational complexity. Usually, the rate control methods used in JPEG2000 implementations takes about 5% to 10% of the whole coding process.

Notice that more than 70% of the encoding time is used by the sample data coding stage. It is for this reason that several optimization techniques for this stage (both BPE and MQ) are described in [100, Chapter 17] and implemented in Kakadu. Using all the described techniques, the time of the sample data coding can be widely reduced, although it is still the stage that takes longer. The DWT uses about 15% of the encoding time and is difficult to optimize because the lifting scheme already minimizes its computational complexity. The results obtained for the decoder are practically the same, thus they are not given.

JPEG2000 allows many different options of parallelization. The results presented here are obtained using a single processing unit, but the coding process may be parallelized in order to increase the computational performance of JPEG2000 applications. Independent tile processing is obviously a rather simple parallel strategy, although tile coding is not

Table 2.8: *Computational load evaluation of the JPEG2000 coding stages.*

| | *Sample Data Transformations* | | | *Sample Data Coding* | | *Code-stream* |
|---|---|---|---|---|---|---|
| *image* | PRE | DWT | Q | BPE | MQ | *re-organization* |
| Portrait | 3.17 % | 18.09% | 4.69% | 55.41% | 13.17% | 5.46% |
| Cafeteria | 2.56 % | 13.19% | 3.77% | 61.99% | 13.43% | 5.05% |
| Fruit | 3.19 % | 14.59% | 4.70% | 58.44% | 13.86% | 5.20% |
| Tableware | 3.21 % | 14.72% | 4.74% | 58.58% | 13.48% | 5.25% |
| Bicycle | 3.11 % | 15.67% | 4.61% | 57.84% | 13.09% | 5.67% |
| Orchid | 3.67 % | 16.79% | 5.43% | 55.90% | 12.46% | 5.75% |
| Musicians | 2.29 % | 10.44% | 3.40% | 64.09% | 14.93% | 4.85% |
| Candle | 2.55 % | 11.66% | 3.77% | 63.11% | 14.15% | 4.74% |
| *average* | 2.55 % | 14.07% | 4.30% | 59.85% | 13.66% | 5.20% |
| *stages av.* | 20.92% | | | 73.51% | | 5.20% |

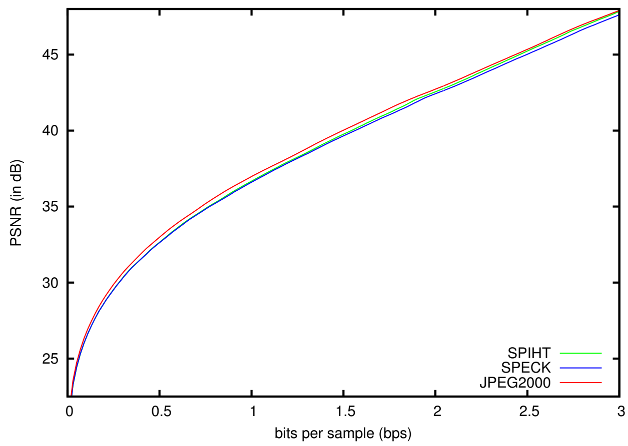
LEGEND: *Each column shows a coding stage in terms of percentage over the whole encoding process. PRE groups the level offset and colour transform operations, DWT stands for the discrete wavelet transform operation using the lifting scheme, Q stands for quantization, BPE represents the bit plane encoding and MQ stands for the MQ-coder.*
RESULTS: *images of the corpus ISO 12640-1, colour images, size 2048x2560.*
CODING PARAMETERS: *ICT, 9/7 DWT 5 levels, derived quantization, RESTART coding var., 64x64 code-blocks, no precincts, 1 quality layer, LRCP progression. Lossy compression.*
IMPLEMENTATION: *BOI v1.2*

recommended. Other options of parallelization may be the independent code-block coding or the parallelization within the code-block coding through the variations described in Table 2.5. Besides, some hardware-based coding architectures using smart parallelization techniques have recently appeared [31, 118, 70, 37].

### 2.3.2 Coding performance

Compared to other coding systems, JPEG2000 has state-of-the-art coding performance. However, it is difficult to perform a fair comparison among different coding systems only considering the coding performance, because each coding system is devised to provide different features. Here, we only perform a brief comparison of JPEG2000 against two popular image coding systems in order to show that JPEG2000 has a good coding performance while supplying advanced features. A more extensive comparison among different standards and coding systems is provided in [81], considering several issues.

We have compared JPEG2000 against SPIHT [77] and SPECK [71]. SPIHT is a common reference used in image coding. Graphic 2.1 contains the average results obtained when applying a lossy compression to the images of the corpus ISO 12640-1, depicted in both a graphic and a table to better appreciate the differences. JPEG2000 is mainly devised for lossy compression, and although it allows a lossless mode, the results it obtains are not as competitive as a coding system devised for lossless compression, for instance the standard JPEG-LS [41]. A comparison of lossless compression among JPEG2000 and other coding systems is provided in [88, 81].




| *bps* | *SPIHT* | *SPECK* | *JPEG2000* |
|---|---|---|---|
| 0.0625 | 25.00 | 25.03 | 25.31 |
| 0.125 | 27.04 | 27.04 | 27.34 |
| 0.25 | 29.55 | 29.53 | 29.87 |
| 0.5 | 32.65 | 32.63 | 32.98 |
| 1 | 36.67 | 36.60 | 36.97 |
| 2 | 42.55 | 42.42 | 42.73 |
| 3.5 | 50.71 | 50.40 | 50.32 |

LEGEND: *The coding performance is expressed in dB.*

RESULTS: *average of corpus ISO 12640-1, gray scaled, size 2048x2560. 200 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, expounded quantization, 64x64 code-blocks, no precincts, 1 quality layer, LRCP progression. Lossy compression. Targeted bit-rates.*
SPIHT CODING PARAMS.: *9/7 DWT 5 levels, arithmetic coding.*
SPECK CODING PARAMS.: *9/7 DWT 5 levels.*
IMPLEMENTATIONS: *JPEG2000 - Kakadu v4.5, SPIHT/SPECK - QccPack v0.51 [33]*

Graphic 2.1: *Coding performance evaluation of JPEG2000, SPIHT and SPECK.*

# Chapter 3

# From embedded code-streams to quality layers

This chapter provides an in-depth study of the rate control within the JPEG2000 core coding system. The first section is focused on the rate control methods found in the literature, beginning with the optimal Post Compression Rate-Distortion optimization method proposed in EBCOT and reviewing all the approaches until August 2006. We supply both an overview of each method and a comparison among them. The issue of the second section is the quality scalability of JPEG2000 code-streams, reviewing how quality layers are allocated within the code-stream and which is the coding performance obtained with usual allocation strategies.

## 3.1 Rate control methods

### 3.1.1 Introduction

The first coding systems based on the DWT, such as [83, 77, 102], constructed a single quality embedded code-stream. This means that, if $D^j$ stands for the distortion of the recovered image at bit-rate $R^j$, the distortion $D^{j+1}$ at any bit-rate $R^{j+1}$ satisfying $R^j <$

$R^{j+1}$ is less than at $D^j$, i.e. the condition $D^j > D^{j+1}$ is always respected. Recall that a quality embedded code-stream also means that the segment of the code-stream at bit-rate $R^j$ is a prefix of the code-stream at bit-rate $R^{j+1}$. In those coders, this property was given by the bit plane coder itself, since the code-stream was constructed considering the whole image scanning all the coefficients of the image in a bit-plane by bit-plane fashion. The main drawback of these coding systems was that neither spatial nor resolution scalability were supported.

An efficient way to support both spatial and resolution scalability is by considering the independent coding of code-blocks, like EBCOT [93] does. Recall that the sample data coding stage constructs an embedded code-stream for each code-block. When a target bit-rate, say $R^{max}$, has to be attained, the EBCOT paradigm raises the question of how to select the best code-stream segments of each code-block which best recover the image. Besides, the complementary problem of how to achieve the minimum bit-rate for a target quality, say $D^{max}$, also needs to be addressed. These issues compel to develop a method able to control the bit-rate, or distortion, of the constructed code-streams.

When the goal is to optimize the quality at the desired bit-rate $R^{max}$, the rate control method searches the set of coding segments that minimizes the overall image distortion satisfying $R \leq R^{max}$. Conversely, when the goal is to optimize the bit-rate for a desired distortion, say $D^{max}$, the rate control method searches the set of coding segments which minimizes the overall code-stream bit-rate satisfying $D \leq D^{max}$.

The JPEG2000 standard does not supply a specific rate control method. Instead, it defines a flexible code-stream organization supporting quality layers, and describes a possible rate control method as a guideline. Therefore, implementations can use their own rate control methods to optimize the code-stream construction for a target bit-rate or for a target distortion. This is probably the cause of the apparition of more than twenty different rate control methods since 2002; they are reviewed in Section 3.1.3. The guideline proposed in the standard is the Post Compression Rate-Distortion Optimization (PCRD or PCRD-opt) method described in EBCOT, which obtains optimal (or near-optimal) results and is the main reference of rate control in JPEG2000.

### 3.1.2 The optimal PCRD method

Let us introduce the rate control issue as the optimization of the quality for a desired bit-rate, as is formulated in EBCOT. The optimization of the bit-rate for a desired quality and the construction of quality layers are complementary problems, which are not considered here. Before starting the sample data coding stage of JPEG2000, each subband is partitioned in code-blocks. Each code-block $\mathcal{B}_i$ can be truncated at the end of each coding pass, giving $N$ different truncation points referred to as $n_j, 0 \leq j < N$ and referring the bit-rate of each truncation point of the code-block $\mathcal{B}_i$ as $R_i^{n_j}, R_i^{n_j} \leq R_i^{n_{j+1}}$. When the distortion is measured in terms of MSE, the distortion of the code-block $\mathcal{B}_i$ at the truncation point $n_j$ is referred to as $D_i^{n_j}$ and may be calculated as

$$D_i^{n_j} = w_{b_i}^2 \sum_{k \in \mathcal{B}_i} (y[k] - \hat{y}^{n_j}[k])^2$$

where $\hat{y}^{n_j}[k]$ denotes the coefficients quantized at the truncation point $n_j$, $y[k]$ represents the original coefficients of code-block $\mathcal{B}_i$ and $w_{b_i}$ stands for the L2norm of the subband $b_i$ where the code-block $\mathcal{B}_i$ belongs.

Considering that the total distortion of the image is an additive metric, calculated as $D = \sum_i D_i^{n_j}$, and that the total bit-rate of the code-stream is given by $R = \sum_i R_i^{n_j}$, we can approach the rate-distortion optimization problem as a generalized Lagrange multiplier for a discrete set of points [30] as follows: $\{n_j^\lambda\}$ stands for the set of truncation points which minimizes

$$(D(\lambda) + \lambda R(\lambda)) = \sum_i (D_i^{n_j^\lambda} + \lambda R_i^{n_j^\lambda})$$

where $\lambda$ is the Lagrange multiplier. That value of $\lambda$ which minimizes this expression yielding $R(\lambda) = R^{max}$ represents the optimal solution to the rate-distortion optimization problem, being $\{n_j^\lambda\}$ the set of optimal truncation points for that $R^{max}$.

Note that this rate-distortion optimization method needs to collect some information during the encoding of each coding pass. In particular, it needs the distortion and bit-rate of every truncation point of every code-block, referred to as $D_i^{n_j}$ and $R_i^{n_j}$ respectively, in order to calculate the rate-distortion slope, given by $S_i^{n_j} = \triangle D_i^{n_j} / \triangle R_i^{n_j}$ where $\triangle D_i^{n_j} = D_i^{n_{j-1}} - D_i^{n_j}$ and $\triangle R_i^{n_j} = R_i^{n_j} - R_i^{n_{j-1}}$. The achievement of the optimal solution through the Lagrange multiplier compels to select only those truncation points that lie in the *convex hull*, i.e. those truncation points with strictly decreasing rate-distortion slope. The truncation point $n_{j+1}$ with $S_i^{n_{j+1}} \leq S_i^{n_j}$ is not a *feasible truncation point* and must never be selected. Figure 3.1 depicts the feasible truncation points of one code-block lying in the convex hull.

The search of the optimal truncation points can be calculated as follows: if $\mathcal{N}_j$ denotes the complete set of feasible truncation points and $j_k$ is the enumeration of these feasible truncation points with rate-distortion slopes $S^{j_k}$ satisfying $S^{j_k} < S^{j_{k+1}}$, the determination of $\{n_j^\lambda\}$ is simplified to $n_j^\lambda = max\{j_k \in \mathcal{N}_j | S^{j_k} > \lambda\}$ for some value of $\lambda$.

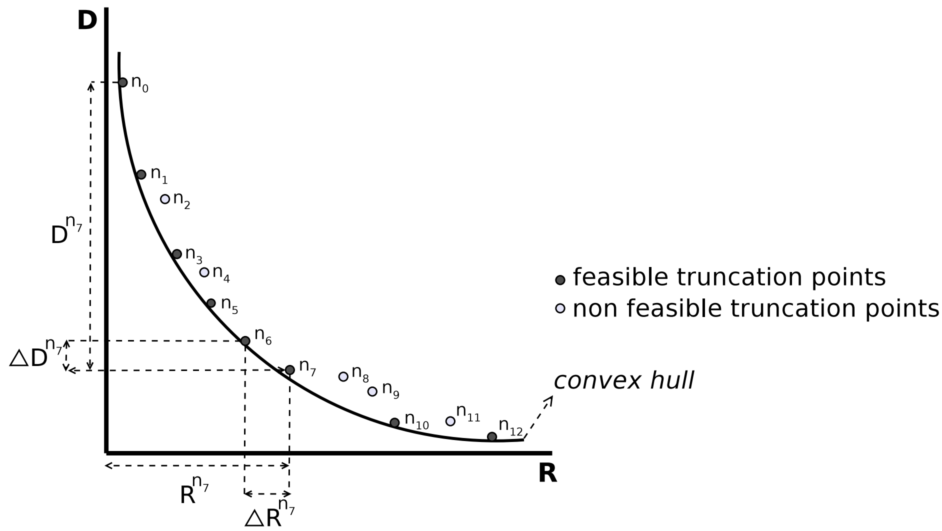



Figure 3.1: *The convex hull.*

Although this method gives the optimal solution for $R(\lambda) = R^{max}$, when $R(\lambda) \neq R^{max}$ the set of points $\{n_j^\lambda\}$ might be non optimal. However it is still a good approximation and it is usually given as the optimal solution to the optimization problem. It is worth noting that this approach benefits from the fact that the sample data coding of each code-block produces a dense set of feasible truncation points[1] that can approximate fairly well the optimal solution. Besides, implementations should also take into account that the packet headers and main headers fill some bytes. The length of the latter can be easily estimated, but the length of the former can only be calculated once $\{n_j^\lambda\}$ is determined. However, this difficulty can be overcome by experimental estimations and/or recalculating $\{n_j^\lambda\}$ if the bit-rate is widely exceeded.

Summarizing, the PCRD method is applied in two steps: the first one uses the distortion contribution and the bit-rate of each coding pass to calculate the set of feasible truncation points of code-blocks. The second step uses the generalized Lagrange multiplier to estimate the optimal set of truncation points for a target bit-rate. This process approximates the optimal solution and has a low computational complexity. However, the original formulation of the PCRD method described in EBCOT [93] applies the Lagrange multiplier once the complete image is encoded, i.e. it compels to encode all coding passes of all code-blocks, even if the target bit-rate is so small that, in the final code-stream, only few coding passes are included. This is the only drawback of the optimal PCRD method, which in some cases might become meaningful. Taubman himself proposed a solution in [95], which is discussed in the following section. Almost all the new rate control methods, including the methods we propose in this thesis, also optimize this issue.

### 3.1.3 Other rate control methods

We have found twenty-four rate control methods related to JPEG2000 in the literature, published in the last five years. It is not easy to classify and assess the performance of all of them because each one is focused on different parts of the optimization problem, and it is not easy to set global categories which define them perfectly. Besides, their experimental results are performed using different image corpora, different JPEG2000 implementations and even different coding parameters. To perform a comparison among them is undoubtedly not easy. However, we consider this classification and comparison absolutely necessary in

[1] In Section 4.2 this fact is corroborated with the experimental result of Table 4.1 –page 90–.

order to review the state-of-the-art and to assess, in some way, the contribution of each one.

Fourteen of the twenty-four rate control methods are directly related with our research, and although the other ten methods are also related to rate control, their purposes are far away from ours and their results can not be compared with the results obtained in this thesis. The fourteen methods directly related with our research are classified in three categories, which roughly describe the type of technique used to perform the rate control. In some cases, a method could be classified in more than one category. When possible, the remaining ten methods not directly related with this research are classified in these categories; since this is not always possible, three new categories are proposed. The following items summarize all these categories:

- Directly related with our research
  - *Deterministic*: methods that use distortion measures based on the original image. These measures are used by a PCRD like method to asses the rate-distortion of each coding pass of code-blocks.
  - *Model-based*: methods based on some theoretical model that characterizes the rate-distortion contributions of the truncation points. The model is used before or during the sample data coding stage to predict the coding passes that are likely to be included in the final code-stream.
  - *Quantized*: methods based on the calculation of adequate step sizes for each subband in order to attain a target bit-rate.
- Not directly related with our research
  - *Tiling*: methods that address the bit-rate optimization problem for the use of tiles.
  - *Source-channel*: methods focused on the source-channel encoding and rate control.

– *Opt-Lagrange*: methods focused on the search of the set of optimal truncation points using approaches different from the generalized Lagrange multiplier.

The purpose of this classification is to give an approximation of how the rate control is performed in each method. In addition, we also consider some features of the methods that point out the most important advantages and disadvantages of each one. The explanation of the considered features is given in Table 3.1.

Table 3.1: *Features considered for the evaluation of each rate control method.*

| |
|---|
| • **LAYERS CONSTRUCTION**: *yes* if the method allows the construction of quality layers implicitly; *no* otherwise. If the rate control method does not allow the construction of quality layers, implementations need to allow their construction in some other way. |
| • **ANY DISTORTION MEASURE**: *yes* if the method can optimize the distortion using a different measure than the MSE; *no* otherwise. |
| • **INCREMENTAL ENCODING**: *yes* when the encoding of each code-block is not performed at once, meaning that the whole image, as well as some information related to the MQ-coder, needs to be maintained in memory; *no* otherwise. |
| • **DISTORTION MEASURES BASED ON THE ORIGINAL IMAGE**: *yes* if the method uses distortion measures based on the original image; *no* otherwise. When the distortion measures might be extracted from the code-stream, meaning that the original image is indeed not needed, we write *dec* in this feature. |
| • **USE IN THE DECODER**: *yes* if the method could also be used to control the rate-distortion once the code-stream is already encoded without needing to decode it; *no* otherwise. See Section 4.1 –page 83– for a further explanation of this feature. |
| • **QUALITY/BIT-RATE OPTIMIZATION**: *qual* if the method only allows the optimization of the quality for a target bit-rate, *bit* when it only allows the optimization of the bit-rate for a target quality and *both* when it allows both optimizations. |
| • **COMPUTATIONAL COMPLEXITY REDUCTION**: *yes* if the method does not compel to encode all the coding passes of all code-blocks; *no* otherwise. |

We have considered two types of experimental results for the fourteen methods directly related with this research: the coding performance and the speed-up. The coding performance is considered in terms of MSE since it is the most common distortion measure used in the literature. We denote, for instance, 0.1 dB worse than the optimal PCRD method just as 0.1. The speed-up is calculated as follows: $T_{\text{PCRD}}/T_{\text{method}}$ where $T_{\text{PCRD}}$ means the time taken by the sample data coding stage when encoding the full image (i.e., all coding passes of all code-blocks are encoded), like the PCRD does, and $T_{\text{method}}$ is the time that the proposed method takes by the sample data coding stage. In some cases, the authors do not specify the time but the number of encoded symbols, although both types of results have been considered equivalent. The coding parameters, image corpus, implementation and type of results (extracted from a graphic or from a table) are also specified.

A classification of JPEG2000 rate control methods similar to the one presented in this section is found in [116], distinguishing between content-adaptive and model-based rate control methods, where content-adaptive means that the rate control methods use distortion measures based on the original image (equivalent to our deterministic category) and model-based means the same as our category. In [103] there is a comparison of the coding performance and the speed-up among some of the rate control methods reviewed here.

When the authors do not name the proposed rate control method, we use the last name of the first author to identify it. When a method is described in more than one paper, we emphasize in bold font the reference that has been used for the results extraction and the summary.

| | |
|---|---|
| **1. NAME:** | Post Compression Rate-Distortion optimization (PCRD) |
| **PAPER/S:** | [92] October 1999, **[93] July 2000** |
| **CLASS.:** | deterministic |

*EXPLANATION*

This method is explained in Section 3.1.2.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | no | | |

**RESULTS**

Sub-optimal results in terms of coding performance (considered optimal).

The remaining rate control methods are compared to PCRD (in terms of coding performance and computational complexity).

| | |
|---|---|
| **2. NAME:** | M-SWRC, M-EWRC |
| **PAPER/S:** | [32] December 2000, **[51] January 2005** |
| **CLASS.:** | deterministic |

*EXPLANATION*

The main contribution of the these rate control methods is the achievement of uniform quality within all the components of an image. The methods address the optimization problem from the point of view of constrained resource devices, in particular from the on-board processing of hyper-spectral images. The proposed coding system carries out the encoding process to the so-called scan elements independently. Rather than using tiles, the scan elements are made up with an specified number of image lines. All the stages of the JPEG2000 core coding system are performed to each scan element, including the wavelet transform, that uses a line-based processing. Once the scan element is encoded, it is added to a rate control buffer and, after the number of specified scan elements are encoded, this buffer sends out a quality layer using a smart rate allocation strategy based on the rate-distortion slope of each scan element. The proposed coding system has a high flexibility, allowing the specification of the scan element size, the buffer size, the number of quality layers, the desired rate-distortion slope for each quality layer, etc. The main difference between both methods is that M-EWRC enhances the rate allocation strategy using an extra buffer. This rate control method has also been applied in motion JPEG2000 [24].

| | | | |
|---|---|---|---|
| **FEATURES** | | | |
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |
| **NO RELATED RESULTS** | | | |

| | |
|---|---|
| **3. NAME:** | Masuzaki |
| **PAPER/S:** | **[65] May 2002**, [64] June 2002 |
| **CLASS.:** | model-based |

*EXPLANATION*

This method groups the subbands provided by the Mallat decomposition in two subband groups for each resolution level, referred to as $G_{k,0} = \{HL_k, LH_k\}$ and $G_{k,1} = \{HH_k\}$ ($k$ denotes the decomposition level), whereas subband $LL$ is considered as a separate group. Experimentally, the authors relate the number of coding passes included in each one of these groups with the PSNR obtained at different target bit-rates. They conclude that the relation can be roughly expressed using a linear equation, which is used to predict the number of coding passes for each subband group at a specified bit-rate before starting the encoding process. Besides, while the encoding process is performed, the proposed algorithm updates the estimation in order to better estimate the number of needed coding passes for each subband group.

| | | | |
|---|---|---|---|
| **FEATURES** | | | |
| *LAYERS CONSTRUCTION:* | no | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | no |
| *USE IN THE DECODER:* | yes | *QUALITY/BIT-RATE OPT.:* | qual |
| *COMP. COMP. REDUCTION:* | yes | | |
| **RESULTS** | | | |
| *IMPLEMENTATION:* | VM 9.1 | *RESULTS:* | graphic |
| *CORPUS:* | not specified | | |
| *CODING PARAMETERS:* | not specified (assumed lossy compression) | | |
| **CODING PERFORMANCE** | | | |
| | for all bit-rates $\sim$1 | | |
| **SPEED-UP (average)** | | | |
| | 0.5 bps - 3.4 | | |

| | |
|---|---|
| **4. NAME:** | Minimal Slope Discarding (MSD) |
| **PAPER/S:** | **[17] June 2002**, [18] May 2003 |
| **CLASS.:** | deterministic |
| *EXPLANATION* | |

The main idea behind the proposed method is to relate the rate-distortion slope of coding passes with the bit-rate. This relation is calculated jointly in the encoding process using a table with $2^J$ entries, each one representing a rate-distortion slope value. When the encoding of each coding pass is performed, the number of bytes for each rate-distortion slope is accumulated in the corresponding entry of the table. Then, for each code-block, the encoding is stopped when the accumulated bytes of the current rate-distortion slope is higher than the target bit-rate. Although the first code-blocks need to be encoded completely, the more code-blocks are encoded, the less number of coding passes are needed to encode for the remaining code-blocks and, therefore, the computational complexity is reduced. The main drawback of this method is that the target bit-rate cannot be precisely attained without using large tables, which is not recommended by the authors.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |

**RESULTS**

| | | | |
|---|---|---|---|
| *IMPLEMENTATION:* | VM 7.2 | *RESULTS:* | table |
| *CORPUS:* | Lena, Pepper, Jet, Baboon | | |
| *CODING PARAMETERS:* | 9/7 DWT 5 levels. Lossy compression. Targeted bit-rates. | | |

**CODING PERFORMANCE (worst / best / average)**

| | |
|---|---|
| 0.0625 bps - 0.017 / -0.001 / 0.0 | 0.5 bps - 0.035 / 0.0 / 0.0 |
| 0.125 bps - 0.11 / 0.006 / 0.03 | 1 bps - 0.035 / 0.0 / 0.0 |
| 0.25 bps - 0.016 / -0.019 / 0.0 | |

**SPEED-UP (min / max / average)**

| | |
|---|---|
| 0.0625 bps - 5.8 / 8.6 / 7.5 | 0.5 bps - 1.7 / 2.7 / 1.92 |
| 0.125 bps - 4 / 6.6 / 5.12 | 1 bps - 1.17 / 1.6 / 1.27 |
| 0.25 bps - 2.4 / 4.3 / 3.08 | |

**5. NAME:** Long

**PAPER/S:** [59] June 2002, **[91] June 2002**

**CLASS.:** quantized + deterministic

*EXPLANATION*

The authors discuss how to improve the lossy coding performance when using the 5/3 filter-bank by proposing a slightly modified measure based on the L2norm, which increases the weight of the lower subbands. Since lossless compression can not use floating point processing, weights are rounded to the nearest integer, in the same way proposed in [14]. When this method is compared to the use of the L2norm to weight each subband, improvements of more than 1 dB are achieved. The authors also claim that the method could be extended to the 9/7 filter-bank in order to improve the coding performance of JPEG2000. However, the calculations of the modified L2norm are the same as in [76]. Since this method just modifies the subband weight, the PCRD must also be applied.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | no | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | no |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | qual |
| *COMP. COMP. REDUCTION:* | no | | |

**NO RELATED RESULTS**

**6. NAME:** Kakadu

**PAPER/S:** **[95] July 2002**

**CLASS.:** deterministic + model-based

*EXPLANATION*

The proposed method uses the PCRD approach to achieve the optimal results (in terms of MSE or the desired quality measure) but, instead of encoding all the coding passes of all code-blocks, a prediction that only encodes those coding passes that are likely to be included in the final code-stream is used. The method has two steps: the first one collects statistics related with the rate-distortion slope from the already encoded code-blocks, and the second step is executed at the end of each coding pass in order to decide whether it is necessary to continue encoding or not, using the statistical information of the first step. The statistical information and the way in which the prediction is performed are not explained, although it is said that a conservative strategy is used to achieve optimal results. Details of the method can be found in the source code of Kakadu.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |

**RESULTS**

| | | | |
|---|---|---|---|
| *IMPLEMENTATION:* | Kakadu | *RESULTS:* | table |
| *CORPUS:* | Bicycle | | |

*CODING PARAMETERS:* not specified (assumed lossy compression)

**CODING PERFORMANCE**

optimal results

**SPEED-UP (average)**

| | |
|---|---|
| 0.25 bps - 5.3 | 1 bps - 2.38 |
| 0.5 bps - 3.63 | 2 bps - 1.64 |

**7. NAME:** Battiato

**PAPER/S:** **[13] August 2002**

**CLASS.:** tiling

*EXPLANATION*

The proposed rate control method is devised for the optimization of the image quality when encoding using tiles. The method performs two steps: the first one classify each tile depending on its features. In this step, two edge-detection filters are used in order to classify the tile in one of four possible categories (plain, edge, texture or unknown). The second step assigns a number of credits to each tile depending on its classification and on simple energy measures, in order to calculate its weight. The tile weights are used to determine a bit-rate for each tile, achieved with the PCRD method. The proposed method is compared to the calculation of the tile bit-rates just dividing the overall bit-rate by the number of tiles, although only visual results are reported.

**NO RELATED RESULTS**

**8. NAME:** Parisot

**PAPER/S:** [69] September 2002, **[67] September 2002**, [68] September 2002

**CLASS.:** quantized

*EXPLANATION*

The proposed rate control method is based on the estimation of an optimal step size for each subband, which can be determined depending on a target bit-rate or on a target

MSE. The estimation is based on a theoretical model and on statistical information and is performed before starting the encoding. Once the step size is set, the quantization and encoding process are performed straightforward, without needing any post compression rate-distortion stage at the end of the encoding process. It is worth noting that the proposed method is not tested within the JPEG2000 framework, but using the bit plane coder defined in EBCOT.

| | | | |
|---|---|---|---|
| **FEATURES** | | | |
| *LAYERS CONSTRUCTION:* | no | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |
| **RESULTS** | | | |
| *IMPLEMENTATION:* | own, VM 8.6 | *RESULTS:* | table |
| *CORPUS:* | Lena, Goldhill, Hotel | | |
| *CODING PARAMETERS:* | 9/7 DWT 3 levels. Lossy compression. Targeted bit-rates. | | |
| **CODING PERFORMANCE (worst / best / average)** | | | |
| 0.25 bps - 0.1 / -0.14 / 0.0 | 1 bps - 0.11 / -0.11 / 0.0 | | |
| 0.5 bps - 0.13 / -0.12 / -0.01 | 2 bps - 0.6 / 0.12 / 0.31 | | |
| **SPEED-UP** | | | |
| not specified | | | |

**9. NAME:** IREC, E-IREC

**PAPER/S:** [115] September 2002, [114] March 2004, **[116] May 2006**

**CLASS.:** deterministic

*EXPLANATION*

Two rate control methods are presented: the GHRaC and the IREC. The purpose of GHRaC is to optimize the computational complexity of the PCRD by replacing the use of the generalized Lagrange multiplier by a greedy marginal analysis. The presented greedy algorithm uses a d-heap structure in order to optimize the number of iterations needed to find the solution of the rate-distortion optimization problem. The main idea behind this method is to select one by one the best truncation points among all the code-blocks. IREC is based on two assumptions: the first one is that the GHRaC method only needs the immediately subsequent feasible truncation point of each code-block, but not all of them. The second assumption is based on statistical information and states that, in general, feasible truncation points only include one or, at most, two or three consecutive coding passes. Thanks to these assumptions, the IREC local-K method starts encoding $K$ coding passes for all the code-blocks and calculating their rate-distortion slope as well as the feasible truncation points; then GHRaC determines the next truncation point to be included in the final code-stream and, for that code-block, are encoded $K - j$ coding passes, where $j$ stands for the number of coding passes included in the current feasible truncation point. This process is repeated until the specified bit-rate is achieved. The IREC algorithm is further improved if, instead of coding $K$ coding passes for all code-blocks, the rate-distortion slope is estimated before starting the encoding. This modification is called E-IREC. Here, the analysis of these methods only considers the E-IREC approach, since it gives the best results.

| | | | |
|---|---|---|---|
| **FEATURES** | | | |
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | yes | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |
| **RESULTS** | | | |
| *IMPLEMENTATION:* | Jasper | *RESULTS:* | graphic |
| *CORPUS:* | Baboon, Barbara, Lena, Fruits | | |
| *CODING PARAMETERS:* | 9/7 DWT 4 levels, 32x32 code-blocks. Lossy compression. Targeted bit-rates. | | |
| **CODING PERFORMANCE** | | | |
| | optimal results | | |
| **SPEED-UP (min / max / average)** | | | |
| 0.0625 bps - 12 / 16 / 14 | 0.5 bps - 2 / 3 / 2.5 | | |
| 0.125 bps - 5 / 9 / 7.5 | 1 bps - 1 / 1.5 / 1.25 | | |
| 0.25 bps - 3 / 5 / 4 | 2 bps - 1 / 1 / 1 | | |

| **10, 11, 12. NAME:** SBRA, PSRA, PSOT | |
|---|---|
| **PAPER/S:** | [113] April 2003, [112] July 2003, [111] September 2003, **[110] March 2005** |
| **CLASS.:** | deterministic, model-based + deterministic, deterministic |
| *EXPLANATION* | |

*Successive Bit-plane Rate Allocation (SBRA)*

SBRA uses a slightly modified rate-distortion slope measure in order to determine which coding passes are likely to be included in the final code-stream. The main advantage of this method is that, thanks to the smart theoretical approach presented, the selection of the coding passes that have to be encoded is performed at the encoding time of each code-block independently and without needing to compare rate-distortion slopes among different code-blocks. This does not compel to maintain the whole image in memory when the encoding process is carried out. One feature of the method is that it uses the bit-plane boundaries to calculate rate-distortion slope measures and to determine if more coding passes are needed for that code-block. The main drawback of SBRA is that the target bit-rate can not be perfectly attained, although it is well approximated.

*Priority Scanning Rate Allocation (PSRA)*

The main assumption used in this method is that the probability of a coding pass to be included in the final code-stream is directly related with the bit-plane where it is situated, being the highest bit-planes those ones that contain the coding passes with a higher probability to be included in the final code-stream. The coding passes are encoded from the highest to the lowest bit-plane, encoding all the coding passes situated at the same level until the lowest bit-plane is reached or the target bit-rate is achieved. The rate-distortion slope is calculated at the end of each coding pass, and it is used at the last step of the algorithm to discard those code-blocks with the smallest rate-distortion slopes, attaining almost perfectly the target bit-rate.

*Priority Scanning with Optimal Truncation (PSOT)*

The main idea behind this method is the same as in PSRA but the goal here is to achieve the optimal solution. The method does exactly the same as PSRA except for the last step. PSRA stops encoding when the accumulated bit-rate is greater than the target bit-rate and then some coding passes are discarded, starting from that with the smallest rate-distortion slope value $S_{min}$, to better attain the target bit-rate. Instead of stop encoding, PSOT continues encoding after exceeding the target bit-rate in order to encode all the coding passes of all code-blocks until their coding passes have a rate-distortion slope equal or less than $S_{min}$. Then the generalized Lagrange multiplier is applied to select the optimal set of truncation points. Note that although pathological cases may cause different solutions for PCRD and PSOT, this method usually achieves the same solution as PCRD.

| | | | |
|---|---|---|---|
| **RESULTS** | | | |
| *IMPLEMENTATION:* | Jasper | *RESULTS:* | graphic |
| *CORPUS:* | Lena, Barbara, Goldhill, Boat, Mandrill, Peppers, Zelda, Aerial2, Bicycle, Cafeteria, Chart, Mat, Target, Tools, Portrait | | |
| *CODING PARAMETERS:* | 9/7 DWT 5 levels, 64x64 code-blocks. Lossy compression. Targeted bit-rates. | | |
| | **SBRA** | | |
| **FEATURES** | | | |
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |
| **CODING PERFORMANCE (average)** | | | |
| 0.0625 bps - 0.29 | 0.25 bps - 0.48 | 1 bps - 0.52 | |
| 0.125 bps - 0.42 | 0.5 bps - 0.52 | | |
| **SPEED-UP (average)** | | | |
| 0.0625 bps - 16 | 0.25 bps - 6.75 | 1 bps - 2.8 | |
| 0.125 bps - 9 | 0.5 bps - 3.6 | | |
| | **PSRA** | | |
| **FEATURES** | | | |
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | yes | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | qual |

| | | | |
|---|---|---|---|
| *COMP. COMP. REDUCTION:* | yes | | |
| **CODING PERFORMANCE (average)** | | | |
| 0.0625 bps - 0.28 | 0.25 bps - 0.18 | 1 bps - 0.12 | |
| 0.125 bps - 0.22 | 0.5 bps - 0.15 | | |
| **SPEED-UP (average)** | | | |
| 0.0625 bps - 16 | 0.25 bps - 6.75 | 1 bps - 2.8 | |
| 0.125 bps - 9 | 0.5 bps - 3.6 | | |
| | **PSOT** | | |
| **FEATURES** | | | |
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | yes | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | qual |
| *COMP. COMP. REDUCTION:* | yes | | |
| **CODING PERFORMANCE** | | | |
| | optimal results | | |
| **SPEED-UP (average)** | | | |
| 0.0625 bps - 6.75 | 0.25 bps - 2.75 | 1 bps - 1.5 | |
| 0.125 bps - 4.5 | 0.5 bps - 2.16 | | |

| | |
|---|---|
| **13. NAME:** | Wu |
| **PAPER/S:** | **[108] March 2003** |
| **CLASS.:** | source-channel |

*EXPLANATION*

This paper addresses the transmission of JPEG2000 code-streams over noisy channels. A hybrid optimization rate allocation method is proposed that uses the coding variations of JPEG2000 related with error resilience and some of the capabilities of the code-stream organization. Although other codes could be used, results for convolutional codes and turbo codes are provided, achieving close to optimal results.

**NO RELATED RESULTS**

**14. NAME:** Ardizzone

**PAPER/S:** **[7] September 2003**

**CLASS.:** tiling

*EXPLANATION*

The main goal of the proposed rate control method is to optimize the coding performance when encoding with tiles. The authors state that, when using tiles, it is usual to divide the desired bit-rate by the number of tiles of the image, and then encode each tile at that resulting bit-rate. The proposed method calculates a factor which expresses the informative content of each tile. Using this factor, a bit-rate for each tile is estimated, achieving better results than when using the traditional approach.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | no | | |

**NO RELATED RESULTS**

**15. NAME:** Teerapat

**PAPER/S:** **[80] October 2003**

**CLASS.:** deterministic + model-based

*EXPLANATION*

The proposed method is based on the estimation of the scanning order followed by the encoding process of an image. This scanning order is experimentally determined by the observation of the PCRD. The algorithm first encodes all coding passes of all code-blocks which have some coefficients with magnitude bit-planes higher than 7 and then it finishes the encoding by using the predefined scanning order. Once the target bit-rate is attained, this method encodes $n$ bit-planes further and, at the end, the PCRD is executed to finely select those truncation points included in the final code-stream.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | no | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | yes | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | qual |
| *COMP. COMP. REDUCTION:* | yes | | |

**RESULTS**

| | | | |
|---|---|---|---|
| *IMPLEMENTATION:* | Jasper | *RESULTS:* | table |
| *CORPUS:* | Lena, Baboon, Barbara | | |
| *CODING PARAMETERS:* | not specified (assumed lossy compression) | | |

**CODING PERFORMANCE (average)**

| | |
|---|---|
| 0.1 bps - 0.2 | 1 bps - 0.05 |

(not clear, bps seems displaced)

**SPEED-UP**

not specified

| | |
|---|---|
| **16. NAME:** | SECM, SINC |
| **PAPER/S:** | [4] December 2003, **[5] March 2005** |
| **CLASS.:** | opt-Lagrange |

*EXPLANATION*

Two different methods focused on the optimization of the generalized Lagrange multiplier are proposed. They do not need to compute the slopes of the rate-distortion curves for each code-block since the rate-distortion curves are estimated using a model. This estimation only needs the distortion contribution of each coding pass. Once the estimation of all code-blocks is performed, each curve is classified in either significant or not, in order to process only the significant curves when selecting the best coding passes included in the final code-stream. Although SINC method is a slightly simplified version of SECM, both methods share the same basis in their development.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | no | | |

**NO RELATED RESULTS**

**17. NAME:** Chang

**PAPER/S:** **[19] January 2004**

**CLASS.:** deterministic

*EXPLANATION*

A rate control method focused on the optimization of the bit-rate for a target quality, i.e. distortion constraint, is proposed. The main idea behind it is to compute the rate-distortion slope of the wavelet coefficients quantized at every coding pass before the actual encoding. This computation is efficiently performed just distinguishing significant from refinement bits at each bit-plane, computing the distortion contribution of each coefficient and saving these computations for each code-block. Then, it is easy to determine which coding passes have to be encoded for each code-block and, therefore, the sample data coding stage just encodes those coding passes that are included in the final code-stream.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | no | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | bit |
| *COMP. COMP. REDUCTION:* | yes | | |

**NO RELATED RESULTS**

**18. NAME:** Qin

**PAPER/S:** **[75] June 2004**

| **CLASS.:** | deterministic + model-based |
|---|---|
| *EXPLANATION* | |

An study of the rate-distortion curve of code-blocks is presented. The purpose of this study is to estimate the rate-distortion curve using less calculations and less memory consumption than the one needed with the PCRD method. This is achieved modeling the rate-distortion curve of code-blocks with statistical information of 20 images. This model concludes that the number of significant and refinement bits at each bit-plane is enough to fairly estimate the rate-distortion curve of each code-block. Although it is not explicitly explained, it seems that once the curve of every code-block are estimated, the PCRD is executed over these curves.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | dec |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | no | | |

**RESULTS**

| | | | |
|---|---|---|---|
| *IMPLEMENTATION:* | Jasper | *RESULTS:* | table |
| *CORPUS:* | Lossy: Barbara, Goldhill / Lossless: Lena, Peppers, Canyon | | |
| *CODING PARAMETERS:* | 9/7 DWT. Lossy compression. Targeted bit-rates. 5/3 DWT. Lossless compression. Targeted bit-rates. | | |

**LOSSY CODING PERFORMANCE (worst / best / average)**

| | |
|---|---|
| 0.1 bps - 0.17 / 0.12 / 0.145 | 1 bps - 0.27 / 0.18 / 0.225 |
| 0.3 bps - 0.2 / 0.1 / 0.1 | 2 bps - 0.01 / 0.13 / 0.07 |
| 0.5 bps - 0.24 / 0.12 / 0.17 | |

**LOSSLESS CODING PERFORMANCE (worst / best / average)**

| | |
|---|---|
| 0.1 bps - -0.15 / -0.31 / 0.24 | 1 bps - 0.11 / 0.04 / 0.07 |
| 0.3 bps - 0.13 / 0.06 / 0.08 | 2 bps - 0.13 / 0.06 / 0.08 |
| 0.5 bps - 0.14 / 0.05 / 0.1 | |

**SPEED-UP**

not specified

**19. NAME:** Du

**PAPER/S:** **[26] November 2004**

**CLASS.:** deterministic

*EXPLANATION*

The proposed method uses an encoding order based on the number of bit-planes of each code-block. It starts encoding those code-blocks which have more magnitude bit-planes, assuming that the higher the bit-plane is, higher is the probability to include their coding passes in the final code-stream. To reduce the computational complexity of the sample data coding stage, at the end of each coding pass the rate-distortion slope is calculated and, when the target bit-rate is attained, the smallest rate-distortion slope of all encoded code-blocks is considered as a threshold. The encoding of the remaining code-blocks only considers those coding passes with rate-distortion slopes greater than the current threshold and, obviously, the threshold rate-distortion slope and bit-rate is updated at the end of each coding pass encoding to further reduce the number of encoded coding passes.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | no | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |

**RESULTS**

| | |
|---|---|
| *IMPLEMENTATION:* | not specified |
| *RESULTS:* | graphic (coding performance) table (speedup) |
| *CORPUS:* | Lena, Baboon |
| *CODING PARAMETERS:* | 9/7 DWT. Lossy compression. Targeted bit-rates. |

**CODING PERFORMANCE**

optimal results

**SPEED-UP (min / max / average)**

| | |
|---|---|
| 0.0625 bps - 7.5 / 8.8 / 8.15 | 0.5 bps - 1.8 / 2.7 / 2.25 |
| 0.125 bps - 5.4 / 6.7 / 6.05 | 1 bps - 1.2 / 1.7 / 1.45 |
| 0.25 bps - 2.9 / 4.5 / 3.7 | |

| | |
|---|---|
| **20. NAME:** | Vikram |
| **PAPER/S:** | **[103] January 2005** |
| **CLASS.:** | deterministic + model-based |

*EXPLANATION*

This rate control method uses the Lagrange multiplier to find the optimal solution to the rate-distortion optimization problem. However the Lagrange multiplier selects the coding passes included in the final code-stream before the actual encoding thanks to a model of the rate-distortion slope of the coding passes. This model is the main contribution of the method: the decrement in distortion and increment in bit-rate is computed for each coding pass using the original image, through the number of significant and refinement bits at each bit-plane. The main drawback of the method is that the target bit-rate can not be attained precisely, since the model produce variations of 5% when estimating the actual bit-rate of the final code-stream.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | dec |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |

**RESULTS**

| | | | |
|---|---|---|---|
| *IMPLEMENTATION:* | Jasper | *RESULTS:* | table |
| *CORPUS:* | Mountain, Baboon, Lena, Boy, Church | | |
| *CODING PARAMETERS:* | 9/7 DWT 3 levels, 32x32 code-blocks. Lossy compression. Targeted bit-rates. | | |

**CODING PERFORMANCE (average bit-rate variation / average PSNR)**

| | |
|---|---|
| 0.0625 bps - 2.9% / 0.07 | 0.25 bps - 3.1% / 0.13 |
| 0.125 bps - 2.5% / 0.05 | 0.5 bps - 2.7% / 0.12 |

**SPEED-UP (min / max / average)**

| | |
|---|---|
| 0.0625 bps - 23.8 / 16 / 19 | 0.25 bps - 10.6 / 5.9 / 8.4 |
| 0.125 bps - 16.4 / 10.1 / 13.3 | 0.5 bps - 6.4 / 3.3 / 5 |

**21. NAME:** MEPRD
**PAPER/S:** **[49] January 2005**
**CLASS.:** deterministic

*EXPLANATION*

The proposed rate control method performs the sample data coding stage and the rate control simultaneously. This is carried out calculating the rate-distortion slope at the encoding of each coding pass and then selecting the next coding pass to be encoded depending on the rate-distortion slope value of all code-blocks (i.e. the code-block with the greatest rate-distortion slope is selected). Note that this method does not consider the feasible truncation points, allowing the selection of truncation points which do not lie in the convex hull, although the authors claim that this does not penalize the coding performance when compared to the optimal PCRD method.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | yes | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |

**RESULTS**

| | | | |
|---|---|---|---|
| *IMPLEMENTATION:* | own | *RESULTS:* | table |
| *CORPUS:* | Army, Cathedral, Chimp, Town | | |
| *CODING PARAMETERS:* | 5/3 DWT 3 levels, 32x32 code-blocks. Lossless compression. Targeted bit-rates. | | |

**CODING PERFORMANCE**

optimal results

**SPEED-UP FOR THE COMPLETE ENCODING (min / max / average)**

| | |
|---|---|
| 0.0625 bps - 1.95 / 2.1 / 2 | 0.5 bps - 1.05 / 1.27 / 1.1 |
| 0.125 bps - 1.6 / 1.65 / 1.6 | 0.75 bps - 1 / 1.02 / 1 |
| 0.25 bps - 1.25 / 1.32 / 1.27 | 2 bps - |

**SPEED-UP OF THE SAMPLE DATA CODING STAGE (min / max / average)**

(considering the sample data coding stage a 65% of the whole encoding process)

| | |
|---|---|
| 0.0625 bps - 4 / 5.15 / 4.3 | 0.5 bps - 1.08 / 1.48 / 1.16 |
| 0.125 bps - 2.4 / 2.54 / 2.3 | 0.75 bps - 1 / 1.03 / 1 |
| 0.25 bps - 1.4 / 1.6 / 1.5 | |

**22. NAME:** Chebil

**PAPER/S:** **[20] January 2006**

**CLASS.:** model-based + deterministic

*EXPLANATION*

A model-based method that discards some of the lowest bit-planes before starting the encoding is presented. The discarding is based on three parameters: target bit-rate (low, medium, high), image resolution, and image type (edge, texture, plain). Based on these measures, some bit-planes of each subband are discarded, reducing the computational complexity. Once the encoding is performed, the optimal PCRD method is applied.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | no | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | no |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | qual |
| *COMP. COMP. REDUCTION:* | yes | | |

**NO RELATED RESULTS**

**23. NAME:** Liu

**PAPER/S:** **[58] July 2006**

**CLASS.:** deterministic

*EXPLANATION*

Rather than to optimize the quality for a target bit-rate, this paper is focused on the optimization of the bit-rate for a target quality, and also reducing the computational complexity of the coder. The purpose of the research is to obtain a qualitative measure which fairly identifies the perceptual quality of an image. The developed method uses a vision model based on the human vision perception and on distortion metrics. Besides, the method reduces the computational complexity of the coder thanks to the identification of those coding passes that have to be encoded.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | yes |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | bit |
| *COMP. COMP. REDUCTION:* | yes | | |

**NO RELATED RESULTS**

**24. NAME:** MM

**PAPER/S:** **[50] August 2006**

**CLASS.:** deterministic

*EXPLANATION*

The purpose of the presented rate control methods is to control the rate allocation of volumetric data using low computational resources. Two methods are presented. The first one is based on the PCRD, obtaining optimal results although the computational complexity is highly penalized. The second method is focused on the reduction of the computational resources used by PCRD, using a model based on the data variance of each component. The so-called MM method reduces the computational complexity achieving near-optimal results. The experimental results, as well as the development of the method, are devised for a 3D meteorological data set. Part 2 of the standard is used to apply a KLT transformation to the volumetric data.

**FEATURES**

| | | | |
|---|---|---|---|
| *LAYERS CONSTRUCTION:* | yes | *ANY DIST. MEAS.:* | no |
| *INCREMENTAL ENCODING:* | no | *DIST. MEAS. OR. IMAGE:* | yes |
| *USE IN THE DECODER:* | no | *QUALITY/BIT-RATE OPT.:* | both |
| *COMP. COMP. REDUCTION:* | yes | | |

**NO RELATED RESULTS**

### 3.1.4 Comparison

Some of the reviewed rate control methods are similar, although they are devised from different points of view. We would stress some of these similarities: Chang [19], Qin [75] and Vikram [103] counts the number of significant and refinement bits at each bit-plane in order to model the rate-distortion; MSD [17] and Du [26] use an encoding strategy pretty similar, although Du obtains much better results than MSD; {E-}IREC [116], PSOT [110] and MEPRD [49] use the rate-distortion slope to determine when to stop encoding.

Figure 3.2 depicts the reviewed rate control methods, classified in their categories and in chronological order. The methods directly related with the research of this thesis are emphasized in gray. Note that almost all of them are deterministic. This is explained in [110] because of the difficult task to develop a general rate-distortion model that works well for different images.

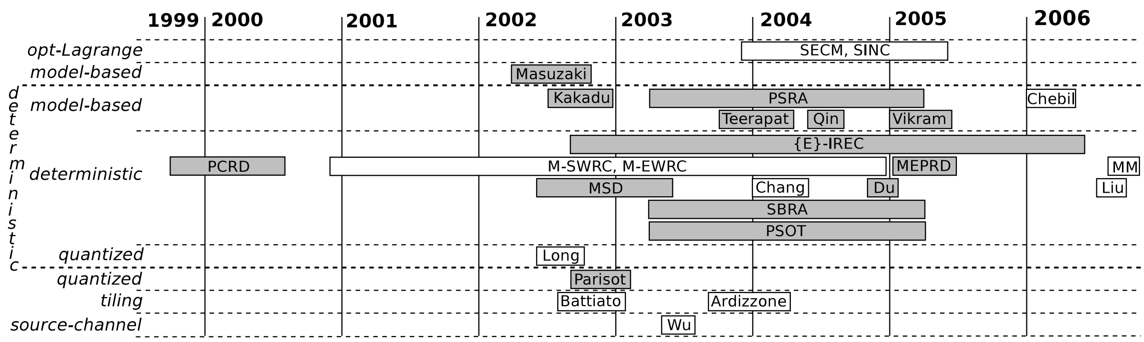



Figure 3.2: *The reviewed JPEG2000 rate control methods.*

Table 3.2 collects the considered features for all the reviewed rate control methods. Almost all of them allow the construction of quality layers implicitly, however some of them do not allow it: Parisot [67] carries out the rate control through the quantization step sizes, and this do not allow to determine fine bit-rates to allocate quality layers; Chang [19], Battiato [13], Teerapat [80] and Chebil [20] use methods that, although they reduce the computational complexity of the encoding process, they still need the Lagrange multiplier to construct quality layers. The incremental encoding is needed only for a few methods and its use allows the achievement of high speed-ups, although it may suppose a drawback for implementations with memory constrained resources.

Table 3.2: *Features of the reviewed rate control methods.*

| *METHOD* | cat | layer | meas | inc | orig | decod | opt | red |
|---|---|---|---|---|---|---|---|---|
| PCRD | det | yes | yes | no | yes | no | both | no |
| M-{S,E}WRC | det | yes | yes | no | yes | no | both | yes |
| Masuzaki | pred | no | no | no | no | yes | qual | yes |
| MSD | det | yes | yes | no | yes | no | both | yes |
| Long | quant + det | no | no | no | no | no | qual | no |
| Kakadu | det + pred | yes | yes | no | yes | no | both | yes |
| Battiato | tile | | | | | | | |
| Parisot | quant | no | no | no | yes | no | both | yes |
| {E-}IREC | det | yes | yes | yes | yes | no | both | yes |
| SBRA | det | yes | yes | no | yes | no | both | yes |
| PSRA | pred + det | yes | no | yes | yes | no | qual | yes |
| PSOT | det | yes | no | yes | yes | no | qual | yes |
| Wu | channel | | | | | | | |
| Ardizzone | tile | yes | yes | no | yes | no | both | no |
| Teerapat | det + pred | no | no | yes | yes | no | qual | yes |
| SECM, SINC | opt-Lag | yes | no | no | yes | no | both | no |
| Chang | det | no | no | no | yes | no | bit | yes |
| Qin | det + pred | yes | no | no | dec | no | both | no |
| Du | det | no | yes | no | yes | no | both | yes |
| Vikram | det + pred | yes | no | no | dec | no | both | yes |
| MEPRD | det | yes | yes | yes | yes | no | both | yes |
| Chebil | pred + det | no | no | no | no | no | qual | yes |
| Liu | det | yes | yes | no | yes | no | bit | yes |
| MM | det | yes | no | no | yes | no | both | yes |
| CPI/ROC/CoRD | pred | yes | no | yes | no | yes | qual | yes |

LEGEND:

| | |
|---|---|
| *cat- category* | *orig- DIST. MEAS. BASED ORIG. IMAGE* |
| *layer- LAYERS CONS.* | *decod- USE IN THE DECODER* |
| *meas- ANY DIST. MEAS.* | *opt- QUALITY/BIT-RATE OPTIMIZATION* |
| *inc- INCREM. ENCOD.* | *red- COMPUTAT. COMPLEX. REDUC.* |

The use of distortion measures based on the original image is needed for almost all methods, and note that those methods that do not use these measures usually obtain a poor coding performance. These methods are the only ones that could be used in the decoder in order to control the rate-distortion of already encoded code-streams, although this issue is usually not addressed by the authors. Almost all the methods allow the rate-distortion optimization for both a target bit-rate or a target quality and most of them reduce the computational complexity.

Table 3.3 collects the coding performance results of the reviewed methods. We only show the average result in order to simplify the reading. Note that the rate control methods which obtain an optimal coding performance all have similar speed-ups, of about 7 at 0.0625 bps, apart from {E-}IREC [116], which obtains a speed-up of 14 at the same bit-rate.

Table 3.3: *Coding performance evaluation of the reviewed rate control methods.*

| | *coding performance* | | | | | | *speed-up* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *METHOD* | .0625 | .125 | .25 | .5 | 1 | 2 | .0625 | .125 | .25 | .5 | 1 |
| PCRD | optimal | | | | | | 1 | | | | |
| Masuzaki | 1 | | | | | | | | | 3.4 | |
| MSD | 0.0 | 0.03 | 0.0 | 0.0 | 0.0 | | 7.5 | 5.12 | 3.08 | 1.92 | 1.27 |
| Kakadu | optimal | | | | | | | | 5.3 | 3.63 | 2.38 |
| Parisot | | | 0.0 | -0.01 | 0.0 | 0.31 | not specified | | | | |
| {E-}IREC | optimal | | | | | | 14 | 7.5 | 4 | 2.5 | 1.25 |
| SBRA | 0.29 | 0.42 | 0.48 | 0.52 | 0.52 | | 16 | 9 | 6.75 | 3.6 | 2.8 |
| PSRA | 0.28 | 0.22 | 0.18 | 0.15 | 0.12 | | 16 | 9 | 6.75 | 3.6 | 2.8 |
| PSOT | optimal | | | | | | 6.75 | 4.5 | 2.75 | 2.16 | 1.5 |
| Teerapat | | 0.2 | | | 0.05 | | not specified | | | | |
| Qin | | 0.145 | 0.1 | 0.17 | 0.225 | 0.07 | not specified | | | | |
| Du | optimal | | | | | | 8.15 | 6.05 | 3.7 | 2.25 | 1.45 |
| Vikram | 0.07 | 0.05 | 0.13 | 0.12 | | | 19 | 13.3 | 8.4 | 5 | |
| MEPRD | optimal | | | | | | 4.3 | 2.3 | 1.5 | 1.16 | |
| CPI | 0.1 | 0.11 | 0.12 | 0.11 | 0.12 | 0.1 | | | | | |
| ROC | 0.1 | 0.11 | 0.11 | 0.1 | 0.07 | 0.06 | 16.6 | 10 | 6.6 | 4 | 2.9 |
| CoRD | 0.08 | 0.1 | 0.1 | 0.07 | 0.06 | 0.04 | | | | | |

LEGEND: *Each column depicts the bit-rate in bps (e.g. 0.0625 bps) and the coding performance is expressed in dB meaning, for instance, 0.1 dB worse than the optimal PCRD method.*

Although the rate control methods presented in this thesis are not introduced until Chapters 4, 5 and 6 we show the features and results obtained by them in the rows named CPI, ROC and CoRD, in order to facilitate the comparison. We must stress that the main difference between the proposed rate control methods and the ones reviewed in this section is that CPI, ROC and CoRD can be used in the decoder side of the coding process.

## 3.2 Quality scalability of JPEG2000

### 3.2.1 Quality layers construction

Quality layers play an important role in the coding system of JPEG2000. They are the fundamental mechanism that supplies quality progression and quality scalability to JPEG2000 code-streams. The quality progression and scalability are useful features for several applications, for instance, when a code-stream needs to be truncated or when it needs to be interactively transmitted using the JPIP protocol defined in the Part 9 of the standard [46]. A code-stream that does not contain quality layers might cause several drawbacks when its manipulation is needed, therefore it is important to encode images constructing an adequate number of quality layers.

The code-stream organization is explained in Section 2.2.5, showing how quality layers are defined within the tile-parts of each tile-stream. Recall that within this organization, quality layers are just a collection of packets. The organization of these packets within the code-stream, such as the progression order or the number of tile-parts, is not considered by the rate control method. The rate control method only defines the content of these packets (i.e. the coding passes that they contain) and the overall bit-rate of these packet collections, which is actually the bit-rate of the quality layers. Quality scalability is achieved by the identification of these packet collections, available through the decoding of the packet headers.

Here, we consider the quality layers construction when encoding a single tile, since the rate control methods are usually defined within this framework. The rate-distortion optimization problem for images with multiple tiles has been addressed in Battiato [13] and Ardizzone [7].

In Section 3.1 we have considered the rate-distortion optimization problem as the optimization of the image quality at a target bit-rate $R^{max}$, or the optimization of the code-stream bit-rate for a target quality $D^{max}$. However, even if a target bit-rate or quality is not specified, rate control should be applied to construct quality layers within the code-stream.

The construction of quality layers considers the optimization of the distortions, referred to as $D^m$, at the bit-rates $R^0 < R^1 < R^2$.... The complementary problem (to optimize the bit-rates $R^n$ for the distortions $D^0 > D^1 > D^2$...) is not considered here. From the point of view of the PCRD method, the construction of quality layers with bit-rates $R^m$ uses exactly the same optimization criterion at each bit-rate as the used for a single bit-rate $R^{max}$. Recall that with the PCRD method, the rate-distortion optimization is simplified to the search of the feasible truncations points $n_j^{\lambda_m} = max\{j_k \in \mathcal{N}_j | S^{j_k} > \lambda_m\}$ for some value of the Lagrange multiplier $\lambda_m$, with $R(\lambda_m) = R^m$, where $\mathcal{N}_j$ denotes the set of feasible truncation points and $j_k$ denotes the enumeration of these feasible truncation points, with associated rate-distortion slopes $S^{j_k}, S^{j_k} < S^{j_{k+1}}$.

Note that the optimization search for the next bit-rate $R^{m+1}$ must not consider the complete set of feasible truncation points $\mathcal{N}_j$ again, considering that the points $n_j^{\lambda_m}$ do no longer belong to $\mathcal{N}_j$. If we consider $S^{j_m}$ as the minimum rate-distortion slope of the selected feasible truncation points for the bit-rate $R^m$, the search of $n_j^{\lambda_{m+1}}$ is simplified to the same problem above, with $n_j^{\lambda_{m+1}} = max\{j_k \in \mathcal{N}_j | S^{j_m} > S^{j_k} > \lambda_{m+1}\}$.

Once the rate control method has selected the sets of feasible truncation points (i.e. $n_j^{\lambda_0}, n_j^{\lambda_1}, n_j^{\lambda_2}, ...$) for the bit-rates $\{R^0, R^1, R^2, ...\}$ the generation of packets is performed straightforward. As explained in Section 2.2.5, the generation of packets encodes some extra information encapsulated at the header of each packet regarding the content of the packet. Obviously this information holds some bytes and, the more quality layers the code-stream has, the larger the information encoded at the packet headers is. Consequently, one might expect a penalization in the coding performance. Fortunately, the packet headers coding and the code-stream syntax are highly efficient and the use of quality layers does not penalize significantly the coding performance.

**Remark 3.2.1** *The coding performance is not penalized even if the code-stream contains a large number of quality layers. For example, when encoding the Musicians image at 1 bps using a single quality layer code-stream, the recovered image has a distortion of 32.47 dB compared to the original one. When using a code-stream containing 20 quality layers, the*

*coding performance is penalized only in 0.04 dB at the same bit-rate. For a code-stream containing 80 quality layers, this penalization is only 0.05 dB!*
*Packets headers only need few bytes, even if many quality layers are generated. The code-stream generated when encoding the Musicians image constructing 160 quality layers is only 0.5% larger than when constructing a single quality layer code-stream.*

However, the most interesting issue related with quality layers is: 1) how to determine the optimal number of quality layers that a code-stream should contain and 2) how to determine the bit-rate of each quality layer. This needs a further discussion, addressed in next section.

### 3.2.2 Rate allocation of quality layers

A consequence of allocating quality layers at the bit-rates $\{R^0, R^1, R^2, ...\}$ is that when the code-stream is decoded at one of these quality layer boundaries, say $R^m$, the distortion of the recovered image is almost optimal. The recovered image is not exactly the optimal one because of the packet headers needed by the quality layers before $m$, which we name *side information* and is calculated as $\sum_{l=0}^{m} e^l$, where $e^l$ denotes the number of bytes needed by the packet headers of the quality layer $l$.

It is clear that if we know, a priori, the bit-rates at which the code-stream is going to be decoded, referred to as *operational bit-rates*, it is easy to optimize the distortion at those bit-rates, minimizing the overall packet headers information just adjusting the bit-rates of quality layers to the operational bit-rates. However, this is not the usual case and it is worth to construct the code-stream using some general allocation strategy which could work reasonably well for most applications and scenarios.

Long before JPEG2000, this rate-distortion optimization problem has been studied on convex hull based approaches [105, 85, 22]. In the framework of JPEG2000, this had not been addressed in any way until 2005, giving just some recommendations on the number and bit-rate of quality layers based on experience [100, Chapter 8.4.1]. From the point of view of the uneven error protection for embedded code-streams, the rate-distortion optimization has been studied under some *expected multi-rate distortion measure* (EMRD) [84, 16, 74] and, from the point of view of JPEG2000, this has been studied in [27]. The EMRD measure is extremely useful to evaluate the optimality of a JPEG2000 code-stream in terms of the rate-distortion optimization and, consequently, the same authors have continued their study, finalized in December 2005 [107].

The EMRD weights the distortion of the image recovered at some bit-rates by the probability to recover the image at those bit-rates. In other words, EMRD defines a function that reflects the probability $p(R)$ of the code-stream $\mathcal{X}$ to be decoded at bit-rate $R, R \in [0, length(\mathcal{X})]$. The averaged EMRD over the complete bit-rate of $\mathcal{X}$ is defined as

$$\int_0^{length(\mathcal{X})} D(R)p(R)dR$$

where $D(R)$ represents the distortion of the recovered image at bit-rate $R$. First, the authors approach the problem considering the side information as null, i.e. $\sum_i e^i = 0$. From this point of view, they consider two bit-rate distributions: the uniform distribution and the logarithmic distribution. The uniform distribution gives a good approximation to see the overall optimality over the complete bit-rate range of a code-stream. The logarithmic distribution gives a better approximation to real scenarios, where the code-stream is commonly more decoded at low bit-rates and less as the bit-rate grows, as stated by the authors.

When considering the uniform distribution, with the constant $p(R) = (1/length(\mathcal{X}))$, the interleaving of the feasible truncation points, i.e. those points that lie on the convex hull, in decreasing rate-distortion slopes is the optimal solution. This is not surprising if we consider that the rate-distortion slope precisely look for the distortion optimality over the complete bit-rate range. However, it is worth noting that this would not be achieved if the authors do not consider that, within the coding passes of a feasible truncation point, the distortion reduction is linear with the number of decoded bits.

The optimization problem for a logarithmic distribution is more interesting, considering $p(R) = v \cdot \alpha^R$, for some $\alpha \neq 1$ and $v$ as a normalizing factor. The proposed approach is based on the modification of the convex hull of code-blocks to a $\alpha$-convex hull varying the horizontal axis in order to adjust it to the logarithmic distribution. Then, the same greedy approximation on the re-calculated rate-distortion slopes on the convex hull is used in order to obtain the optimal quality layers for a JPEG2000 code-stream, evaluated by the EMRD measure. This provides the optimal allocation strategy for the logarithmic distribution.

However, these approximations does not consider the side information and, consequently, do not generate the actual optimal code-stream. The authors claim that this is a challenging optimization problem. The main difficulty here is that the bit-rate of $e^i$ can not be estimated without encoding all packet headers at that bit-rate. The solution to this problem is to use dynamic programming. Although the optimal solution is not achieved, the smart algorithm proposed is optimal for practical implementations and, in addition, it is optimized using a predictive technique which reduces the computational complexity.

Although the approach presented in this research is outstanding and is the first one that analyzes the optimality of JPEG2000 code-streams from the point of view of introducing smart quality layers allocation strategies, the experimental results are quite disappointing. The optimization of the quality layers allocation strategy is performed for uniform, logarithmic and Laplacian distributions. The degree of improvement varies from image to image, but it is usually small. The authors explain this poor improvement due to the already good approach of PCRD and to the optimal fractional bit-plane coding of JPEG2000, which already generates code-stream segments with decreasing rate-distortion slopes.

### 3.2.3 Experimental results

One important feature of the rate control methods developed in this thesis is that they supply quality scalability to code-streams that contain a single or few quality layers. We have seen that the quality scalability is usually provided by a good strategy of quality layers allocation, therefore it is important to evaluate the coding performance achieved by quality layers.

The purpose of this section is to evaluate the coding performance obtained with three usual strategies of quality layers allocation in order to compare them to the rate control methods proposed in the following chapters. We do not assume any pre-defined operational bit-rates because this may vary depending on the scenario of use, so the EMRD is considered a uniform probability distribution. Since in [107] it is stated that the usual PCRD approach already provides the optimal results for quality layers allocation under uniform probability distributions, in the presented experiments we have used the PCRD implementation of Kakadu.

In all comparisons of this section, and in the following chapters, we use the coding performance obtained by the PCRD as a maximum threshold, comparing the coding performance obtained with the use of quality layers against this threshold. Note that the PCRD coding performance is just a theoretical reference because it can only be used in the encoding process, and the methods we will evaluate are devised for already encoded code-streams.

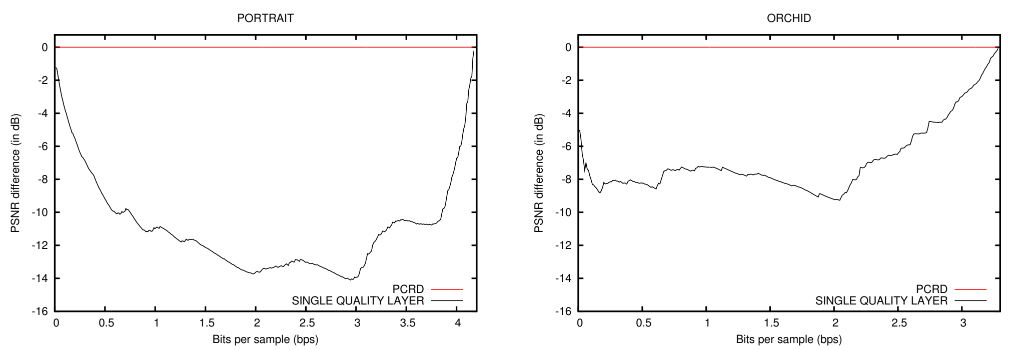
For each allocation strategy, each image of the corpus ISO 12640-1 has been encoded constructing a code-stream containing different number of quality layers. Then, each code-stream has been decoded at 600 bit-rates (called *control points*) equivalently spaced from 0.001 to 6 bps, or finishing at the end of the code-stream if the complete encoding constructs a code-stream shorter than 6 bps. The recovered image at each bit-rate has been compared to the original image in terms of PSNR, and the difference between this PSNR and the PSNR obtained when encoding the image with the PCRD method at this bit-rate has been computed. The optimal PSNR obtained by the PCRD method is depicted in the graphics as the straight line.

It is interesting to see how the coding performance of the evaluated code-streams varies at different bit-rates in order to highlight the quality scalability that can be obtained. For this reason, in all the experimental sections of this thesis we provide results that depict the results for images in single graphics. In order to compare the methods in a more global scenario, we also report results computing the average of the eight images of the corpus ISO 12640-1.

We have used a derived quantization and the RESTART coding variation, which slightly penalize the coding performance. In Section 4.3 –page 95– we explain the use of these parameters.

### Single quality layers code-streams

Graphic 3.1 depicts the coding performance obtained with code-streams which contain a single quality layer. This is the worst coding performance that can be obtained due to the complete lack of quality scalability of the code-stream, reaching an inferior threshold. The average PSNR difference for all the images of the corpus ISO 12640-1 is as low as 12.503 dB worse than the optimal PCRD method. However, the coding performance obtained at low bit-rates is usually of major interest: from 0.001 to 1 bps, the average PSNR difference is 8.167 dB worse than PCRD. Note that in this experiment, the progression order used could affect the coding performance. However, with single component images and without any quality layer, the progressions LRCP, RLCP, RPCL are equivalent, and PCRL, CPRL are also equivalent and give worse results because the progression orders the code-blocks using the spatial position. So, here and in the remaining experiments, we only use the LRCP progression order.

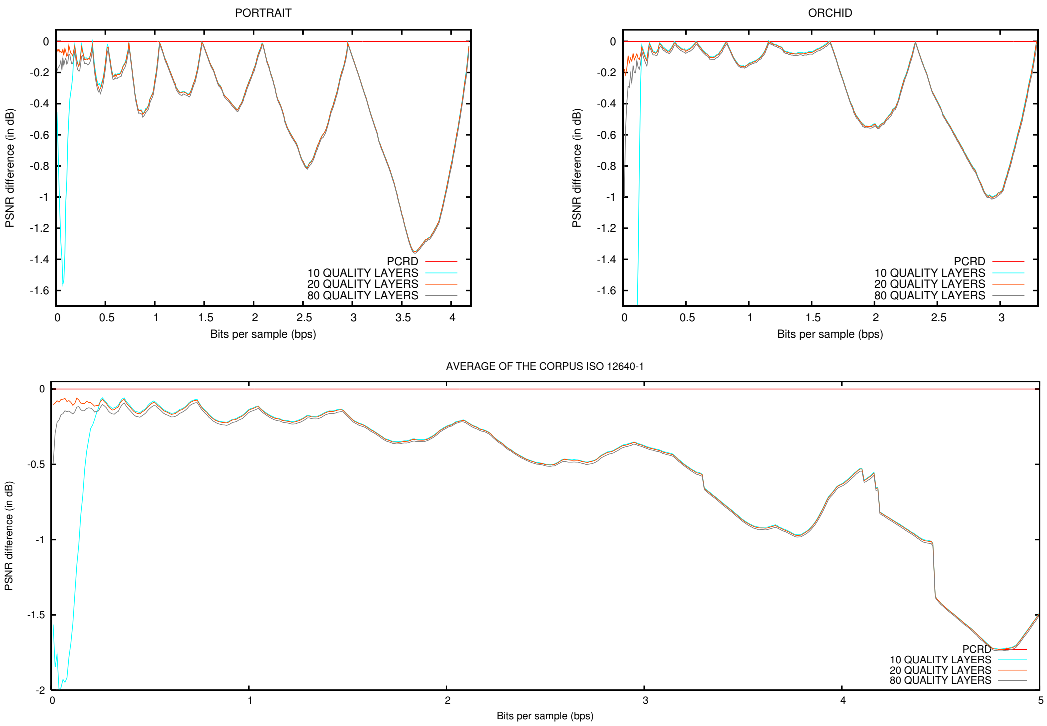



*Coding parameters detailed in Table 3.4*

Graphic 3.1: *Coding performance evaluation of single quality layers code-streams.*

### Common strategies of quality layers allocation

The first strategy of quality layers allocation that we evaluate is when they are spaced logarithmically, in terms of bit-rate, along the complete bit-rate range of the code-stream. This is the default mode of quality layers allocation of Kakadu, yielding high coding performance at very high compression factors. Graphic 3.2 depicts the results obtained with this allocation strategy for two images and in average for all the images of the corpus ISO 12640-1. Note the fluctuations caused by this allocation strategy: at the quality layer boundaries, the coding performance obtained with the code-stream containing different number of quality layers is practically the same as the coding performance obtained with the PCRD method.

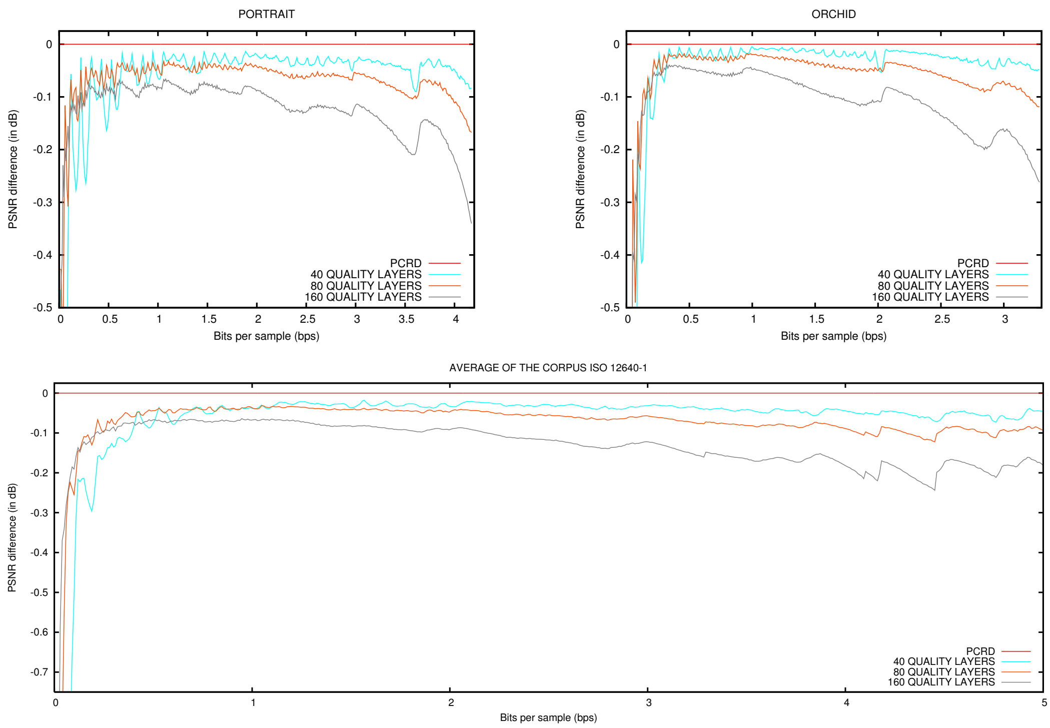



*Average PSNR diff. - 10 quality layers: -0.377 dB (0.001 to 1 bps) -0.638 dB (total)*
*Average PSNR diff. - 20 quality layers: -0.129 dB (0.001 to 1 bps) -0.598 dB (total)*
*Average PSNR diff. - 80 quality layers: -0.164 dB (0.001 to 1 bps) -0.614 dB (total)*

*Coding parameters detailed in Table 3.4*

Graphic 3.2: *Coding performance evaluation of code-streams containing quality layers logarithmically spaced.*

The second strategy of quality layers allocation is to space them equivalently, in terms of bit-rate, along the complete bit-rate range of the code-stream. This yields better coding performance at medium and high bit-rates. Graphic 3.3 depicts the results obtained with this allocation strategy for two images and in average for all images of the corpus ISO 12640-1. The coding performance obtained with equivalently spaced quality layers is more uniform along the bit-rate range.




*Average PSNR diff. - 40 quality layers: -0.205 dB (0.001 to 1 bps) -0.073 dB (total)*
*Average PSNR diff. - 80 quality layers: -0.117 dB (0.001 to 1 bps) -0.085 dB (total)*
*Average PSNR diff. - 160 quality layers: -0.125 dB (0.001 to 1 bps) -0.152 dB (total)*

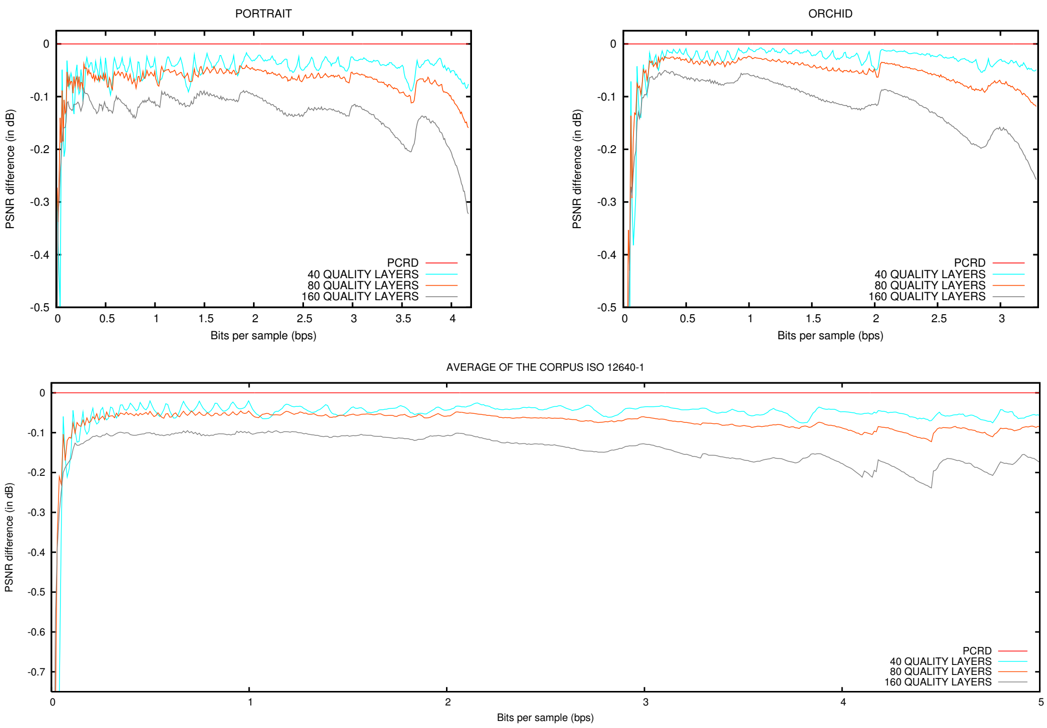
---

*Coding parameters detailed in Table 3.4*

Graphic 3.3: *Coding performance evaluation of code-streams containing quality layers equivalently spaced.*

The main drawback allocating quality layers equivalently spaced is that, at high compression factors, the coding performance is far worse than the obtained with, for instance, code-streams containing quality layers logarithmically spaced. In order to improve the coding performance at low bit-rates we analyze an allocation strategy that distributes the quality layers equivalently spaced, in terms of bit-rate, at three bit-rate intervals: from 0.001 to 0.5

bps, from 0.5 to 1 bps and from 1 bps to the end of the code-stream. Experience indicates that a good allocation strategy is to situate 25% of quality layers in the first interval,15% in the second interval, and the remaining 60% in the third interval. Graphic 3.4 depicts the results obtained with this allocation strategy for two images and in average for all images of the corpus ISO 12640-1. Compared to the previous strategy of quality layers allocation, the coding performance at low bit-rates is improved, while at medium and high bit-rates it is maintained.

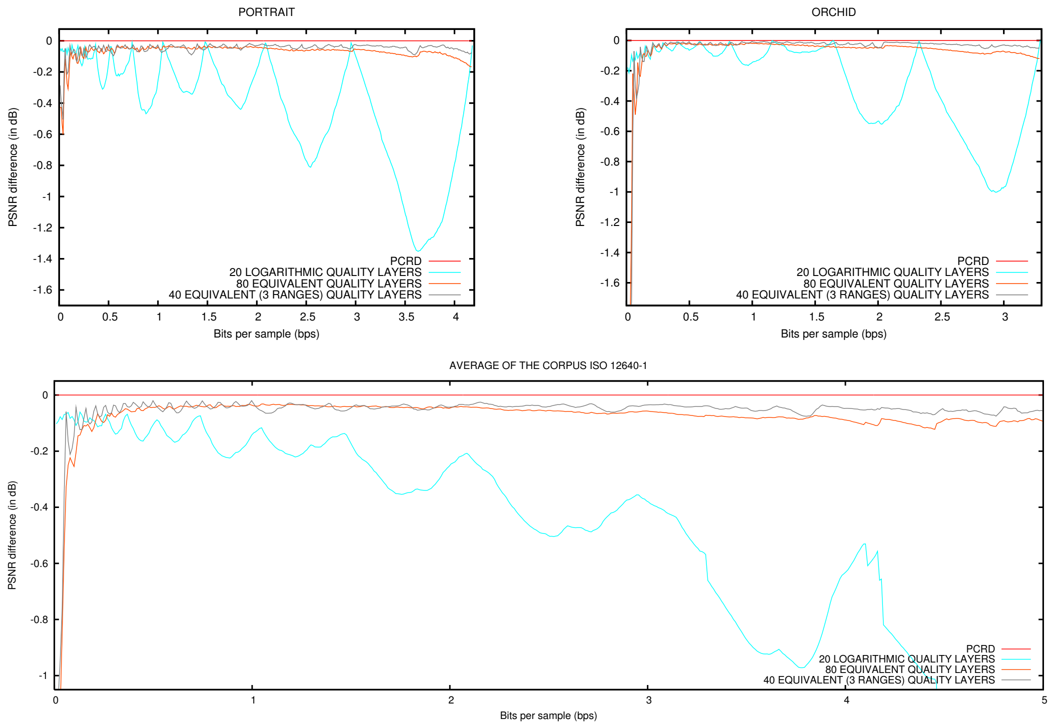



*Average PSNR diff. - 40 quality layers: -0.093 dB (0.001 to 1 bps) -0.058 dB (total)*
*Average PSNR diff. - 80 quality layers: -0.086 dB (0.001 to 1 bps) -0.084 dB (total)*
*Average PSNR diff. - 160 quality layers: -0.134 dB (0.001 to 1 bps) -0.159 dB (total)*

*Coding parameters detailed in Table 3.4*

Graphic 3.4: *Coding performance evaluation of code-streams containing quality layers equivalently spaced in three bit-rate ranges.*

Graphic 3.5 compares the three allocation strategies depicting the best results obtained for each strategy. Note that at high compression factors, the best allocation strategy is the logarithmic, but at medium and low compression factors the best allocation strategy is to distribute quality layers equivalently. It is worth noting that other allocation strategies might obtain a better coding performance than the presented in this section and that, depending on the scenario, this allocation could be optimized to fit the requirements of the application.




*Av. PSNR diff. - 20 logarithmic quality layers: -0.129 dB (0.001 to 1 bps) -0.638 dB (total)*
*Av. PSNR diff. - 80 equivalent quality layers: -0.117 dB (0.001 to 1 bps) -0.085 dB (total)*
*Av. PSNR diff. - 40 equ. (3 rang.) qual. layers: -0.093 dB (0.001 to 1 bps) -0.058 dB (total)*

*Coding parameters detailed in Table 3.4*

Graphic 3.5: *Evaluation of the best coding performance achieved by different allocation strategies of quality layers.*

### Visual comparison

Regarding the qualitative analysis, Figure 3.3 provides a visual comparison of a region of the Portrait image when code-streams containing different number and strategies of quality layer allocation are decoded at 0.03125 bps. Note the visual improvement thanks to the use of quality layers. At the analyzed bit-rate, both strategies of quality layer allocation are visually equivalent. Note also the blur image recovered from the single quality layer code-stream.

Table 3.4: *Coding parameters of the experiments.*

| |
|---|
| RESULTS: *average and two images of the corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.* |
| JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, different number of quality layers, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.* |
| IMPLEMENTATION: *Kakadu v4.5* |


BIT-RATE: *0.03125 bps (compression factor 256:1)* AREA: *600x600*
*Coding parameters detailed in Table 3.4*

Figure 3.3: *Visual comparison among different strategies of quality layers allocation.*

# Chapter 4

# Rate control method for coder and decoder

This chapter introduces the rate control method Coding Passes Interleaving (CPI). The motivations on the development of this rate control method come from two drawbacks that the use of quality layers may cause, explained in the first section. The second section presents the simple interleaving-based rate control method CPI, and in the last section its coding performance is evaluated.

This research has been presented at the Data Compression Conference in March 2006 [12].

## 4.1 Rate control of already encoded code-streams

Let $\mathcal{T}^l$ denote all the packets belonging to the quality layer $l$. When the rate control method sets the number and bit-rate of quality layers, these quality layers are embedded in the code-stream without possibility of further modifications. This causes that, once the code-stream is already encoded, it is not possible to identify which packets within $\mathcal{T}^l$ are better than others, in terms of rate-distortion. Besides, it is also not possible to identify which segments of each packet could be truncated to modify the bit-rate allocation or the number of quality layers. Quality layers are arguably fixed structures that do not allow any modification, causing that the quality scalability of the code-stream may only be set at encoding time.

The inalterability of quality layers might cause two drawbacks. The first one is the lack of quality scalability and quality progression of code-streams that contain a single or few quality layers. The second drawback is the lack of a fine quality scalability for Window Of Interests (WOIs), a mechanism widely used in interactive image transmissions. Let us explain each one of these drawbacks further.

In Section 3.2 we have seen that a code-stream $\mathcal{X}$ with $N$ quality layers allocated at bit-rates $\{R^0, R^1, R^2, ..., R^{N-1}\}$ can only assure the best recovery, in terms of distortion, when the code-stream is decoded at these bit-rates. If we needed to construct a new code-stream, say $\mathcal{X}'$, from $\mathcal{X}$ at a target bit-rate $R'$, and this target bit-rate does not coincide with any quality layers boundary, i.e. $R^i < R' < R^{i+1}$, the constructed code-stream would have a larger distortion than the one obtained if $R'$ coincides with some $R^i$. The penalization in this distortion can not be estimated a priori and depends on how far the bit-rate $R'$ is from the quality layer boundaries. More precisely, the distortion of $\mathcal{X}'$ is increased when $J$ is increased, being $J = min(R' - R^i, R^{i+1} - R')$. However, this should not be a shortcoming since a good strategy of quality layers allocation guarantees a very low penalization on the coding performance. The selection of a good strategy of quality layers allocation is important because the quality scalability and the coding performance of the code-stream is directly related with it. The worst case is reached with code-streams containing a single quality layer; in these cases the coding performance penalization is meaningful, as we have shown in the evaluation of Graphic 3.1 –page 76–.

**Remark 4.1.1** *The truncation and decoding of a code-stream containing 20 quality layers equivalently spaced in terms of bit-rate at 0.5 bps recovers the original image 0.221 dB worse than when the optimal PCRD method encodes the image at 0.5 bps. At 0.6 bps (quality layer boundary) it is only 0.094 dB worse. However, if the original code-stream contains a single quality layer, at 0.5 bps the recovered image is 7.49 dB worse than when* *encoding at the same bit-rate with the PCRD method. See, in Figure 4.1, the same region of the Cafeteria image when 0.5 bps are decoded from code-streams containing a single or 20 quality layers logarithmically spaced in terms of bit-rate.*

*Single quality layer (19.839 dB)* *20 quality layers (27.108 dB)*

Figure 4.1: *Visual comparison of an image recovered from code-streams containing a single or several quality layers.*

Unfortunately, most JPEG2000 coders constructs single quality layers code-streams by default. And, even if the construction of code-streams containing several quality layers is the default mode, we do not know any implementation that uses the best allocation strategy introduced in [107]. This should not be a problem if the code-stream does not need to be truncated or interactively transmitted; otherwise, code-streams containing a single or few quality layers have a very poor coding performance. For institutions, such as hospitals or remote sensing centers, which manage millions of images each year, the construction of code-streams containing a single or few quality layers might become a meaningful issue.

This first drawback is not caused by the standard itself but by an inadequate use of the JPEG2000 standard, which should be avoided constructing code-streams containing a reasonable number of quality layers using some good allocation strategy. However, the second drawback is intrinsically due to the JPEG2000 standard itself. Quality layers provide quality scalability for the whole image, i.e. they assure the best recovery at the bit-rates boundaries if the complete spatial area of the image is decoded. However, in interactive image transmissions, the decoding of the complete image area is not usual, and most applications decode (or transmit) just the region, or WOI, required by the client. Besides, the delivery of the WOI should allow a transmission in such a way the quality is improved increasingly. The question here is what happens when recovering a WOI from an already encoded code-stream at different bit-rates.

The decoding (or transmission) of a WOI just processes those code-blocks belonging to the desired spatial area. In order to improve the quality of the WOI increasingly, we may think that the best option is to decode the WOI at the quality layers boundaries. More precisely, the WOI would be decoded at bit-rates $O^0, O^1, O^2, ..., O^N$ if $O^l$ denotes the bit-rate of the code-stream segments belonging to the WOI code-blocks allocated at quality layer $l$. Note that $O^l \leq R^l$ and only when the WOI is the complete image area, $O^l = R^l$. Although the quality layers bit-rates $\{R^0, R^1, R^2, ..., R^{N-1}\}$ are set by the rate control method following some allocation strategy, $\{O^0, O^1, O^2, ..., O^{N-1}\}$ might have an inadequate allocation that does not supply a suitable decoding or transmission. Obviously, the allocation of WOIs can not be contemplated by the rate control method, which considers the complete image area when constructing quality layers and, therefore, the allocation of WOIs can become inappropriate. The impossibility to control the distortion once the code-stream is constructed might cause a poor coding performance when transmitting WOIs. This is the second drawback pointed out in this section, and it was already described by Taubman in 2003 [98, 99] in the scenario of interactive image transmissions using the JPIP protocol defined in the Part 9 of JPEG2000 [46].

One may expect that the solution to the first drawback could be to re-encode the image, i.e. decode the single quality layer code-stream $\mathcal{X}$ and construct a new code-stream $\mathcal{X}'$ containing several quality layers. However, note that when using lossy compression, $\mathcal{X}$ is not the original image and has a distortion compared to the original image, which we denote as $D_{\mathcal{X}}$. This distortion is increased in each re-encoding, i.e. $D_{\mathcal{X}} < D_{\mathcal{X}'} < D_{\mathcal{X}''}$. This also causes that the quality layers constructed from the re-encoding of $\mathcal{X}$ are not the same as when the original image is available. With lossless compression this problem does not come up.

**Remark 4.1.2** *When the image Candle is encoded to a single quality layer code-stream, the bit-rate of the code-stream is 5.508 bps and has a PSNR of 56.467 dB compared to the original image. When decoding this single quality layer code-stream and encoding it again to a single quality layer code-stream with the same coding parameters, its bit-rate is 5.504 bps and has a PSNR of 55.628 dB compared to the original image. The coding performance penalization of the re-encoding is 0.839 dB. However, at low bit-rates this penalization is smaller.*

Another problem related with the re-encoding of code-streams is the computational complexity that it may entail. Imagine, for example, a hospital which generates about 3 GB of images each day. If we consider that the re-encoding of 10 MB of code-stream takes 1 second, the re-encoding of all the images produced in one year would last more than 75 days!

While there does not exist a solution to the first drawback, Taubman et al. proposed a solution to the second drawback. In [98, 99] it is showed the poor coding performance of WOI transmissions when using quality layers and it is proposed a re-sequencing method which increases the number of quality layers transmitted for the desired WOI. The main idea behind this re-sequencing technique is the consideration of the subband weights and the coefficients affected by the WOI in order to send first those packets that better recover the WOI. This method achieves improvements as large as 8 dB when compared to the transmission of the WOI at quality layers boundaries! However, the method is focused on the transmission of a WOI using the packets already constructed by the coder. If the code-stream contains a single or few quality layers, this re-sequencing method would not improve the coding performance. The only way to improve the coding performance with code-streams containing a single or few quality layers is the division of packets with some rate control method able to model the rate-distortion without using distortion measures based on the original image.

All the rate control methods reviewed in Chapter 3 are devised for the coder. None of them consider the drawbacks explained in this chapter. However, we have evaluated if some of them could be used in the decoder side in order to settle these drawbacks. Column ”orig” of Table 3.2 –page 68– considers whether the rate control method needs rate-distortion measures based on the original image or not. Those ones that do not need them are classified in two categories: the methods that need information of the coding process (such as the number of significant and refinement coefficients,. . . ), marked as ”dec”, and the ones that do not need them, marked as ”no”. Qin [75] and Vikram [103] are the single methods of the former category, and they might address the problem of single quality layers code-streams embedding quality layers without penalizing the coding performance. However, these rate control methods are clearly useless for the transmission of WOIs because they compel to decode segments of the code-stream, which would widely decrease the computational performance of the server in an interactive transmission. There is a single rate control method of the latter category: Masuzaki [65]. Since it does not need to decode the code-stream, it could be effectively used to address both drawbacks presented in this section. However, the problem of this method is that it obtains a poor coding performance (more than 1 dB worse than the PCRD method).

Obviously, without using distortion measures based on the original image, the coding performance of the rate control method can not be as good as the obtained with the reviewed rate control methods. However the purpose of this thesis is to settle both drawbacks without using distortion measures based on the original image. The requirements that we set out for the new rate control method are the following:

1. *Use in the decoder side*: the developed rate control method must provide quality scalability once the code-stream is already encoded, in spite of the number and bit-rate of the quality layers it contains. The coding performance must be as optimal as possible, at least improving the results obtained by Masuzaki [65]. Besides, the computational complexity of the method should be negligible in order to allow the control of interactive image transmissions.

2. *No distortion measures based on the original image*: it must not need to decode any segment of the code-stream. It is understood that the main header and/or packet

headers can be decoded in order to know the image geometry.

3. *Use in the coder side*: although the rate control is devised for the decoder side, its use in the coder side should also be allowed to reduce the computational complexity of the coder, as most rate control methods also do.

4. *JPEG2000 compliant*: the code-streams constructed by the rate control method must be JPEG2000 compliant.

# 4.2 Development of the CPI rate control method

## 4.2.1 Motivation

Recall from Section 2.2.4 that the fractional bit plane coder of JPEG2000 encodes each bit-plane of the code-block $\mathcal{B}_i$ with three coding passes: the Significance Propagation Pass (SPP), referred to as $\mathcal{P}_i^{(p,2)}$, the Magnitude Refinement Pass (MRP), referred to as $\mathcal{P}_i^{(p,1)}$, and the Cleanup Pass (CP), referred to as $\mathcal{P}_i^{(p,0)}$, where $p$ stands for the bit-plane, with the lowest bit-plane $p = 0$. We slightly change the notation of coding passes to $\mathcal{P}_i^c$, where

$$c = (p \cdot 3) + cp$$

and $cp$ stands for the coding pass number, i.e. $cp = \{2 \text{ for } SPP,\ 1 \text{ for } MRP,\ 0 \text{ for } CP\}$. Note that $c$ identifies a bit-plane and coding pass unequivocally, so we name it *coding level*. $C_i$ stands for the highest coding level of the code-block $\mathcal{B}_i$, while the highest and lowest coding level of the image is referred to as $C_{max} = max(C_i)$ and $C_{min} = 0$ respectively.

The embedded code-stream produced by the fractional bit plane coder can be truncated at the end of each coding pass but, as we have seen in Section 3.1.2, the optimal PCRD method applies a convex hull approach to find out the feasible truncation points, i.e. those points which lie on the convex hull. Therefore, the PCRD method may join, for example, the coding passes $\mathcal{P}_i^k, \mathcal{P}_i^{k-1}$. We name the set of consecutive coding passes joined by the PCRD as *coding segments*. Each coding segment must contain, at least, one coding pass.

The rate control method introduced in this chapter has been developed assuming the following two premises:

- *Inclusion of a single or few coding passes at each coding segment*: we assume that almost all the coding passes are feasible truncation points, i.e. most coding segments only contains a single coding pass. In order to prove the reliability of this assumption, Table 4.1 shows the number of coding passes included in each coding segment when encoding several images. Notice than, in average, more than 97% of the coding segments contain only one or two coding passes. This kind of evidence is also given in [114]. In addition to this evidence, it is worth noting that in MEPRD [49] it is stated that the coding performance is not affected even if PCRD selects coding passes not lying on the convex hull.

Table 4.1: *Number of coding passes included in each coding segment.*

| | *# coding passes of each coding segment* | | | |
|---|---|---|---|---|
| *IMAGE* | 1 | 2 | 3 | $\geq 4$ |
| Portrait | 7092 | 2273 | 192 | 15 |
| Cafeteria | 9118 | 2377 | 218 | 10 |
| Fruit Basket | 6030 | 2196 | 240 | 6 |
| Wine and Tableware | 7159 | 2239 | 305 | 14 |
| Bicycle | 8917 | 2197 | 161 | 12 |
| Orchid | 5684 | 2231 | 170 | 44 |
| Musicians | 8070 | 2325 | 465 | 5 |
| Candle | 10095 | 2379 | 230 | 6 |
| *percentage* | 75.38 % | 22.09 % | 2.4 % | 0.13 % |

RESULTS: *average of corpus ISO 12640-1, gray scaled, size 1536x1920.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, RESTART coding variation. Lossy compression.*
IMPLEMENTATION: *BOI v1.0*

- *Dependency of the coding pass inclusion on the coding level*: we assume that a coding pass is more likely to be included in the final code-stream if it belongs to a higher

Table 4.2: *Number of coding passes included in each coding level by the PCRD method at different bit-rates.*

| *cod. level* | *#CoP* | 0.0625 bps | 0.125 bps | 0.25 bps | 0.5 bps | 1 bps | 2 bps |
|---|---|---|---|---|---|---|---|
| ... | | ... | | | | | |
| 25 (MRP) | 53 | 53 | 53 | 53 | 53 | 53 | 53 |
| 24 (CP) | 170 | **149** | 170 | 170 | 170 | 170 | 170 |
| 23 (SPP) | 170 | **143** | 170 | 170 | 170 | 170 | 170 |
| 22 (MRP) | 170 | **93** | **166** | 170 | 170 | 170 | 170 |
| 21 (CP) | 418 | **74** | **285** | 418 | 418 | 418 | 418 |
| 20 (SPP) | 418 | **16** | **185** | **412** | 418 | 418 | 418 |
| 19 (MRP) | 418 | **9** | **92** | **400** | 418 | 418 | 418 |
| 18 (CP) | 604 | | **42** | **452** | **598** | 604 | 604 |
| 17 (SPP) | 604 | | | **121** | **593** | 604 | 604 |
| 16 (MRP) | 604 | | | **11** | **578** | 604 | 604 |
| 15 (CP) | 679 | | | **3** | **575** | 679 | 679 |
| 14 (SPP) | 679 | | | | **125** | **672** | 679 |
| 13 (MRP) | 679 | | | | | **668** | 679 |
| 12 (CP) | 720 | | | | | **625** | 720 |
| 11 (SPP) | 720 | | | | | **249** | 720 |
| 10 (MRP) | 720 | | | | | | 720 |
| 9 (CP) | 735 | | | | | | **728** |
| 8 (SPP) | 735 | | | | | | **712** |
| 7 (MRP) | 735 | | | | | | **172** |
| 6 (CP) | 736 | | | | | | **37** |
| ... | | ... | | | | | |

LEGEND: *#CoP - total number of coding passes of the coding level*
RESULTS: *Bicycle image of the corpus ISO 12640-1, gray scaled, size 1536x1920.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, 1 quality layer. Lossy compression.*
IMPLEMENTATION: *BOI v1.0*

coding level. In the framework of JPEG2000, this kind of assumption is also set in PSRA [110]. Table 4.2 shows the coding passes included at each coding level when encoding the Bicycle image at different bit-rates using the PCRD method. The number of coding passes is emphasized in bold font when not all of them are included for that coding level. Note that only the two or three lowest coding levels include less than 90% of coding passes, apart from the lowest bit-rates that, in some cases, have four coding levels including less than 90% of coding passes.

### 4.2.2 Algorithm

The two premises above suggest that a classical rate-distortion model might be suitable to control the rate-distortion. This classical rate-distortion model is applied here through a rather simple interleaving algorithm. This algorithm encodes the coding passes of code-blocks belonging to the same coding level, beginning from the highest coding level $C_{max}$ to the lowest coding level $C_{min}$ until the target bit-rate is achieved. In each coding level, the coding passes are encoded from those code-blocks belonging the lowest resolution level, referred to as $\mathcal{L}_0$, to the highest resolution level, referred to as $\mathcal{L}_L$. The set of subbands belonging to the resolution level $\mathcal{L}_j$ are denoted as $b_{j,s}$, where $s$ stands for the subband. The interleaving algorithm follows the order $s = [HL, LH, HH]$ or just $s = [LL]$ when $j = 0$. When a target bit-rate has to be attained, this algorithm is formulated in Table 4.3 for the encoding process. We name it Coding Passes Interleaving (CPI). Figure 4.2 depicts how the interleaving is carried out in a single image component with 2 resolution levels; the dash-dotted line indicates the processing order followed by CPI.

Table 4.3: *The Coding Passes Interleaving (CPI) algorithm.*

**set** *bitRate* ← *0*
**for each** *coding level* $c$ **from** $C_{max}$ **to** $C_{min}$ **do**
  **for each** *resolution level* $\mathcal{L}_j$ **from** $j = 0$ **to** $L$ **do**
    **set** $sOrder \leftarrow \begin{cases} [LL] & \text{if } j = 0 \\ [HL, LH, HH] & \text{if } j > 0 \end{cases}$
    **for each** *subband* $b_{j,s}$ **with** $s = sOrder$ **do**
      **for each** *code-block* $\mathcal{B}_i \in b_{j,s}$ **do**
        *ENCODE coding pass* $\mathcal{P}_i^c$
        **set** *bitRate* ← *bitRate* + *length*($\mathcal{P}_i^c$)
        **if** *bitRate* ≥ *targetBitRate* **then**
          *STOP encoding*
        **endif**
      **endfor**
    **endfor**
  **endfor**
**endfor**

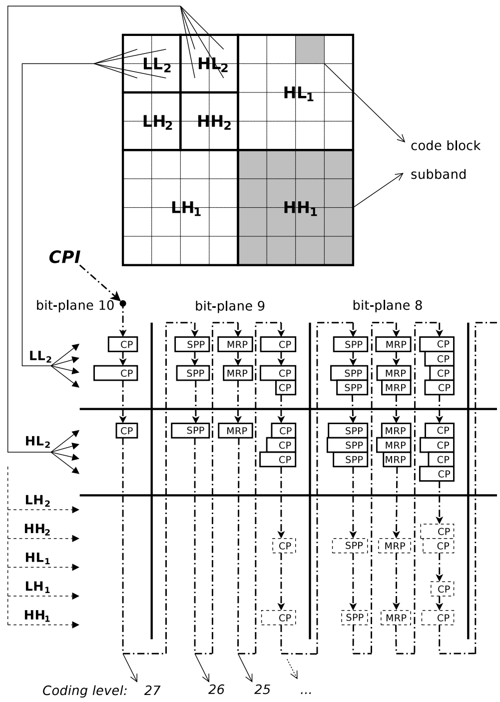



Figure 4.2: *The processing order followed by the CPI algorithm.*

Implementations should take into account that JPEG2000 headers requires some bytes and that the last coding pass must be discarded to do not exceed the target bit-rate. The simple interleaving algorithm CPI does not use any distortion measure based on the original image and it fulfills all the requirements set in Section 4.1. With minor modifications, CPI can be used in the decoder side in order to extract WOIs, or the complete image, at different bit-rates, even if the code-stream contains a single quality layer. Besides, CPI can also construct or re-allocate quality layers within a code-stream. The coding performance obtained when applied in the coder or decoder is exactly the same.

It is worth noting that this method benefits from the weighting performed by the quantization stage. Recall that when using the irreversible wavelet transform, the quantization stage usually uses the L2norm to weight each subband in order to achieve orthonormality. Thanks to this precise weighting and to the adequate order of the coding passes, the CPI algorithm follows a scanning order which selects first those coding passes that have the largest rate-distortion contributions.

The spread belief that the coding passes situated at high bit-planes recover the original images better than those situated at low bit-planes, as well as the scanning order followed by CPI, are two techniques derived from a rate-distortion model commonly used in image compression and in rate control. Actually, during the technical discussion held within the JPEG2000 committee, prior to the final adoption of EBCOT, some versions of the Verification Model considered a related approach [62]. Also, in [100, Chapter 8.4.1] it is pointed out that an interleaving approach might also be appropriate for JPEG2000. On the other hand, the rate control method PSRA [110] also uses a very similar scanning order, although in its last step it uses distortion measures based on the original image.

Although when CPI is applied in the coder it reduces the computational complexity just encoding those coding passes included in the final code-stream, in the coder side CPI has the disadvantage that it must maintain all the wavelet coefficients of the image along with some information of the MQ-coder in memory while the encoding process is performed. This information must be kept in memory to be able to stop and restart the encoding of coding passes belonging to each code-block. This is a clear disadvantage with respect to those methods that do not perform an incremental encoding, since with the combination of line-based processing techniques they can reduce the memory requirements to the bit-rate of the constructed code-stream. This problem does not come up in the decoding side.

A key point of the CPI algorithm is the availability to identify the bit-rate of each coding pass. In the coder, these bit-rates are easily identifiable but, once the code-stream is constructed, the bit-rate of each coding pass is not stored within the code-stream and it can not be known without decoding it. This may prevent the use of CPI but, fortunately, this is easily settle down using the RESTART coding variation, which entails to store the coding passes bit-rate in the packet headers, slightly penalizing the coding performance (code-streams with the RESTART coding variation are about 1% larger than when not using this coding variation). The decoding of the packet headings has negligible costs in terms of memory consumption and computational complexity, thus the CPI has also negligible computational costs in the decoder side.

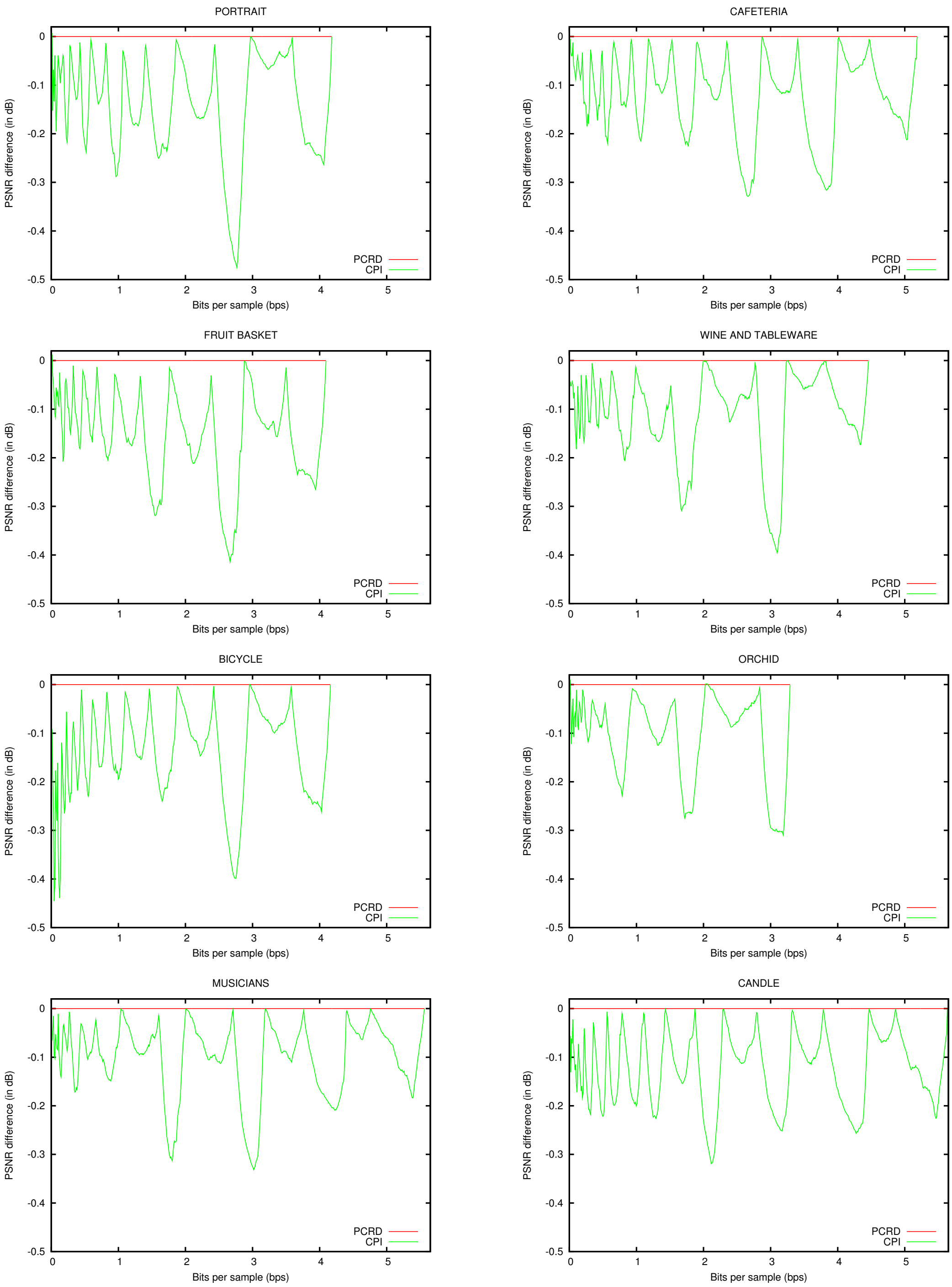
## 4.3 Experimental results

As well as in Section 3.2.3, we use the coding performance obtained by the optimal PCRD method in order to set the maximum threshold, computing the PSNR difference between PCRD and CPI. The proposed CPI can be applied to encode any image at a target bit-rate or to extract/decode a segment of a code-stream, even if it only contains a single or few quality layers. The constructed or extracted code-stream is the same in both cases, thus both results are drawn in a single line in the graphics below.

In all experiments, Kakadu has been used to construct the code-stream with the optimal PCRD method, and the CPI method has been implemented in BOI. The parameters of both implementations are set equally, however the quantization weights used in Kakadu are not exactly the L2norms, so we use a derived quantization to equal the subband weights of both applications. Besides, we use the RESTART coding variation to identify the coding passes bit-rates within the code-stream without needing to decode any segment when using the CPI algorithm.

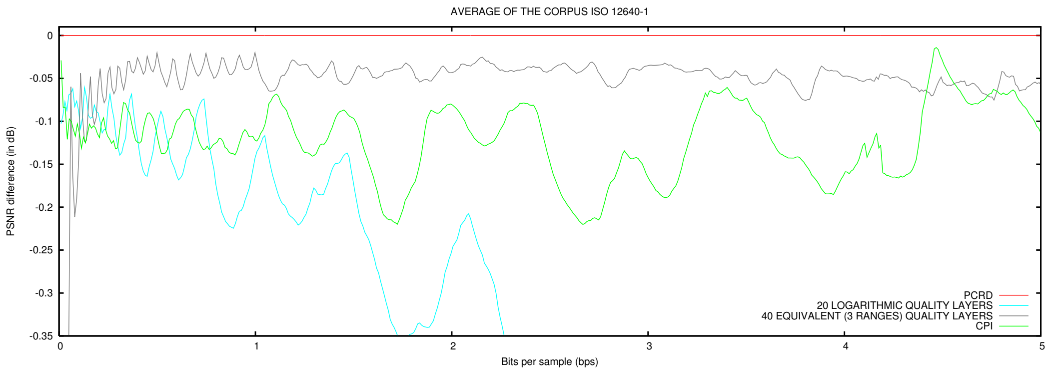
In this section we just present the results obtained by CPI in lossy mode for the images of the corpus ISO 12640-1. In the experimental section of the last rate control method introduced in this thesis (Section 6.4), extensive experimental results are presented, comparing the three rate control methods proposed.

Graphic 4.1 depicts the coding performance of CPI for the images of the corpus ISO 12640-1. We show the eight images in single graphics to better appreciate that CPI obtains a coding performance that fluctuates continuously in all the analyzed images. Graphic 4.2 depicts the average of the eight images, comparing the CPI coding performance against the results obtained in Section 3.2.3 with the use of quality layers (only the best allocation strategies are plotted).




RESULTS: *images of the corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, 1 quality layer, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD - Kakadu v4.5, CPI - BOI v1.2*

Graphic 4.1: *Coding performance evaluation of CPI compared to PCRD.*




*Av. PSNR diff. - CPI: -0.113 dB (0.001 to 1 bps) -0.122 dB (total)*
*Av. PSNR diff. - 20 logarithmic quality layers: -0.129 dB (0.001 to 1 bps) -0.638 dB (total)*
*Av. PSNR diff. - 40 equivalent (3 ranges) quality layers: -0.093 dB (0.001 to 1 bps) -0.058 dB (total)*

---

RESULTS: *average of corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*

JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*

IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI - BOI v1.2*

---

Graphic 4.2: *Coding performance evaluation of CPI and the best strategies of quality layers allocation compared to PCRD.*

# Chapter 5

# Optimized interleaving-based rate control method

This chapter presents the rate control method Reverse subband scanning Order and coding passes Concatenation (ROC), which is devised with the aim to improve the coding performance of CPI and based on simple modifications of the CPI algorithm. The development of ROC has involved a study of the rate control performed by CPI, which is explained in the first section. The second section describes the new rate control method ROC and the third section assess its coding performance.

This research has been accepted for publication to the IEEE Signal Processing Letters journal [11].

## 5.1 Analysis of the CPI method

The scanning order followed by CPI is also used in other coding systems that achieve a regular coding performance among all bit-rates. Therefore, it might be expected that CPI should also obtain a coding performance similar to that of the optimal PCRD method, but it does not. As we can see in the experimental results of Section 4.3, the coding performance of CPI is not well balanced and, in some cases, is 0.5 dB worse than the optimal PCRD method. This fact arises two questions: when these differences occur and why they are produced.

The first question can be readily answered analyzing the graphics that compare CPI with the PCRD method. Note that in all graphics the coding performance of CPI exhibits a similar behavior among all bit-rates: it fluctuates continuously from 0.001 dB to 0.5 dB worse than the optimal PCRD method. A detailed analysis on these bit-rates discloses that the best coding performance, which is the same as the optimal PCRD method, is obtained when CPI ends the scanning of a coding level that contains coding passes of type SPP, or when CPI ends the scanning of a coding level that contains coding passes of type CP. Graphic 5.1 details the coding performance achieved by CPI when encoding the Cafeteria image, depicting the coding levels included in several bit-rate ranges. Notice that, as reported, the coding performance of CPI and PCRD coincides at the end of every coding pass of type SPP or CP. These coincidences are also observed in all the other images analyzed in Section 4.3. Although using a very similar scanning order, it is surprising that these kinds of fluctuations are not reported in PSRA [110].

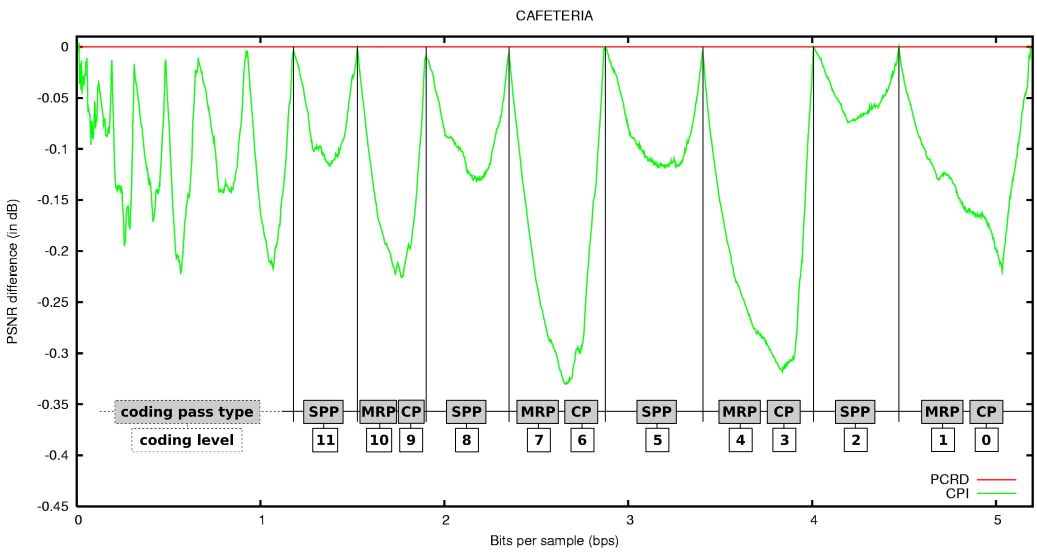
Identifying when the differences occur gives us the clue to answer why they are produced. It is clear that we must focus our attention between the bit-rate intervals where CPI decreases its coding performance. We have developed several techniques trying to disclose the cause of this coding performance drop. Two of them give us a good explanation of what is happening between these bit-rate intervals.

The first technique is the tracking of the optimal PCRD method in the bit-rate intervals where the CPI coding performance drops. More precisely, if $R^i, R^{i+1}$ denotes two consecutive bit-rates where CPI and the PCRD method obtain the same coding performance, we divide the bit-rate interval $(R^i, R^{i+1}]$ in $t$ sub-intervals, denoted as

$$R_1^i = (R^i, R^i + S], R_2^i = (R^i + S, R^i + 2 \cdot S], ..., R_t^i = (R^i + (t-1) * S, R^i + t \cdot S]$$

where

$$S = \frac{R^{i+1} - R^i}{t} \ .$$




RESULTS: *Cafeteria image of the corpus ISO 12640-1, gray scaled, size 2048x2560. 2000 control points.*

JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, 1 quality layer, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*

IMPLEMENTATIONS: *PCRD - Kakadu v4.5, CPI - BOI v1.2*

Graphic 5.1: *Detailed coding performance evaluation of CPI compared to PCRD.*

We analyze the coding passes included by the optimal PCRD method at bit-rate $R_j^i$ that are not included in $R_{j-1}^i$ consecutively in order to disclose the scanning order followed by the PCRD method, if any. One may expect that the PCRD method does not follow any scanning order, selecting coding passes of all the subbands and resolutions levels indistinctly, just depending on the rate-distortion slope of the feasible truncation points. To our pleasant surprise, this is not completely true.

Instead of starting the selection of those coding passes situated at the lowest resolution levels, the tracking of the PCRD method discloses that, at medium and high bit-rates, PCRD selects coding passes belonging to the highest resolution levels first. Besides, when observing the intervals where CPI scans coding levels with coding passes of type MRP and

CP, we notice that the PCRD method usually includes coding passes of type CP before completing the inclusion of the coding level with coding passes of type MRP. These observations indicate that the rate-distortion contributions are not distributed as the scanning order of CPI follows. This is addressed more precisely with the second technique developed to understand the drops of the CPI coding performance.

We have studied the results obtained with the tracking of the PCRD method from several points of view. The approach that gives the best explanation is to evaluate each resolution and coding level between the bit-rates where the coding performance of CPI and the optimal PCRD method coincide using the MSE. $D_i^c$ stands for the distortion of code-block $\mathcal{B}_i$ at the coding level $c$, and is computed as

$$D_i^c = w_{b_i}^2 \sum_{k \in \mathcal{B}_i} (y_i[k] - \hat{y}_i^c[k])^2$$

where $w_{b_i}$ is the weight of the subband $b_i$ to which code-block $\mathcal{B}_i$ belongs, $y_i[k]$ are the original samples of the code-block $B_i$, and $\hat{y}_i^c[k]$ are the samples quantized at coding level $c$. Let $\triangle D_i^c = D_i^{c+1} - D_i^c$ be the distortion contribution of code-block $\mathcal{B}_i$ at coding level $c$; let $\triangle D^c = D^{c+1} - D^c$ be the sum of the distortion contributions of all code-blocks of the image at the coding level $c$; and let $\triangle D_{\mathcal{L}_j}^c = D_{\mathcal{L}_j}^{c+1} - D_{\mathcal{L}_j}^c$ be the sum of the distortion contributions of all code-blocks belonging to the resolution level $\mathcal{L}_j$ at the coding level $c$.

We consider the distortion contribution of each resolution level in percentage between the bit-rates at which the coding performance of both methods coincides. This is, considering the scanning of a coding level with coding passes of type SPP or considering two consecutive coding levels with coding passes of type MRP and CP. If $\triangle D^{c+1,c} = \triangle D^{c+1} + \triangle D^c$, the following expression summarizes this percentage computation

$$C_{\mathcal{L}_j}^c = \begin{cases} \dfrac{\triangle D_{\mathcal{L}_j}^c}{\triangle D^c} & \text{if } c \; mod \; 3 = 2 \quad \text{(SPP coding pass)} \\[2ex] \dfrac{\triangle D_{\mathcal{L}_j}^c}{\triangle D^{c,c-1}} & \text{if } c \; mod \; 3 = 1 \quad \text{(MRP coding pass)} \\[2ex] \dfrac{\triangle D_{\mathcal{L}_j}^c}{\triangle D^{c+1,c}} & \text{if } c \; mod \; 3 = 0 \quad \text{(CP coding pass)} \end{cases} .$$

Table 5.1 shows the percentages $C^c_{\mathcal{L}_j}$ for the Cafeteria image. The resolution levels that have the greatest contributions are emphasized in bold font, and the right most column shows the bit-rate achieved at the end of each coding level.

Table 5.1: *Percentage of the distortion contribution of each coding and resolution level.*

| *cod. level* | $\mathcal{L}_0$ | $\mathcal{L}_1$ | $\mathcal{L}_2$ | $\mathcal{L}_3$ | $\mathcal{L}_4$ | $\mathcal{L}_5$ | *bit-rate* |
|---|---|---|---|---|---|---|---|
| 33 (CP) | **99.13 %** | 0.87 % | | | | | |
| 32 (SPP) | **96.59 %** | 3.41 % | | | | | |
| 31 (MRP) | **27.15 %** | 0.76 % | | | | | |
| 30 (CP) | 9.98 % | **52.74 %** | 9.31 % | | | | |
| 29 (SPP) | **50.36 %** | **39.80 %** | 9.84 % | | | | |
| 28 (MRP) | **12.62 %** | 2.98 % | 0.57 % | | | | |
| 27 (CP) | 0.14 % | **25.45 %** | **44.86 %** | **13.22 %** | 0.14 % | | |
| 26 (SPP) | 7.65 % | **33.10 %** | **42.50 %** | **16.29 %** | 0.47 % | | $<$ 0.001 bps |
| 25 (MRP) | 4.22 % | 2.63 % | 2.63 % | 1.00 % | 0.03 % | | 0.013 bps |
| 24 (CP) | 0.00 % | 4.64 % | **21.01 %** | **40.75 %** | **22.97 %** | 0.12 % | 0.015 bps |
| 23 (SPP) | 0.93 % | 9.70 % | **24.64 %** | **36.83 %** | **27.46 %** | 0.44 % | 0.03 bps |
| 22 (MRP) | 1.39 % | 1.58 % | 2.49 % | 2.89 % | 2.08 % | 0.02 % | 0.06 bps |
| 21 (CP) | 0.00 % | 0.36 % | 4.60 % | **21.82 %** | **48.78 %** | **13.99 %** | 0.11 bps |
| 20 (SPP) | 0.15 % | 2.37 % | 10.13 % | **26.51 %** | **43.56 %** | **17.28 %** | 0.18 bps |
| 19 (MRP) | 0.66 % | 1.10 % | 2.38 % | 4.25 % | 5.82 % | 1.89 % | 0.20 bps |
| 18 (CP) | 0.00 % | 0.04 % | 0.74 % | 7.04 % | **30.89 %** | **45.18 %** | 0.30 bps |
| 17 (SPP) | 0.03 % | 0.73 % | 3.97 % | 14.56 % | **39.42 %** | **41.30 %** | 0.46 bps |
| 16 (MRP) | 0.50 % | 1.01 % | 2.72 % | 6.38 % | **11.84 %** | 9.07 % | 0.51 bps |
| 15 (CP) | 0.00 % | 0.01 % | 0.18 % | 2.15 % | **14.77 %** | **51.37 %** | 0.64 bps |
| 14 (SPP) | 0.01 % | 0.31 % | 1.87 % | 8.72 % | **30.40 %** | **58.69 %** | 0.90 bps |
| 13 (MRP) | 0.37 % | 0.88 % | 2.76 % | 7.55 % | **17.66 %** | **21.24 %** | 1.02 bps |
| 12 (CP) | 0.00 % | .02 % | 0.08 % | 0.85 % | 7.26 % | **41.34 %** | 1.14 bps |
| 11 (SPP) | 0.00 % | 0.14 % | 1.04 % | 5.53 % | **23.76 %** | **69.53 %** | 1.49 bps |
| 10 (MRP) | 0.26 % | 0.68 % | 2.27 % | 6.96 % | **19.05 %** | **32.21 %** | 1.70 bps |
| 9 (CP) | 0.00 % | 0.01 % | 0.08 % | 0.54 % | 5.05 % | **32.88 %** | 1.85 bps |
| 8 (SPP) | 0.00 % | 0.09 % | 0.60 % | 3.76 % | **19.74 %** | **75.82 %** | 2.29 bps |
| 7 (MRP) | 0.18 % | 0.49 % | 1.77 % | 6.00 % | **18.62 %** | **41.07 %** | 2.63 bps |
| 6 (CP) | 0.00 % | 0.01 % | 0.11 % | 0.82 % | 4.58 % | **26.36 %** | 2.80 bps |
| 5 (SPP) | 0.00 % | 0.06 % | 0.44 % | 2.91 % | **16.83 %** | **79.76 %** | 3.34 bps |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 (MRP) | 0.15 % | 0.44 % | 1.60 % | 5.96 % | **20.82 %** | **57.42 %** | 3.84 bps |
| 3 (CP) | 0.00 % | 0.00 % | 0.01 % | 0.10 % | 0.88 % | **12.62 %** | 3.94 bps |
| 2 (SPP) | 0.00 % | 0.04 % | 0.28 % | 2.05 % | **12.94 %** | **84.69 %** | 4.41 bps |
| 1 (MRP) | 0.13 % | 0.39 % | 1.51 % | 5.77 % | **21.54 %** | **69.05 %** | 5.08 bps |
| 0 (CP) | 0.00 % | 0.00 % | 0.00 % | 0.00 % | 0.03 % | 1.58 % | 5.12 bps |

RESULTS: *Cafeteria image of the corpus ISO 12640-1, gray scaled, size 2048x2560.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, 1 quality layer, LRCP progression. Lossy compression.*
IMPLEMENTATION: *BOI v1.2*

Note that, within two consecutive coding levels containing coding passes of type MRP and CP, the CP has usually greater distortion contributions than the MRP (apart from the lowest coding levels). Note also that the distortion contribution is not well balanced among coding levels and resolution levels: at high coding levels, the resolution levels that have the greatest distortion contributions are the lowest ones, while at low coding levels, the highest resolution levels have the greatest distortion contributions. In order to see this unbalanced distribution better, Table 5.2 evaluates the distortion contribution in percentage per bit-planes and grouping the resolution levels $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ and the resolution levels $\mathcal{L}_4, \mathcal{L}_5$. In this Table, from bit-plane 6 onwards the greatest distortion contributions are situated at the highest resolution levels.

## 5.2 Development of the ROC rate control method

Based on the previous analysis and on practical experimentation, we propose minor modifications to the CPI algorithm to improve its coding performance. All the features of CPI are still maintained and therefore the new rate control method has exactly the same properties as CPI.

Tables 5.1 and 5.2 suggest that, at high coding levels, the scanning order followed by CPI is already correct. Therefore, from $C_{max}$ to $C_{med}$ (defined below) the new algorithm

Table 5.2: *Percentage of the distortion contribution of each coding level grouping resolution levels.*

| *Bit-plane* {*cod. levels*} | $\sum_{\mathcal{L}_0,\mathcal{L}_1,\mathcal{L}_2,\mathcal{L}_3}$ | $\sum_{\mathcal{L}_4,\mathcal{L}_5}$ | *bit-rate* |
|---|---|---|---|
| 11 $\{33\}$ | **100.00 %** | 0.00 % | |
| 10 $\{32, 31, 30\}$ | **100.00 %** | 0.00 % | |
| 9 $\{29, 28, 27\}$ | **99.92 %** | 0.08 % | |
| 8 $\{26, 25, 24\}$ | **85.11 %** | 14.89 % | 0.014 bps |
| 7 $\{23, 22, 21\}$ | **50.73 %** | 49.27 % | 0.11 bps |
| 6 $\{20, 19, 18\}$ | 28.31 % | **71.69 %** | 0.30 bps |
| 5 $\{17, 16, 15\}$ | 17.06 % | **82.94 %** | 0.64 bps |
| 4 $\{14, 13, 12\}$ | 11.40 % | **88.60 %** | 1.14 bps |
| 3 $\{11, 10, 9\}$ | 8.09 % | **91.91 %** | 1.85 bps |
| 2 $\{8, 7, 6\}$ | 6.35 % | **93.65 %** | 2.80 bps |
| 1 $\{5, 4, 3\}$ | 5.50 % | **94.50 %** | 3.94 bps |
| 0 $\{2, 1, 0\}$ | 6.32 % | **93.68 %** | 5.12 bps |

*Same coding parameters as in Table 5.1*

follows the same scanning order as CPI. With the aim to include first the code-blocks that have the greatest distortion contributions, from $C_{med}$ to $C_{min}$ the new algorithm scans the resolution levels from $\mathcal{L}_L$ to $\mathcal{L}_0$. In each resolution level, the code-blocks belonging to the $HH$ subband are encoded first and then the code-blocks belonging to the $\{LH, HL\}$ subbands are encoded in an interleaved way. When scanning a coding level with coding passes of type MRP and $c > 1$, the encoding of a MRP is concatenated by the CP of the same code-block. The resulting algorithm is named Reverse subband scanning Order and coding passes Concatenation (ROC), and is formulated in Table 5.3.

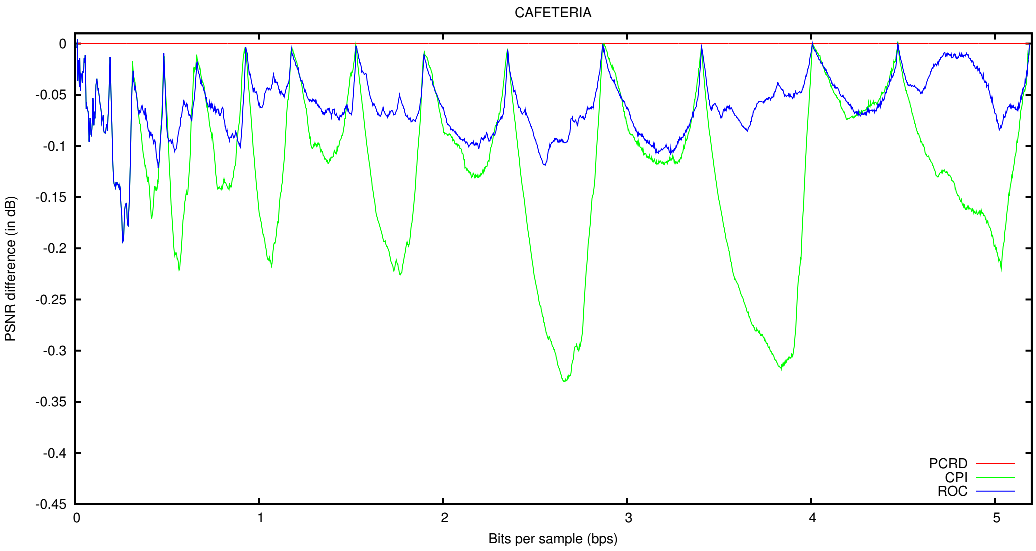
If we would consider the analysis of Table 5.2, $C_{med}$ would be set to the coding level where the resolution levels $\mathcal{L}_4, \mathcal{L}_5$ have the largest distortion contributions, i.e. at bit-plane 6, setting $C_{med} = 17$. In terms of bit-rate, this usually happens when reaching 0.3 or 0.4 bps. However, the distortion analysis of this Table is based on distortion measures based on the original image, which are not available once the code-stream has been constructed.

Therefore we must use some other measure. Experimental evidence suggests that a suitable measure is the number of code-blocks that belong to the highest resolution level and have to be encoded in the current coding level: $C_{med}$ is set to that coding level which encodes, at least, 55% of the code-blocks belonging to $\mathcal{L}_L$.

Table 5.3: *The Reverse subband scanning Order and coding passes Concatenation (ROC) algorithm.*

**set** *bitRate* ← *0*
**for each** *coding level* $c$ **from** $C_{max}$ **to** $C_{med}$ **do**
  follow the CPI algorithm
**endfor**
**while** *(*$c \geq 0$*)* **do**
 **for each** *resolution level* $\mathcal{L}_j$ **from** $j = L$ **to** $0$ **do**
  **set** $sOrder = \begin{cases} [LL] & \text{if } j = 0 \\ [HH, \{HL, LH\}] & \text{if } j > 0 \end{cases}$
  **for each** *subband* $b_{j,s}$ **with** $s = sOrder$ **do**
   **for each** *code-block* $\mathcal{B}_i \in b_{j,s}$ **do**
    **if** *(c mod* $3 = 2$ *OR* $c = 1, 0$*)* **then**
     *ENCODE coding pass* $\mathcal{P}_i^c$
     **set** *bitRate* ← *bitRate* + *length(*$\mathcal{P}_i^c$*)*
     **set** *c* ← *c - 1*
    **else**
     *ENCODE coding pass* $\mathcal{P}_i^c, \mathcal{P}_i^{c-1}$
     **set** *bitRate* ← *bitRate* + *length(*$\mathcal{P}_i^c$*)* + *length(*$\mathcal{P}_i^{c-1}$*)*
     **set** *c* ← *c - 2*
    **endif**
    **if** *bitRate* ≥ *targetBitRate* **then**
     *STOP encoding*
    **endif**
   **endfor**
  **endfor**
 **endfor**
**endwhile**

As we can see in Graphic 5.2, the coding performance of CPI is highly improved when applying the modifications of ROC. In order to evaluate the coding performance gain that



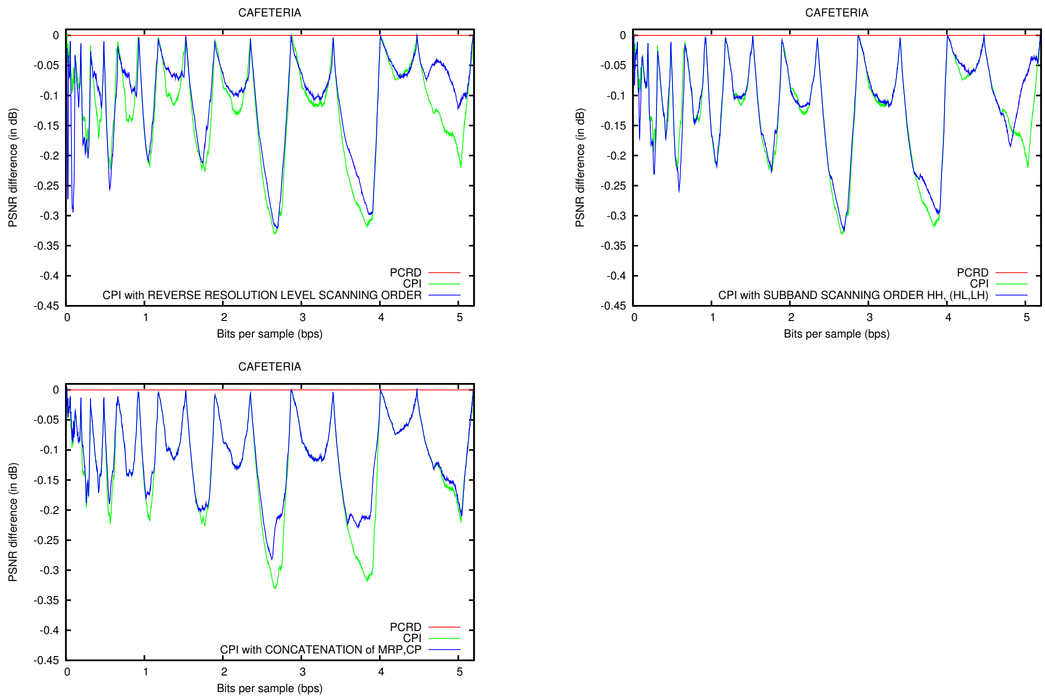

*Same coding parameters as in Graphic 5.1*

Graphic 5.2: *Detailed coding performance evaluation of CPI and ROC compared to PCRD.*

each one of these modifications supply, Graphic 5.3 depicts the coding performance obtained when applying each modification separately and along the complete bit-rate range, i.e. without applying the CPI at low bit-rates. Note that these modifications applied separately slightly improve the coding performance.

## 5.3 Experimental results

This experimental section shows the same experiments as in Section 4.3, comparing ROC to CPI and to the use of quality layers. Graphic 5.4 depicts the coding performance obtained by ROC and CPI for the images of the corpus ISO 12640-1. ROC almost always improves the coding performance of CPI and, in average (see Graphic 5.5), ROC enhances in a 40% the coding performance of CPI. Unfortunately, in some cases ROC not only does not improve the coding perfromance of CPI, but it also decreases it. The worst case is the

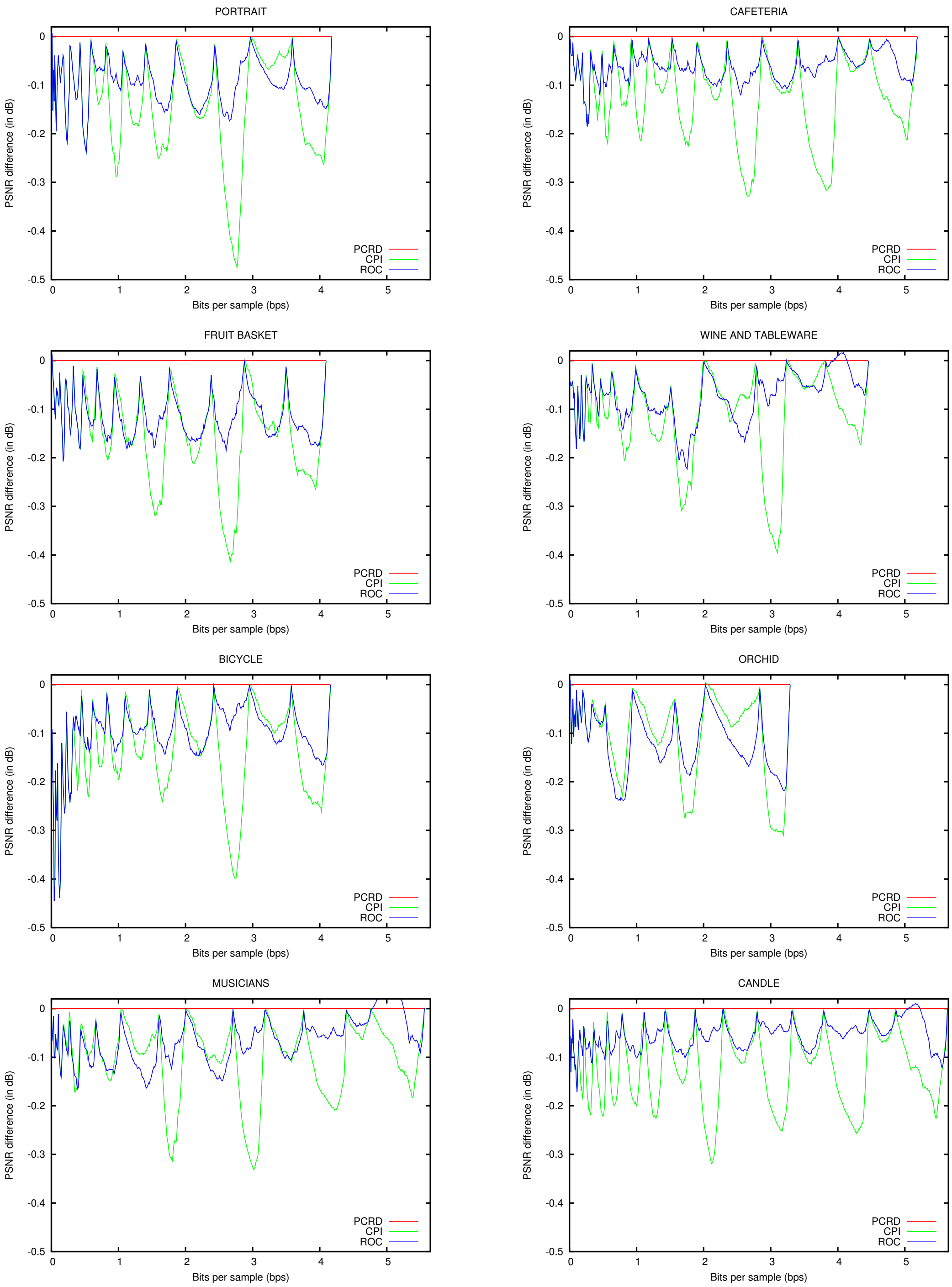



*Same coding parameters as in Graphic 5.1*

Graphic 5.3: *Detailed coding performance evaluation of CPI and three CPI modifications, compared to PCRD.*

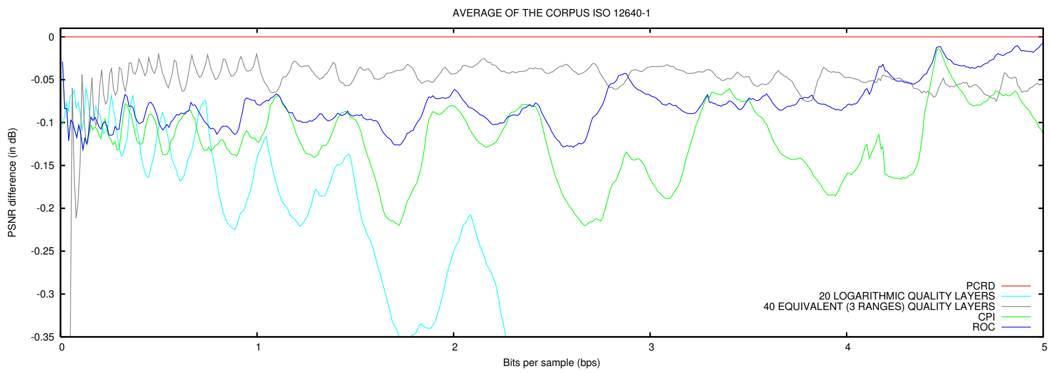
Orchid image: at 2.66 bps ROC is 0.12 dB worse than CPI.

In these graphics we also appreciate a curious result. In some images, for instance in Musicians or Candle, and for very low compression factors, the coding performance of ROC improves the coding performance achieved by the optimal PCRD method. This only happens when ROC encodes the coding passes of the lowest coding levels, i.e. $c = 1$ or $c = 0$ and we think that it may be caused by the bias reconstruction parameter of the dequantization process (see Section 2.2.6 –page 34–), which can produce small variations on the reconstructed image. When using a reconstruction parameter of $\gamma = \frac{1}{2}$, these curious results do not appear and PCRD always obtains better results than ROC.




RESULTS: *images of the corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, 1 quality layer, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD - Kakadu v4.5, CPI/ROC - BOI v1.2*

Graphic 5.4: *Coding performance evaluation of CPI and ROC compared to PCRD.*




*Av. PSNR diff. - CPI: -0.113 dB (0.001 to 1 bps) -0.122 dB (total)*
*Av. PSNR diff. - ROC: -0.096 dB (0.001 to 1 bps) -0.074 dB (total)*
*Av. PSNR diff. - 20 logarithmic quality layers: -0.129 dB (0.001 to 1 bps) -0.638 dB (total)*
*Av. PSNR diff. - 40 equivalent (3 ranges) quality layers: -0.093 dB (0.001 to 1 bps) -0.058 dB (total)*

RESULTS: *average of corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI/ROC - BOI v1.2*

Graphic 5.5: *Coding performance evaluation of CPI, ROC and the best strategies of quality layers allocation compared to PCRD.*

# Chapter 6

# Rate control by the characterization of the rate-distortion slope

In this chapter we introduce a rate control method based on the Characterization of the Rate-Distortion slope (CoRD). This characterization uses the coding level, the subband and the number of magnitude bit-planes of the code-block in order to estimate the rate-distortion slopes. CoRD shares the same features of CPI and ROC, it obtains the best coding performance among them, and it solves some of the drawbacks of ROC. Extensive experimental results comparing CPI, ROC and CoRD are supplied in the last section of this chapter.

It is foreseen to submit this research to a journal after the public defense of this dissertation.

## 6.1 Introduction

The main goal of ROC is to improve the results of CPI in order to achieve a regular coding performance among all bit-rates. Although this goal has been achieved, ROC has two objectionable points. The first point is related with the theoretical approach used to explain the unbalanced distribution of the distortion contributions among resolution and coding levels, introduced in Section 5.1 and expounded in Tables 5.1, 5.2 –page 103–. These tables show the distortion contribution of each resolution level among the coding levels using a percentage based exclusively on the MSE. This percentage is computed as

$$C^c_{\mathcal{L}_j} = \frac{\triangle D^c_{\mathcal{L}_j}}{\triangle D^c}$$

where $\triangle D^c$ denotes the distortion contribution (i.e. $\triangle D^c = D^{c+1} - D^c$) of all code-blocks at the coding level $c$, and $\triangle D^c_{\mathcal{L}_j}$ denotes the distortion contribution of all code-blocks belonging to the resolution level $\mathcal{L}_j$ at the coding level $c$.

Note that this approach does not take into account the number of coefficients contained at each resolution level. This is, the lowest resolution levels contain less coefficients because they have smaller sizes than the highest resolution levels. This causes that the greatest percentages of MSE are commonly found at the highest resolution levels just because, at the same coding level, they include more coefficients than the lowest resolution levels. Only at the highest coding levels, when the coefficients of the highest resolution levels are not included yet, the lowest resolution levels have the largest percentages of MSE.

One might expect that this approach could have been better approximated if it had considered the number of coefficients belonging to each resolution level or, following the rate-distortion approach of the PCRD method, if it had considered the increment in bit-rate of the code-stream generated for each coding level. The consideration of the number of coefficients could be expressed as

$$C'^c_{\mathcal{L}_j} = \frac{\triangle D^c_{\mathcal{L}_j} / Q_{\mathcal{L}_j}}{\triangle D^c / Q}$$

where $Q$ stands for the number of coefficients of the image and $Q_{\mathcal{L}_j}$ stands for the number of coefficients belonging to the resolution level $\mathcal{L}_j$. The same for the bit-rate increment at each coding level,

$$C''^c_{\mathcal{L}_j} = \frac{\triangle D^c_{\mathcal{L}_j} / \triangle R^c_{\mathcal{L}_j}}{\triangle D^c / \triangle R^c}$$

where $\triangle R^c_{\mathcal{L}_j}$ stands for the bit-rate increment at coding level $c$ for those coding passes belonging to the resolution level $\mathcal{L}_j$, and $\triangle R^c$ stands for the bit-rate increment of the complete coding level $c$. Both $C'^c_{\mathcal{L}_j}$ and $C''^c_{\mathcal{L}_j}$ express a rate-distortion measure. Table 6.1 compares $C^c_{\mathcal{L}_j}$, $C'^c_{\mathcal{L}_j}$ and $C''^c_{\mathcal{L}_j}$ at some coding levels of the Cafeteria image. For $C'^c_{\mathcal{L}_j}$ and $C''^c_{\mathcal{L}_j}$, the greatest percentages are found at other resolution levels than the highest ones and nor $C'^c_{\mathcal{L}_j}$ neither $C''^c_{\mathcal{L}_j}$ justify the concatenation of coding passes of type MRP with CP. Studying all the whole tables, we conclude that these approaches do not explain either the distortion, or rate-distortion, contributions among resolution and coding levels.

Table 6.1: *Percentage of the rate-distortion contribution of each coding and resolution level using three measures.*

| | *cod. level* | $\mathcal{L}_0$ | $\mathcal{L}_1$ | $\mathcal{L}_2$ | $\mathcal{L}_3$ | $\mathcal{L}_4$ | $\mathcal{L}_5$ |
|---|---|---|---|---|---|---|---|
| $C^c_{\mathcal{L}_j}$ | 17 (SPP) | 0.03 % | 0.73 % | 3.97 % | 14.56 % | 39.42 % | 41.30 % |
| | 16 (MRP) | 0.50 % | 1.01 % | 2.72 % | 6.38 % | 11.84 % | 9.07 % |
| | 15 (CP) | 0.00 % | 0.01 % | 0.18 % | 2.15 % | 14.77 % | 51.37 % |
| $C'^c_{\mathcal{L}_j}$ | 17 (SPP) | 0.75 % | 21.23 % | 28.88 % | 26.51 % | 17.94 % | 4.70 % |
| | 16 (MRP) | 14.60 % | 29.42 % | 19.87 % | 11.63 % | 5.40 % | 1.03 % |
| | 15 (CP) | 0.01 % | 0.22 % | 1.29 % | 3.93 % | 6.74 % | 5.86 % |
| $C''^c_{\mathcal{L}_j}$ | 17 (SPP) | 7.15 % | 20.88 % | 19.63 % | 18.23 % | 17.58 % | 16.53 % |
| | 16 (MRP) | 3.08 % | 9.09 % | 9.26 % | 9.86 % | 10.79 % | 12.43 % |
| | 15 (CP) | 1.31 % | 3.38 % | 9.90 % | 10.55 % | 10.71 % | 9.63 % |

RESULTS: *Cafeteria image of the corpus ISO 12640-1, gray scaled, size 2048x2560.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, RESTART coding variation. Lossy compression.*
IMPLEMENTATION: *BOI v1.2*

The second objectionable point of ROC is the algorithm formulation using three different scanning orders: at high coding levels the scanning order is the same one used in CPI, then it is slightly modified changing the resolution level order and concatenating MRP and CP, and at the lowest bit-plane this concatenation is not performed. The change from the first scanning order to the second one is based on practical experimentation, but in some cases it may not work appropriately. For example, when encoding small images or when retrieving a WOI, the 55% of the code-blocks belonging to $\mathcal{L}_L$ can represent just a few code-blocks, causing an inappropiate change from one scanning order to the other one. Besides, the first two scanning orders are pretty different and the change between them is quite sharp. It is reasonable to think that it might exist a smooth way to do this change.

These two objectionable points have motivated the research presented in this chapter. We introduce a new approach that better explains the rate-distortion contributions of resolution levels among coding levels, based on the characterization of the rate-distortion slope of code-blocks. As well as CPI and ROC, the rate control developed using this characterization fulfills the requirements set in Section 4.1 and improves the coding performance of ROC, solving the drawbacks pointed above.

## 6.2 Optimal scanning order

In short, the main difference between ROC and the rate control method introduced in this chapter is that the latter also takes into account the number of magnitude bit-planes of each code-block. Let us discuss this statement. The scanning orders of CPI and ROC are based on the subbands. This means that their scanning orders are defined just considering the subbands of the image but, within each subband, CPI and ROC do not perform any distinction among the code-blocks that the subband contains. However, the decoding of the packet headers (needed for both CPI and ROC), also allows the identification of the number of magnitude bit-planes of each code-block. This information is used in the rate control method presented here to distinguish differences among the code-blocks of a subband.

We use an approach similar to the optimal PCRD method, performing the same operations, but estimating, instead of actually computing, the rate-distortion slope of the code-blocks. All coding levels of code-blocks belonging to the same subband and with the same number of magnitude bit-planes are estimated with the same rate-distortion slope, thus they are grouped in a so-called code-block set. Each code-block set is identified by $b_{j,g}$ and $K$, where $K$ stands for the number of magnitude bit-planes of the code-blocks and $b_{j,g}$ denotes the resolution level and subband to which the code-blocks belong, with $j = \mathcal{L}_j$ and with $g$

according to

$$g = \begin{cases} 0 & \text{if } subband = \{LL, HL, LH\} \\ 1 & \text{if } subband = \{HH\} \end{cases} .$$

We do not make distinctions among code-blocks of subbands $\{HL, LH\}$ because these subbands contain, theoretically, the same type of information. More precisely, the subbands $\{HL, LH\}$ of the same resolution level share the same quantization step size (which is usually the L2norm of the subband), therefore the signal difference among them just corresponds to the characteristics of the image, depending on the vertical and horizontal details. It is clear that an interleaving-based rate control method can not make any distinction.

This type of subband grouping is also used in Masuzaki [65]. The inclusion of the subband $LL$ in this group is for the sake of simplicity, since $LL$ is also distinguished by the resolution level (i.e., when $j = 0$). We refer to the code-blocks with the same $b_{j,g}, K$ as the set of code-blocks $\mathcal{G}$.

In order to attain a target bit-rate, the same operations performed by the PCRD are applied using the rate-distortion slope estimations of the code-block sets. This is, the convex hull identifies the feasible truncation points of each $\mathcal{G}$ and then a generalized Lagrange multiplier is applied. As the reader probably has already noticed, the problem here is how to fairly estimate the rate-distortion slopes of each code-block set. This estimation is performed using the characterization of the rate-distortion slope introduced in the following section. However, before using this characterization, we first evaluate the coding performance that can be obtained when using the actual rate-distortion slopes, in order to validate this approach from a theoretical point of view.

The use of actual rate-distortion measures is simple. First, the bit-rate and distortion of each $\mathcal{G}$ is computed for every coding level. Following the notation introduced in Section 5.1, $D_i^c$ stands for the distortion of code-block $\mathcal{B}_i$ at coding level $c$ and is computed as

$$D_i^c = w_{b_i}^2 \sum_{k \in \mathcal{B}_i} (y_i[k] - \hat{y}_i^c[k])^2$$

and here $D_{\mathcal{G}}^c$ stands for the sum of distortions of the code-blocks belonging to the set $\mathcal{G}$, this is

$$D_{\mathcal{G}}^c = \sum_{\mathcal{B}_i \in \mathcal{G}} D_i^c \ .$$

Let $R_i^c$ be the bit-rate of the code-stream generated for the code-block $\mathcal{B}_i$ at the coding pass $\mathcal{P}_i^c$, and $R_{\mathcal{G}}^c$ be the sum of these bit-rates for the code-blocks belonging to the set $\mathcal{G}$. Then, the slope of each set of code-blocks is computed as

$$S_{\mathcal{G}}^c = \frac{\triangle D_{\mathcal{G}}^c}{\triangle R_{\mathcal{G}}^c} = \frac{D_{\mathcal{G}}^{c+1} - D_{\mathcal{G}}^c}{R_{\mathcal{G}}^c - R_{\mathcal{G}}^{c+1}} \ .$$

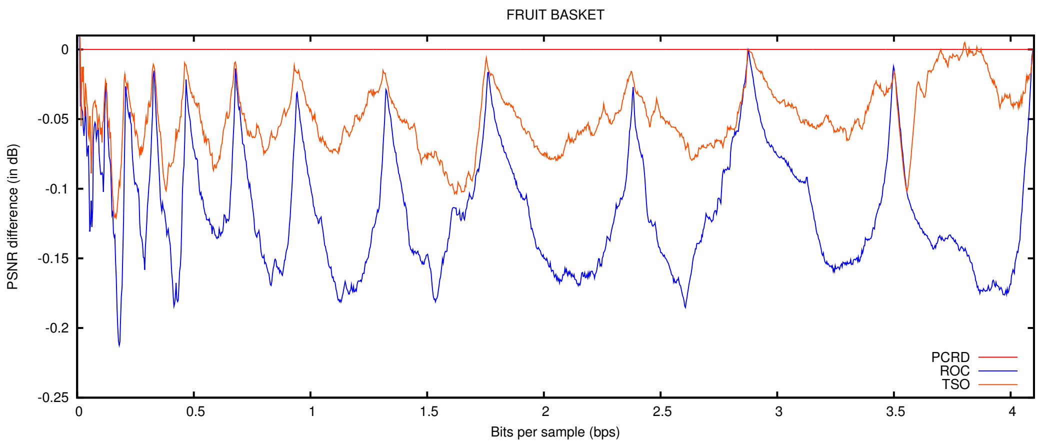
After the identification of the feasible truncation points, the search of the optimal truncation points for a specified target bit-rate can be performed straightforward. We name this rate control method Theoretic optimal Scanning Order (TSO), noting that its interest is only from a theoretic point of view, since it can only be applied at encoding time.

Graphic 6.1 depicts the coding performance achieved with ROC and TSO, compared to the PCRD method when encoding the Fruit Basket image. As expected, at almost all bit-rates the coding performance of TSO is better than the one obtained with ROC. The results for the other images of the corpus ISO 12640-1 are similar, which validates this approach.

## 6.3 Development of the CoRD rate control method

### 6.3.1 Characterization of the rate-distortion slope

To obtain a coding performance similar to the obtained with TSO, we need to properly estimate the rate-distortion slope of the code-block sets. As Yeung and Au state, to find a good model for all images is a challenging problem [110]: "*Model based rate allocation is an attractive approach for fast rate control as it can provide the optimal quality when the coefficients follow the model assumption. However, the major drawback is the degree of model accuracy. It is unlikely that we can find a model that is good for all images.*".




RESULTS: *Fruit Basket image of the corpus ISO 12640-1, gray scaled, size 2048x2560. 2000 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, 1 quality layer, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD - Kakadu v4.5, ROC/TSO - BOI v1.2*

Graphic 6.1: *Detailed coding performance evaluation of ROC and TSO compared to PCRD.*

The rate-distortion slope is a measure which reflects the efficiency achieved by the sample data coding stage when encoding each coding pass. Therefore, if we want to develop a fair characterization, we have to go beyond the rate-distortion slope itself, looking for the causes that change this efficiency and focusing our attention on the coding levels of the code-block sets.

### Concatenation of coding passes

We begin from what we already know. Considering the lowest and greatest rate-distortion slope of all code-blocks of the image at the same coding level $c$, referred to as $S^c_{min}$ and $S^c_{max}$ respectively, from CPI and ROC we know that these slopes satisfy the condition

$$S^{c+1}_{min} > S^c_{max} > S^c_{min} > S^{c-1}_{max} \qquad \text{(condition 1)}$$

when $c$ denotes the coding level containing coding passes of type SPP. However, the first main modification of ROC with respect to CPI is that ROC concatenates the coding pass of type MRP with the consecutive coding pass of type CP of all code-blocks from a particular coding level onwards. This means that, if $c$ denotes the coding level MRP, we can only assure that

$$S^{c+1}_{min} > S^{c-1}_{max} \geq S^{c}_{max} > S^{c}_{min} \geq S^{c-1}_{min} > S^{c-2}_{max} \qquad \text{(condition 2).}$$

For coding passes of type SPP, condition 1 coincides with the study presented in [100, Chapter 8.3.3], that concludes that the best order for the coding passes is $[SPP, MRP, CP]$. However, for coding passes of type MRP and CP, condition 2 does not coincide with this study, and the approach used in ROC is not conclusive either.

We explain the concatenation expressed by condition 2 due to the great difference in the *distortion decrement* between the encoding of a significant coefficient and the encoding of a refinement coefficient, compared to the small difference in the *bit-rate increment* between the encoding of a significant coefficient and the encoding of a refinement coefficient. For instance, the model used in Vikram [103] (which is derived from [53]) estimates the decrement in distortion and increment in bit-rate of a code-block at the bit-plane $p$ according to

$$\triangle D = (N_{\text{sig}} + 0.25 \cdot N_{\text{ref}}) \cdot (2^p)^2 \qquad \triangle R = 2 \cdot N_{\text{sig}} + N_{\text{ref}} + N_{\text{insig}}$$

where $N_{\text{sig}}, N_{\text{ref}}, N_{\text{insig}}$ denote, respectively, the number of significant, refinement and insignificant coefficients at bit-plane $p$ ($p = 0$ denotes the lowest bit-plane). Note that the encoding of a significant coefficient decreases the distortion in an exponential way, causing that the differences in distortion between the encoding of a significant coefficient and a refinement coefficient are meaningful, specially at high bit-planes. On the other hand, the increment on the bit-rate is lineal, and the differences in distortion between the encoding of a significant coefficient and a refinement coefficient are not that large.

Besides, this model does not consider that the MQ-coder reduces the increment in bit-rate, specially for coding passes of type SPP and CP. In addition, this model does not distinguish either between coding passes type. If we take into account that the run mode of

CP can encode four insignificant coefficients in a single bit, this model might calculate the decrement in distortion and increment in bit-rate for coding passes of type MRP and CP as

$$\triangle D^{MRP} = (0.25 \cdot N_{\text{ref}}) \cdot (2^p)^2 \quad \triangle R^{MRP} = N_{\text{ref}}$$
$$\triangle D^{CP} = N_{\text{sig}} \cdot (2^p)^2 \quad \triangle R^{CP} = 2 \cdot N_{\text{sig}} + 0.25 \cdot N_{\text{insig}}$$

Through these estimations, we can calculate when the rate-distortion slope of coding passes of type CP, referred to as $S^{CP}$, is greater than the rate-distortion slope of coding passes of type MRP, referred to as $S^{MRP}$, by

$$S^{CP} > S^{MRP} \equiv \frac{\triangle D^{CP}}{\triangle R^{CP}} > \frac{\triangle D^{MRP}}{\triangle R^{MRP}} \rightarrow N_{\text{sig}} > 0.125 \cdot N_{\text{insig}}$$

inferring that $S^{CP} > S^{MRP}$ when at least 12.5% of the coefficients encoded in a coding pass of type CP are significant. Apart from at the highest bit-planes, this happens quite often, as we can see in Table 6.2 for the Musicians image. This approach perfectly fits with the scanning order followed by ROC, which concatenates MRP with CP from a particular coding level onwards.

Table 6.2: *Percentage of significant coefficients encoded in coding passes of type CP at different bit-planes, in average for different code-block sets.*

| *bit-plane* | $b_{3,0}$ K=9 | $b_{3,1}$ K=8 | $b_{4,0}$ K=7 | $b_{4,1}$ K=7 | $b_{5,0}$ K=6 | $b_{5,1}$ K=6 |
|---|---|---|---|---|---|---|
| 8 | 1.00 % | | | | | |
| 7 | 1.33 % | 0.36 % | | | | |
| 6 | 1.15 % | 1.15 % | 0.11 % | 0.43 % | | |
| 5 | 2.39 % | 1.84 % | 1.80 % | 2.00 % | 0.59 % | 0.29 % |
| 4 | 10.05 % | 15.83 % | 14.30 % | 13.91 % | 7.51 % | 3.33 % |
| 3 | 17.84 % | 27.84 % | 22.51 % | 23.47 % | 16.75 % | 12.47 % |
| 2 | 22.86 % | 30.00 % | 26.24 % | 17.15 % | 17.13 % | 15.55 % |
| 1 | 41.43 % | 25.87 % | 75.00 % | 30.56 % | 20.77 % | 22.97 % |
| 0 | - | - | - | 50.00 % | 32.95 % | 39.81 % |

RESULTS: *Musicians image of the corpus ISO 12640-1, gray scaled, size 2048x2560.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, expounded quantization using L2norms, 64x64 code-blocks, RESTART coding variation.*
IMPLEMENTATION: *BOI v1.2*

**The balloon effect**

The second main modification of ROC with respect to CPI is that ROC changes the scanning order of the resolution levels and subbands. To explain this change, we consider the following assumption: the coding passes that encode the largest number of significant coefficients have the greatest rate-distortion slope values. This assumption is based on the meaningful difference between the large decrement in distortion compared to the small increment in bit-rate when a significant coefficient is encoded. For coding passes of type SPP and CP, the rate-distortion model proposed in Vikram [103] estimates this according to

$$\triangle D = N_{\text{sig}} \cdot (2^p)^2 \qquad \triangle R = 2 \cdot N_{\text{sig}} + N_{\text{insig}}$$

and, although the encoding of a significant coefficient increases twice the bit-rate of the encoding of an insignificant coefficient, $\triangle D$ is decremented by $(2^p)^2$! Therefore, it is expected that, as more significant coefficients are encoded in a coding pass, greater its rate-distortion slope is, specially at high bit-planes. This causes that, at the same coding level, the code-blocks that have the largest number of significant coefficients, have the greatest rate-distortion slopes. Similar assumptions are also used in Qin [75] with the aim to estimate the rate-distortion slope by using the number of significant and refinement coefficients at each bit-plane.

Our goal is to identify the number of significant coefficients encoded at each coding level of a code-block. We focus our attention on the code-blocks of one subband, distinguishing the magnitude bit-planes of each code-block. Table 6.3 shows the average number of significant coefficients encoded at each bit-plane for the code-blocks belonging to the subband $b_{3,0}$ of the Candle image, grouped by the coding pass type (SPP or CP).

It is worth noting two issues in this table. The first one is that, for coding passes of type CP, the number of significant coefficients encoded at each bit-plane increases from the first to the third highest bit-plane, and then decreases progressively until reaching the lowest bit-plane. The same for coding passes of type SPP, but the increase is from the first to the sixth (or seventh) highest bit-plane. We can see this as a balloon, where the width

Table 6.3: *Average number of significant coefficients encoded in each bit-plane of the code-block sets in subband $b_{3,0}$.*

| | CP | | | SPP | | |
|---|---|---|---|---|---|---|
| *bit-plane* | K=11 | K=10 | K=9 | K=11 | K=10 | K=9 |
| 10 | 1 | | | | | |
| 9 | 23 | 15 | | 5 | | |
| 8 | 238 | 110 | 27 | 68 | 25 | |
| 7 | 118 | 159 | 129 | 392 | 195 | 50 |
| 6 | 53 | 112 | 163 | 418 | 451 | 266 |
| 5 | 71 | 77 | 125 | 505 | 583 | 507 |
| 4 | 94 | 55 | 82 | 609 | 621 | 646 |
| 3 | 16 | 27 | 33 | 616 | 550 | 669 |
| 2 | 1 | 10 | 11 | 403 | 421 | 514 |
| 1 | 0 | 9 | 8 | 198 | 282 | 365 |
| 0 | 0 | 1 | 1 | 139 | 184 | 240 |

RESULTS: *Candle image of the corpus ISO 12640-1, gray scaled, size 2048x2560.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, expounded quantization using L2norms, 64x64 code-blocks, RESTART coding variation.*
IMPLEMENTATION: *BOI v1.2*

represents the number of significant coefficients encoded at the bit-plane (see Figure 6.1). The consequence of this issue is that, from a particular bit-plane onwards, at the same bit-plane the code-blocks that have the largest number of significant coefficients are those ones that have less magnitude bit-planes. This can be appreciated in Table 6.3 from bit-plane 6 and below for coding passes of type CP and from bit-plane 4 and below for coding passes of type SPP.

The second remarkable issue of this Table is that the number of significant coefficients at the highest bit-plane depends on the magnitude bit-planes of the code-block. This is, at their highest bit-plane, the code-blocks that encode more number of significant coefficients are those ones which have the lowest number of magnitude bit-planes. In Table 6.3 this relation is always respected. The consequence of this issue is that the code-blocks with a lower number of magnitude bit-planes increase the number of encoded significant coefficients

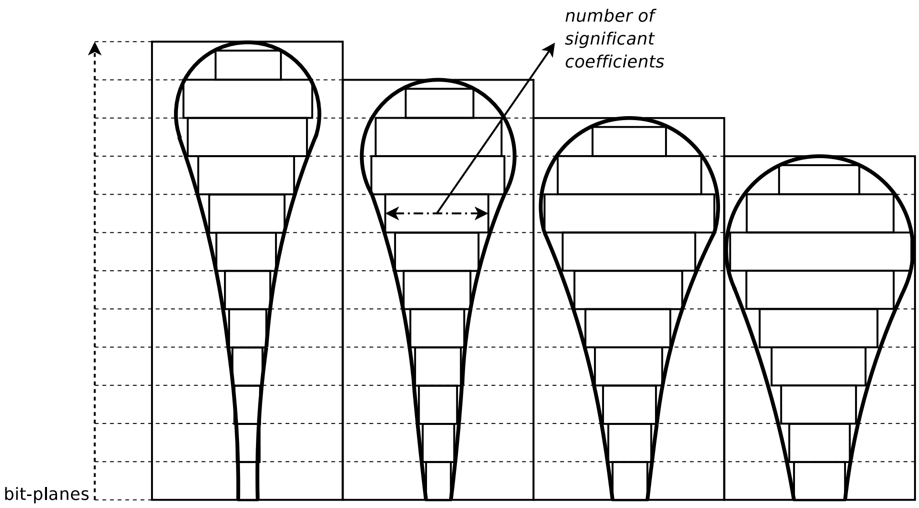



Figure 6.1: *The balloon effect.*

faster than those code-blocks that have a larger number of magnitude bit-planes. Figure 6.1 depicts this as the shape of the balloon: the shorter the balloon is, the wider it is in the middle. We call these issues the *balloon effect*.

On the other hand, the code-blocks belonging to the subbands of the lowest resolution levels have greater quantization weights than the code-blocks belonging to the highest resolution levels, i.e. the L2norm of the lowest resolution levels is greater than the L2norm of the highest resolution levels. Therefore, the code-blocks of the lowest resolution levels have more magnitude bit-planes than the code-blocks of the highest resolution levels. Figure 6.2 depicts this fact by the height of two code-blocks belonging to subbands $HH_3$ and $HH_1$. Notice the consequence of this fact combined with the balloon effect: at the highest bit-planes, the best scanning order is achieved when starting at the lowest resolution levels but, from a particular bit-plane onwards, the best scanning order is achieved when starting from the highest resolution levels, just because from that bit-plane onwards the highest resolution levels have, in average, more significant coefficients, and therefore it is expected that they have the greatest rate-distortion slopes.

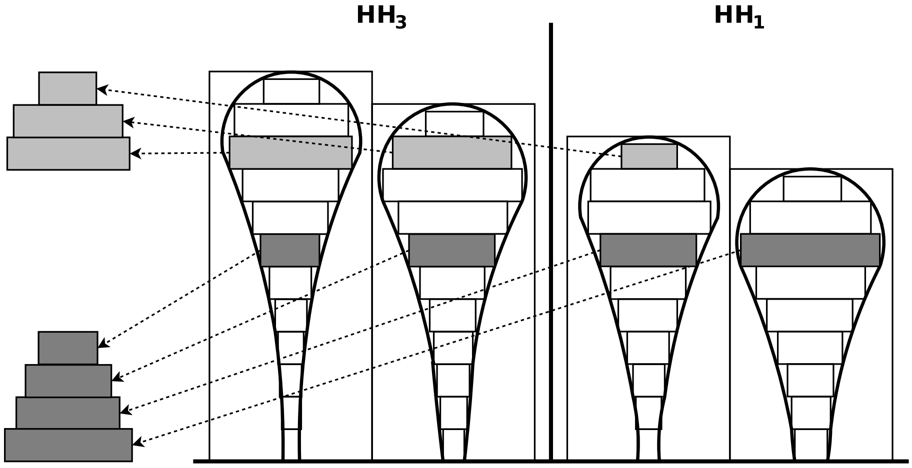



Figure 6.2: *The balloon effect among different subbands.*

We explain why the balloon effect occurs as follows. First, we know that the largest number of significant coefficients are usually found at high bit-planes, specially for coding passes of type CP. This could be caused for the application of the high-pass filter of the DWT, which sets most coefficients of a subband to a null value, apart from the areas where high frequencies are detected. These high frequencies areas usually have large values concentrated in the same spatial locations. Second, the scan performed in coding passes of type CP visits almost all the coefficients of the code-block at the highest bit-planes, therefore, it is expected that it discloses these areas of high frequencies. The more significant coefficients the CP encodes, the less coefficients at the following bit-plane it visits, and consequently the less significant coefficients it encodes. At the lowest bit-planes, almost all the coefficients still insignificant are neighbours of already significant coefficients, so while CP does not visit almost any coefficient, the SPP encodes all the coefficients of the medium and lowest bit-planes. This also explains why the balloon effect is more emphasized in coding passes of type CP than in coding passes of type SPP.

Although the balloon effect is observed in all the subbands of an image, the subband $LL$ is completely different. We believe this is caused due to its content, which is the response to the low-pass filter of the DWT, whereas all the other subbands are combinations of low-pass and high-pass filters. However, since the $LL$ subband only contains a small number of code-blocks, for simplicity we do not consider $LL$ differently in the characterization of the rate-distortion slope. Experimental evidence indicates that to consider this subband as a separate case does not improve significantly the coding performance.

### 6.3.2 Algorithm

We formulate the characterization of the rate-distortion slope as

$$S_{\mathcal{G}}^{c} = \begin{cases} c + \mathcal{F}_{SPP} & \text{if } c \; mod \; 3 = 2 \quad \text{(SPP coding passes)} \\ c + \mathcal{F}_{MRP} & \text{if } c \; mod \; 3 = 1 \quad \text{(MRP coding passes)} \\ c + 1 + \mathcal{F}_{CP} & \text{if } c \; mod \; 3 = 0 \quad \text{(CP coding passes)} \end{cases}$$

and, in order to fulfill the condition $S_{min}^{c+1} > S_{max}^{c}$ introduced in the previous section, we restrict the values to $\mathcal{F}_{\{SPP|MRP|CP\}} = [0, 1)$. In order to assure that the coding passes of type MRP are concatenated with the consecutive coding pass of type CP, we set $\mathcal{F}_{MRP} = 0$ except for the MRP of the highest bit-plane, where $\mathcal{F}_{MRP} = 0.99$, reflecting the characterization of this coding pass type explained in the previous section. For coding passes of type SPP and CP, $\mathcal{F}_{\{SPP|CP\}}$ represents the balloon effect within each subband, and is calculated as

$$\mathcal{F}_{\{SPP|CP\}} = \begin{cases} \mathcal{F}_{init} \cdot (\mathcal{F}_{inc})^{K_S - p} & \text{if } p \geq K_{balloon} \\ 1 - (\mathcal{F}_{dec} \cdot (K_{balloon} - p)) & \text{otherwise} \end{cases}$$

where $p$ denotes the bit-plane of the coding level $c$, and $K_S = K - 1$ for CP coding passes and $K_S = K - 2$ for SPP coding passes. From the highest to the lowest bit-plane, this expression increases the rate-distortion slope exponentially from bit-plane $K - 1$ to bit-plane $K_{balloon}$, and decreases the rate-distortion lineally from $K_{balloon} - 1$ to $0$. $K_{balloon}$ is set to the bit-plane that causes $\mathcal{F}_{\{SPP|CP\}} \geq 1$, this is

$$K_{balloon} = P \text{ such that } \nexists \;\; p > P,\;\; \mathcal{F}_{init} \cdot (\mathcal{F}_{inc})^{K_S - p} \geq 1\ .$$

The parameter $\mathcal{F}_{dec}$ sets the decreasing increment at each bit-plane and $\mathcal{F}_{inc}$ the increasing increment. $\mathcal{F}_{init}$ must reflect the rate-distortion slope initialization at the highest bit-plane, in other words, the width of the top of the balloon. Good choices for these three parameters are given in Table 6.4, where $\#K = K_{max} - K_{min} + 1$ with $K_{max}$ and $K_{min}$ denoting the maximum and minimum $K$ of the subband to which $\mathcal{G}$ belongs respectively.

These choices have been determined experimentally through a benchmark that considers the images of the corpora used in the following experimental section. The results suggest that: 1) $\mathcal{F}_{init}$ should be lower as larger is the number of magnitude bit-planes of the code-block set, being $0.075$ and $0.05$ the largest values for coding passes of type CP and SPP respectively, 2) $\mathcal{F}_{inc}$ needs to be set larger for coding passes of type CP than for coding passes of type SPP. These points coincide with the characterization explained above. $\mathcal{F}_{dec}$ has been adjusted for a lineal decreasing without reaching a null value. Experience indicates that slight variations on these parameters do not penalize the coding performance.

Table 6.4: *Choices of parameters $\mathcal{F}_{init}$, $\mathcal{F}_{inc}$ and $\mathcal{F}_{dec}$.*

| $\mathcal{F}_{CP}$ | $\mathcal{F}_{SPP}$ |
|---|---|
| $\mathcal{F}_{init} = \frac{0.075}{\#K} \cdot (K_{max} - K + 1)$ | $\mathcal{F}_{init} = \frac{0.05}{\#K} \cdot (K_{max} - K + 1)$ |
| $\mathcal{F}_{inc} = 10$ | $\mathcal{F}_{inc} = 4$ |
| $\mathcal{F}_{dec} = \frac{1}{K_{balloon} + 2}$ | $\mathcal{F}_{dec} = \frac{1}{K_{balloon} + 2}$ |

Table 6.5 shows the rate-distortion slope estimations calculated through this characterization. For the sake of simplicity, the code-block sets depicted in this example corresponds to the popular Lena image (size 512x512, gray-scaled) when 2 DWT levels are applied, since examples for images of the ISO 12640-1 corpus involves about 18 code-block sets. It is worth noticing the smooth change that the scanning order follows when we track the inclusion of code-block sets (from the greatest rate-distortion slope to the lowest one): at the

same coding level and at the highest bit-planes the scanning order includes first the code-block sets from the lowest resolution levels, and then it progressively changes including first the code-block sets from the highest resolution levels. This scanning order is similar to the one followed by ROC, but without rough changes. We name this rate control method as CoRD, standing for Characterization of the Rate-Distortion slope.

Table 6.5: *Example of the rate-distortion slope characterization.*

| | $b_{0,0}$ $(\mathcal{L}_0, LL)$ | $b_{1,0}$ $(\mathcal{L}_1, HL/LH)$ | | $b_{1,1}$ $(\mathcal{L}_1, HH)$ | | $b_{2,0}$ $(\mathcal{L}_2, HL/LH)$ | | $b_{2,1}$ $(\mathcal{L}_2, HH)$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *cod. level* | $K = 10$ | $K = 9$ | $K = 8$ | $K = 8$ | $K = 7$ | $K = 8$ | $K = 7$ | $K = 7$ | $K = 6$ | $K = 5$ |
| 27 (CP) | 28.08 | | | | | | | | | |
| 26 (SPP) | 26.05 | | | | | | | | | |
| 25 (MRP) | 25.99 | | | | | | | | | |
| 24 (CP) | 25.75 | 25.04 | | | | | | | | |
| 23 (SPP) | 23.2 | 23.03 | | | | | | | | |
| 22 (MRP) | 22.0 | 22.99 | | | | | | | | |
| 21 (CP) | 22.99 | 22.38 | 22.08 | 22.04 | | 22.04 | | | | |
| 20 (SPP) | 20.8 | 20.1 | 20.05 | 20.03 | | 20.03 | | | | |
| 19 (MRP) | 19.0 | 19.0 | 19.99 | 19.99 | | 19.99 | | | | |
| 18 (CP) | 19.88 | 19.99 | 19.75 | 19.38 | 19.08 | 19.38 | 19.08 | 19.03 | | |
| 17 (SPP) | 17.99 | 17.4 | 17.2 | 17.1 | 17.05 | 17.1 | 17.05 | 17.02 | | |
| 16 (MRP) | 16.0 | 16.0 | 16.0 | 16.0 | 16.99 | 16.0 | 16.99 | 16.99 | | |
| 15 (CP) | 16.77 | 16.87 | 16.99 | 16.99 | 16.75 | 16.99 | 16.75 | 16.25 | 16.05 | |
| 14 (SPP) | 14.85 | 14.99 | 14.8 | 14.4 | 14.2 | 14.4 | 14.2 | 14.07 | 14.03 | |
| 13 (MRP) | 13.0 | 13.0 | 13.0 | 13.0 | 13.0 | 13.0 | 13.0 | 13.0 | 13.99 | |
| 12 (CP) | 13.66 | 13.74 | 13.85 | 13.85 | 13.99 | 13.85 | 13.99 | 13.99 | 13.5 | 13.08 |
| 11 (SPP) | 11.7 | 11.82 | 11.99 | 11.99 | 11.8 | 11.99 | 11.8 | 11.27 | 11.13 | 11.05 |
| 10 (MRP) | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.99 |
| 9 (CP) | 10.55 | 10.62 | 10.7 | 10.7 | 10.82 | 10.7 | 10.82 | 10.82 | 10.99 | 10.75 |
| 8 (SPP) | 8.56 | 8.66 | 8.79 | 8.79 | 8.99 | 8.79 | 8.99 | 8.99 | 8.53 | 8.2 |
| 7 (MRP) | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 6 (CP) | 7.43 | 7.49 | 7.56 | 7.56 | 7.66 | 7.56 | 7.66 | 7.66 | 7.79 | 7.99 |
| 5 (SPP) | 5.42 | 5.49 | 5.59 | 5.59 | 5.74 | 5.59 | 5.74 | 5.74 | 5.99 | 5.8 |
| 4 (MRP) | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 |
| 3 (CP) | 4.32 | 4.36 | 4.42 | 4.42 | 4.49 | 4.42 | 4.49 | 4.49 | 4.59 | 4.74 |
| 2 (SPP) | 2.28 | 2.32 | 2.39 | 2.39 | 2.49 | 2.39 | 2.49 | 2.49 | 2.66 | 2.99 |
| 1 (MRP) | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 0 (CP) | 1.21 | 1.24 | 1.28 | 1.28 | 1.32 | 1.28 | 1.32 | 1.32 | 1.39 | 1.49 |

The algorithm of the rate control method CoRD is formulated in Table 6.6. The first operation is to construct the code-blocks sets. The complete collection of the code-block sets is denoted as $\mathcal{G}^i$ with $0 \leq i < \#\mathcal{G}$, with $\#\mathcal{G}$ standing for the number of code-block sets of the image. The second step of the algorithm is to estimate the rate-distortion slopes of every $\mathcal{G}^i$. The computation of the feasible truncation points of each $\mathcal{G}^i$ is based on the slope value, just discarding those points with $S^c_{\mathcal{G}^i} \leq S^{c-1}_{\mathcal{G}^i}$ and it is performed simultaneously with the estimation of the rate-distortion slope. Then the algorithm searches the maximum slope value and encodes the coding passes of that $\mathcal{G}^i$, updating the bit-rate consecutively until the specified target bit-rate is achieved.

Table 6.6: *The Characterization of the Rate-Distortion slope (CoRD) algorithm.*

*CONSTRUCT code-block sets* $\mathcal{G}^i$
**for each** *code-block set* $\mathcal{G}^i$ **from** $i = 0$ **to** $\#\mathcal{G}$ **do**
  **for each** *coding level* $c$ **from** $c = K^i \cdot 3 - 2$ **to** $0$ **do**

$$\textbf{set } S^c_{\mathcal{G}^i} \leftarrow \begin{cases} c + \mathcal{F}_{SPP} & \text{if } c \mathbin{mod} 3 = 2 \\ c + \mathcal{F}_{MRP} & \text{if } c \mathbin{mod} 3 = 1 \\ c + 1 + \mathcal{F}_{CP} & \text{if } c \mathbin{mod} 3 = 0 \end{cases}$$

    **if** $S^c_{\mathcal{G}^i} \geq S^{c+1}_{\mathcal{G}^i}$ **then**
      *DISCARD* $S^{c+1}_{\mathcal{G}^i}$ *as a feasible point*
    **endif**
  **endfor**
**endfor**
**set** *bitRate* ← *0*
**set** $currS^c_{\mathcal{G}^i}$ ← *null*
**do**
  **set** $currS^c_{\mathcal{G}^i}$ ← *searchMaxSlope of* $S^c_{\mathcal{G}^i}$
  **for each** *coding pass* $\mathcal{P}$ **belonging to** $currS^c_{\mathcal{G}^i}$ **do**
    *ENCODE coding pass* $\mathcal{P}$
    **set** *bitRate* ← *bitRate + length(*$\mathcal{P}$*)*
    **if** *bitRate* ≥ *targetBitRate* **then**
      *STOP encoding*
    **endif**
  **endfor**
**while** *(*$currS^c_{\mathcal{G}^i} \neq$ *null)*

Notice that the algorithm performs a full search among all the slopes of the code-block sets. This greedy approach can be used because the grouping of code-blocks widely reduces the number of truncation points that we need to manage. In the PCRD method this step is usually performed applying a bi-section search, although greedy algorithms have also been used, for instance in {E}-IREC [116].

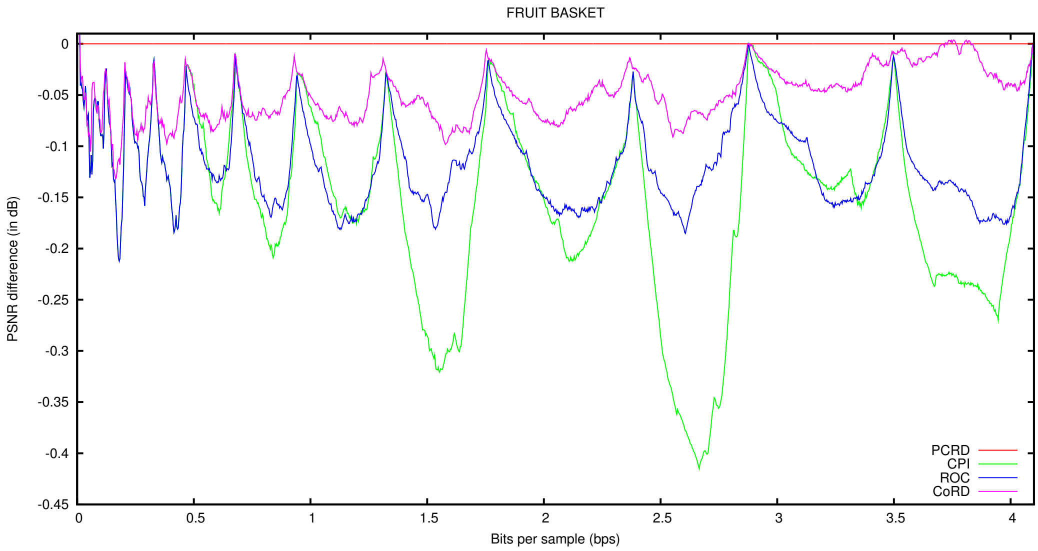
**Remark 6.3.1** *The search among rate-distortion slopes is simplified by grouping code-blocks in sets. The number of code-blocks considered by the PCRD method when encoding the Cafeteria image (size 2048x2560, 5 DWT levels, code-block size of 64x64) is 1286; CoRD considers 37 different code-block sets. With a code-block size of 32x32, PCRD considers 5124 code-blocks; CoRD considers 52 different code-blocks sets.*

The computational complexity of the CoRD method is negligible. The single operation that is computationally demanding is the decoding of packet headers, when CoRD is used in the decoding side, in order to know the magnitude bit-planes of code-blocks and the bit-rates of coding passes. This operation usually takes less than 1% of the time employed to decode a code-stream.

Graphic 6.2 compares the coding performance between TSO and CoRD. Note that CoRD characterizes the rate-distortion slope practically as well as with the actual slope values used in TSO. Graphic 6.3 depicts the coding performance obtained with CPI, ROC and CoRD compared to optimal PCRD method for the Fruit Basket image. CoRD obtains the best coding performance.


*Same coding parameters as in Graphic 6.1*

Graphic 6.2: *Detailed coding performance evaluation of TSO and CoRD compared to PCRD.*




*Same coding parameters as in Graphic 6.1*

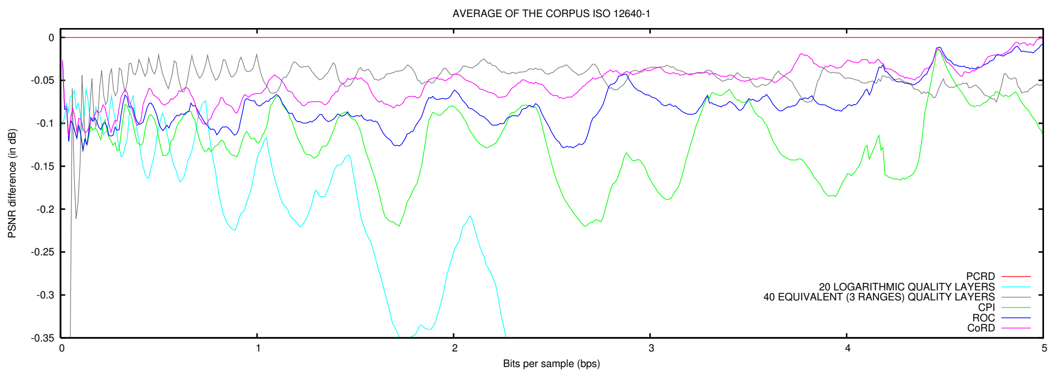
Graphic 6.3: *Detailed coding performance evaluation of CPI, ROC and CoRD compared to PCRD.*

## 6.4 Experimental results

We assess the coding performance of CPI, ROC, CoRD and the use of quality layers in four experimental sections. Section 6.4.1 presents the same experiments as in the experimental sections of CPI and ROC. Sections 6.4.2, 6.4.3 present the coding performance obtained when using different JPEG2000 coding parameters and different image corpus respectively. The coding performance obtained when extracting WOIs from a code-stream is presented in Section 6.4.4. Section 6.4.5 shows the computational complexity reduction achieved when applying CoRD to the coder and Section 6.4.6 provides a visual comparison.

### 6.4.1 Coding performance

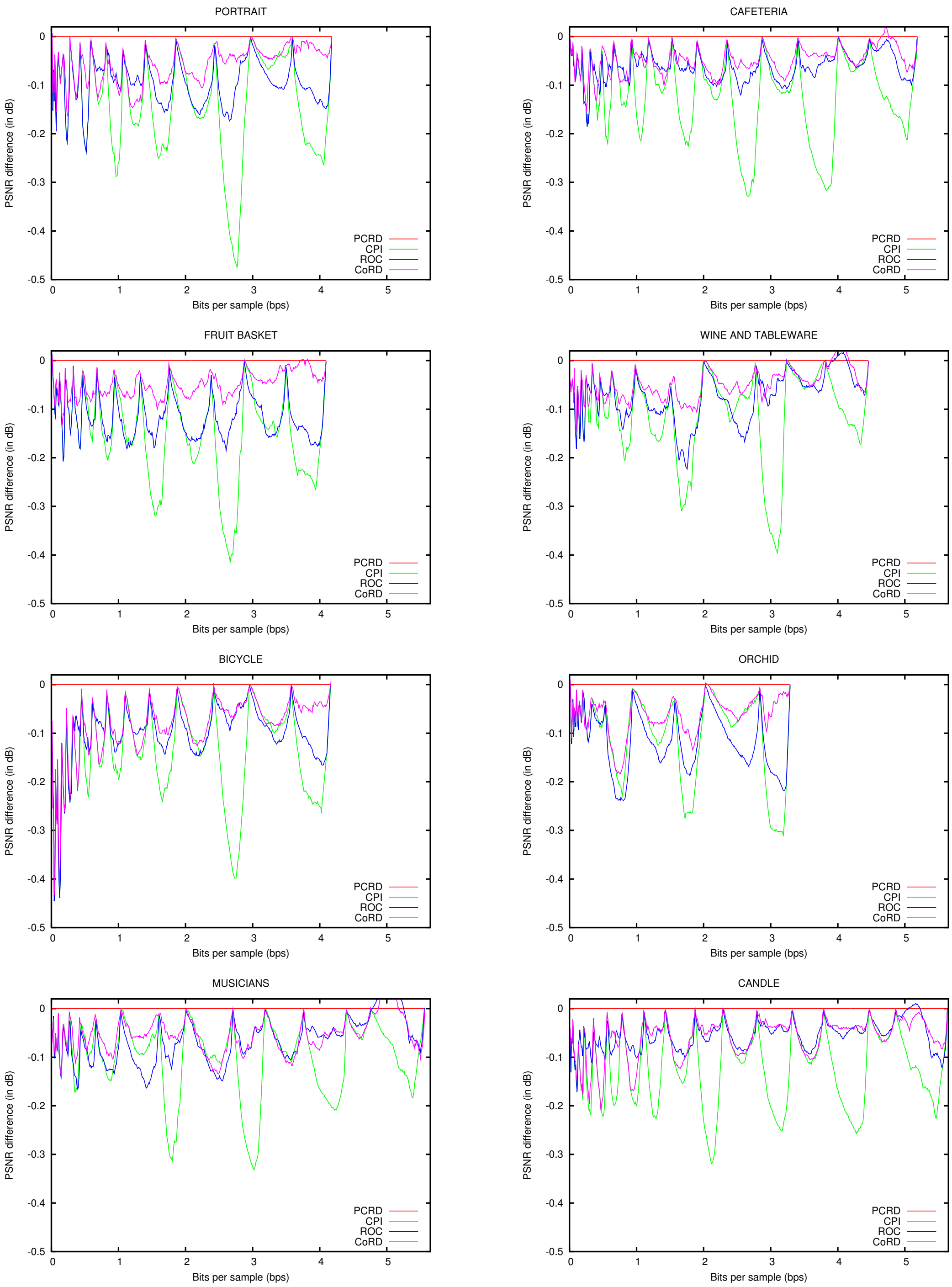
Graphic 6.4 and Graphic 6.5 depict the coding performance of CPI, ROC and CoRD for the corpus ISO 12640-1, in average and for each image respectively. CoRD improves the coding performance achieved by ROC in almost all images. Besides, in those images where ROC obtains its worst results (see Orchid, for example), CoRD maintains a good coding performance. Note that the coding performance of CoRD is almost always less than 0.1 dB worse than the PCRD method, while in some images ROC is 0.2 dB worse than PCRD. The rough change among scanning orders of ROC is not observed in CoRD, which obtains a similar coding performance in almost all bit-rates. The better regularity of CoRD with respect to ROC is observed in the average of the eight images. Besides, compared to the best strategies of quality layers allocation, CoRD obtains the best results in average.




*Av. PSNR diff. - CPI: -0.113 dB (0.001 to 1 bps) -0.122 dB (total)*
*Av. PSNR diff. - ROC: -0.096 dB (0.001 to 1 bps) -0.074 dB (total)*
*Av. PSNR diff. - CoRD: -0.081 dB (0.001 to 1 bps) -0.051 dB (total)*
*Av. PSNR diff. - 20 logarithmic quality layers: -0.129 dB (0.001 to 1 bps) -0.638 dB (total)*
*Av. PSNR diff. - 40 equivalent (3 ranges) quality layers: -0.093 dB (0.001 to 1 bps) -0.058 dB (total)*

RESULTS: *average of corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI/ROC/CoRD - BOI v1.2*

Graphic 6.4: *Coding performance evaluation of CPI, ROC, CoRD and the best strategies of quality layers allocation compared to PCRD.*



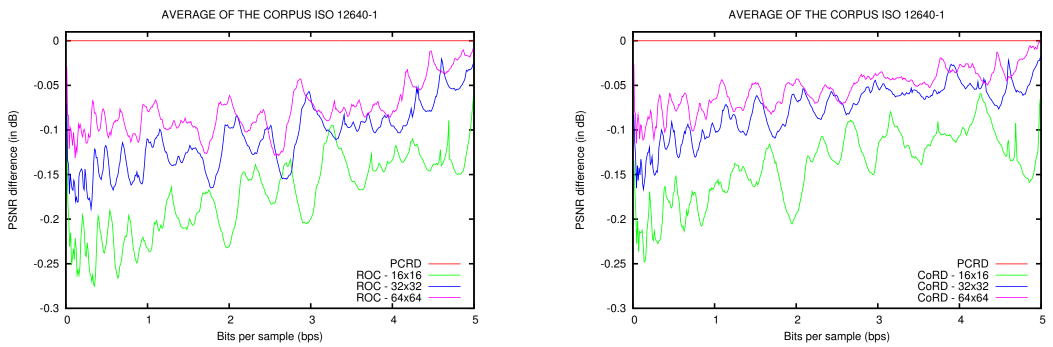

RESULTS: *images of the corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, 1 quality layer, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD - Kakadu v4.5, CPI/ROC/CoRD - BOI v1.2*

Graphic 6.5: *Coding performance evaluation of CPI, ROC and CoRD compared to PCRD.*

### 6.4.2 Other coding parameters

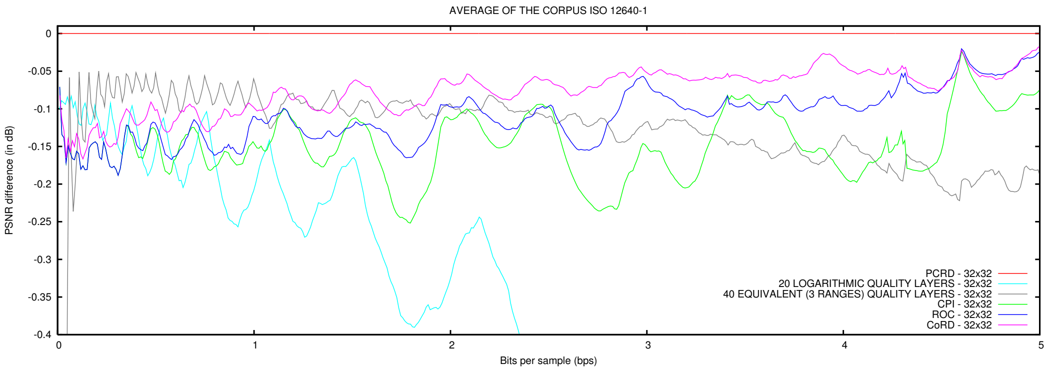
The core coding system of JPEG2000 allows a large variety of coding parameters, but changes on most of these parameters do not affect the coding performance achieved by CPI, ROC or CoRD. Only the code-block size and the number of DWT levels can affect the characterization performed in CoRD and, besides, in interactive image transmissions it is usual to use code-block sizes of 32x32. Therefore we evaluate how the changes in these parameters penalize the coding performance. We also evaluate the use of the Le Gall 5/3 filter-bank to assess the coding performance achieved by the lossless mode of JPEG2000.

Graphic 6.6 evaluates the coding performance achieved when using code-block sizes of 16x16, 32x32 and 64x64. ROC and CoRD work reasonably well for these code-block sizes, although when using code-blocks of size 16x16 the coding performance is slightly penalized. Graphic 6.7 depicts an evaluation of the coding performance for CPI, ROC, CoRD and the use of quality layers when using code-block sizes of size 16x16 and size 32x32. Code-blocks of size 32x32 achieve almost equivalent results as when using code-blocks of size 64x64, but for code-blocks of 16x16 the coding performance is widely penalized by the use of quality layers, whereas with CPI, ROC and CoRD it is maintained.

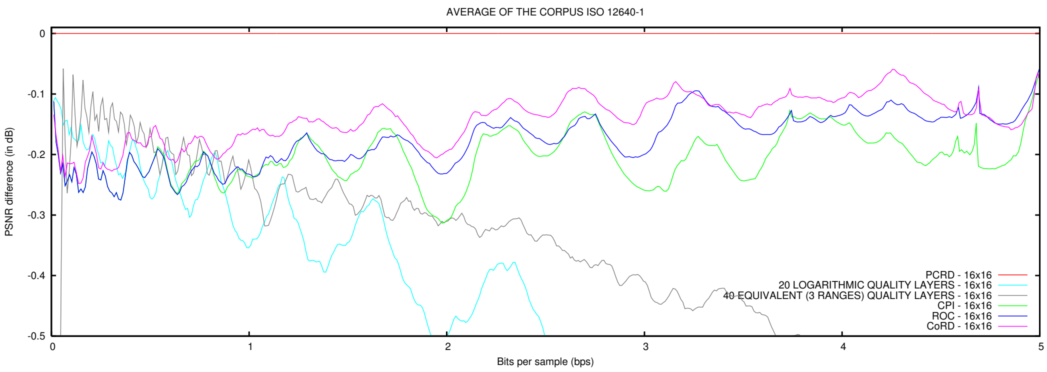



RESULTS: *images of the corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 / 32x32 / 16x16 code-blocks, no precincts, RESTART coding variation, 1 quality layer, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD - Kakadu v4.5, ROC/CoRD - BOI v1.2*

Graphic 6.6: *Coding performance evaluation of ROC and CoRD compared to PCRD when using different code-block sizes.*




*Av. PSNR diff. - CPI: -0.157 dB (0.001 to 1 bps) -0.147 dB (total)*
*Av. PSNR diff. - ROC: -0.149 dB (0.001 to 1 bps) -0.102 dB (total)*
*Av. PSNR diff. - CoRD: -0.119 dB (0.001 to 1 bps) -0.074 dB (total)*
*Av. PSNR diff. - 20 logarithmic quality layers: -0.154 dB (0.001 to 1 bps) -0.676 dB (total)*
*Av. PSNR diff. - 40 equivalent (3 ranges) quality layers: -0.123 dB (0.001 to 1 bps) -0.151 dB (total)*




*Av. PSNR diff. - CPI: -0.230 dB (0.001 to 1 bps) -0.196 dB (total)*
*Av. PSNR diff. - ROC: -0.229 dB (0.001 to 1 bps) -0.158 dB (total)*
*Av. PSNR diff. - CoRD: -0.193 dB (0.001 to 1 bps) -0.137 dB (total)*
*Av. PSNR diff. - 20 logarithmic quality layers: -0.227 dB (0.001 to 1 bps) -0.898 dB (total)*
*Av. PSNR diff. - 40 equivalent (3 ranges) quality layers: -0.217 dB (0.001 to 1 bps) -0.445 dB (total)*

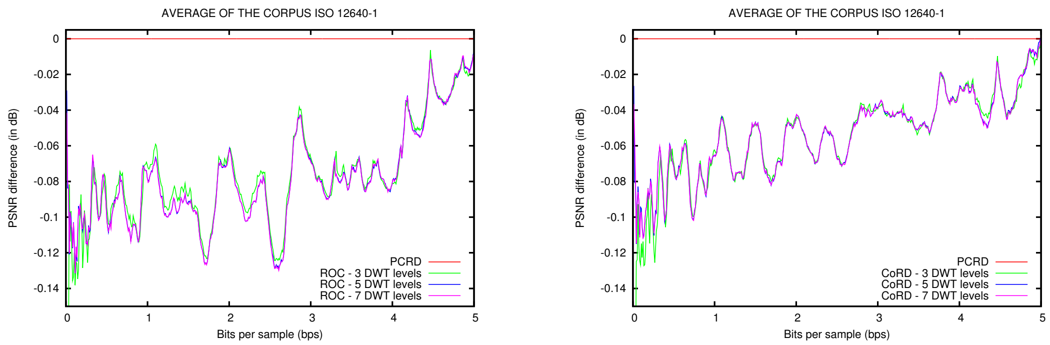
---

RESULTS: *average of corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 16x16 / 32x32 code-blocks, no precincts, RESTART coding variation, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI/ROC/CoRD - BOI v1.2*

---

Graphic 6.7: *Coding performance evaluation of CPI, ROC, CoRD and the best strategies of quality layers allocation compared to PCRD when using different code-block sizes.*

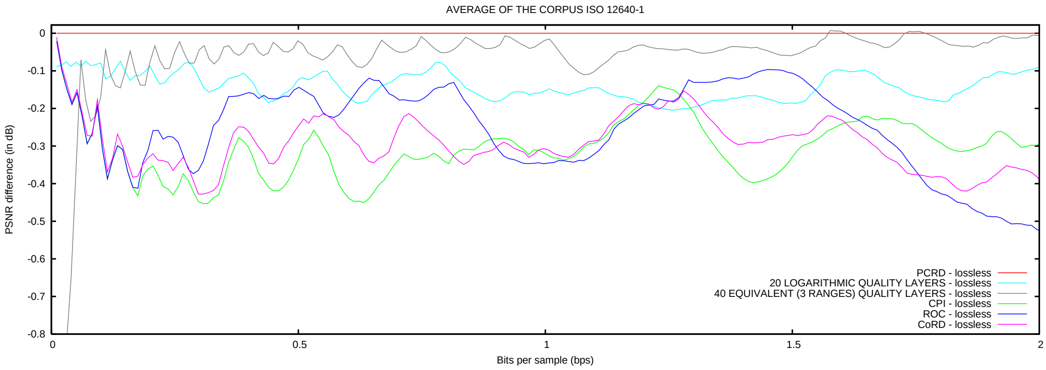
Graphic 6.8 evaluates the coding performance obtained with ROC and CoRD when using 3, 5 and 7 DWT levels. Apart from at very low bit-rates, where it is usual that the use of few DWT levels penalize the coding performance, the change in the number of DWT levels does not affect the coding performance of the rate control methods.




RESULTS: *images of the corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 3/5/7 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, 1 quality layer, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD - Kakadu v4.5, ROC/CoRD - BOI v1.2*

Graphic 6.8: *Coding performance evaluation of ROC and CoRD compared to PCRD when using different DWT levels.*

When encoding in lossless mode, the quantization stage does not weight the subbands, therefore CPI, ROC and CoRD must simulate the subband weighting like for instance using the techniques proposed by Bilgin et al. in [14], or by Long et al. in [59]. However, the results are not as good as when the subband is weighted appropriately, as we can see in Graphic 6.9. Besides, CoRD does not achieve the best results and, for some bit-rate, CPI is even better than ROC and CoRD. These results would probably be improved changing the values of $\mathcal{F}_{init}$, $\mathcal{F}_{inc}$ and $\mathcal{F}_{dec}$ for the rate control method CoRD, although it is clear that an in-depth research should be performed.




*Av. PSNR diff. - CPI: -0.307 dB (0.001 to 2 bps)*
*Av. PSNR diff. - ROC: -0.249 dB (0.001 to 2 bps)*
*Av. PSNR diff. - CoRD: -0.291 dB (0.001 to 2 bps)*
*Av. PSNR diff. - 20 logarithmic quality layers: -0.141 dB (0.001 to 2 bps)*
*Av. PSNR diff. - 40 equivalent (3 ranges) quality layers: -0.065 dB (0.001 to 2 bps)*

RESULTS: *average of corpus ISO 12640-1, gray scaled, size 2048x2560. 600 control points.*
JPEG2000 CODING PARAMS.: *5/3 DWT 5 levels, 64x64 code-blocks, no precincts, RESTART coding variation, LRCP progression. Lossless compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI/ROC/CoRD - BOI v1.2*

Graphic 6.9: *Coding performance evaluation of CPI, ROC, CoRD and the best strategies of quality layers allocation compared to PCRD for lossless compression.*

### 6.4.3 Other image corpora

We evaluate now the coding performance of the developed rate control methods using two different image corpus, comprised of images belonging to the medical community and to the remote sensing community. The corpus of medical images is compounded by six monochrome (8 bps) images: one computed radiology, two mammographies and three radiologies, depicted in Figure 6.3. The corpus of remote sensing images is compounded by five monochrome images scaled to 8 bps: three images have been acquired in February 8, 2006 through the Thematic Mapper (TM) sensor of a Landsat-5 satellite, and two images have been acquired in June 27, 2006 through the Haute Resolution Geometrique (HRG) sensor of a Satellite Pour l'Observation de la Terre (SPOT-5). We have selected different image regions including vegetation, cultivated areas and urban areas, depicted in Figure 6.4.

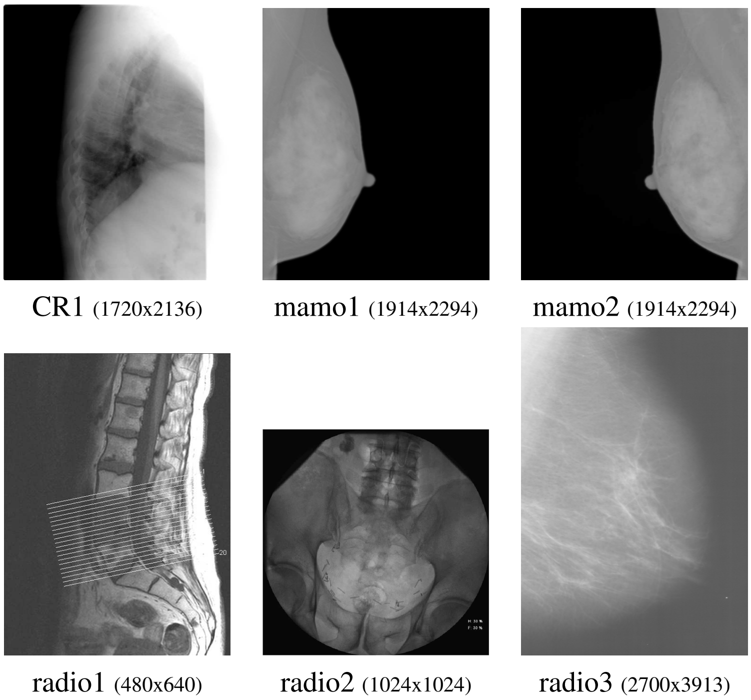



Figure 6.3: *Images of the medical corpus.*

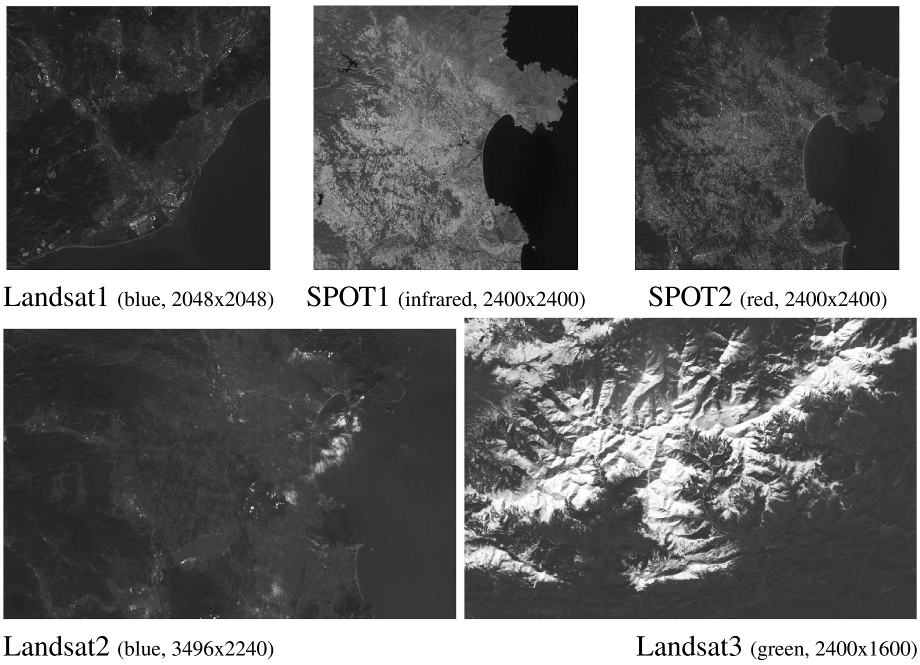
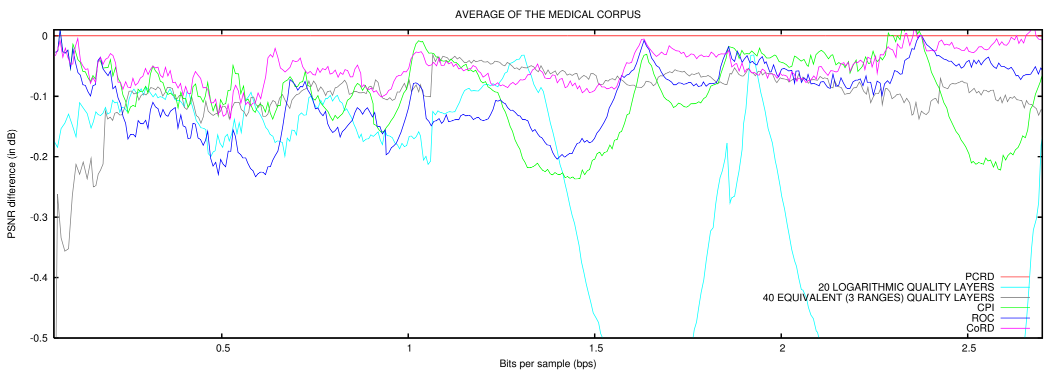



Figure 6.4: *Images of the remote sensing corpus.*

In order to rapidly evaluate the coding performance of the rate control methods introduced in this thesis we compute an average of all the images for each corpus, although the images are acquired with different devices and have different sizes. Graphic 6.10 depicts the average results obtained for the medical corpus. Surprisingly, CPI obtains better results than ROC, specially at low bit-rates. We believe this is caused due to a wrong change from the first scanning order to the second one. Graphic 6.11 depicts the average results for the remote sensing corpus. Again, ROC presents a bad coding performance, which is outperformed by CPI and CoRD. Surprisingly, CPI is slightly better than CoRD from 0.001 to 1 bps, although the difference is only 0.001 dB.

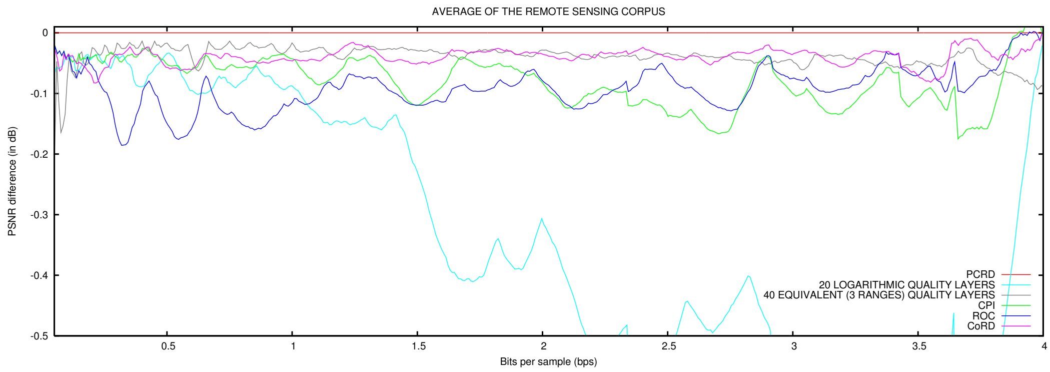
These results shows the reliability of CoRD, since it usually improves, or at least maintains, the coding performance of CPI and/or ROC. As well as for the corpus ISO 12640-1, in these image corpora CoRD achieves a better coding performance than the one obtained with the analyzed strategies of quality layers allocation.




*Av. PSNR diff. - CPI: -0.093 dB (0.001 to 1 bps) -0.097 dB (total)*
*Av. PSNR diff. - ROC: -0.137 dB (0.001 to 1 bps) -0.104 dB (total)*
*Av. PSNR diff. - CoRD: -0.069 dB (0.001 to 1 bps) -0.053 dB (total)*
*Av. PSNR diff. - 20 logarithmic quality layers: -0.136 dB (0.001 to 1 bps) -0.377 dB (total)*
*Av. PSNR diff. - 40 equivalent (3 ranges) quality layers: -0.135 dB (0.001 to 1 bps) -0.096 dB (total)*

RESULTS: *average of medical corpus, gray scaled images, different sizes. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI/ROC/CoRD - BOI v1.2*

Graphic 6.10: *Coding performance evaluation of CPI, ROC, CoRD and the best strategies of quality layers allocation compared to PCRD for a corpus of medical images.*




*Av. PSNR diff. - CPI: -0.044 dB (0.001 to 1 bps) -0.079 dB (total)*
*Av. PSNR diff. - ROC: -0.115 dB (0.001 to 1 bps) -0.090 dB (total)*
*Av. PSNR diff. - CoRD: -0.045 dB (0.001 to 1 bps) -0.039 dB (total)*
*Av. PSNR diff. - 20 logarithmic quality layers: -0.067 dB (0.001 to 1 bps) -0.377 dB (total)*
*Av. PSNR diff. - 40 equivalent (3 ranges) quality layers: -0.065 dB (0.001 to 1 bps) -0.047 dB (total)*

RESULTS: *average of remote sensing corpus, gray scaled images, different sizes. 600 control points.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, no precincts, RESTART coding variation, LRCP progression. Lossy compression. Targeted bit-rates and single quality layer for PCRD.*
IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI/ROC/CoRD - BOI v1.2*

Graphic 6.11: *Coding performance evaluation of CPI, ROC, CoRD and the best strategies of quality layers allocation compared to PCRD for a corpus of remote sensing images.*

## 6.4.4 Extraction of WOIs

The experiments performed in this section are devised to assess the coding performance that can be achieved with the use of quality layers compared to CPI, ROC and CoRD when a Window Of Interest (WOI) is extracted from a code-stream. When using quality layers, the extraction of the WOI is performed at quality layers boundaries, assessing the best coding performance that can be obtained with quality layers. The image SPOT1 from the corpus of remote sensing images has been encoded to a code-stream containing 10 quality layers. Three WOIs of different sizes have been decoded at the quality layers boundaries, and the PSNR of the extracted WOI compared to the original one has been computed. These results are showed in the column LAYER of Table 6.7. The bit-rate of the decoded code-stream when extracting the WOI at the quality layer boundary has been used to extract the same WOI using CPI, ROC and CoRD at the same bit-rate, but from a code-stream containing a single quality layer. The recovered WOIs have been compared to the original ones in terms of PSNR. The results are showed in the remaining columns of Table 6.7, emphasizing in bold font the best results.

Although the differences among the use of quality layers and the proposed rate control methods are, in general, small, it is worth noting that CoRD achieves the best results in most cases. ROC presents some variations and although it is, in general, better than the use of quality layers, in some cases it is outperformed by CPI. Note also that the greater the size of the WOI is, better the coding performance achieved by the use of quality layers is.

Table 6.7: *Coding performance evaluation of CPI, ROC, CoRD and the quality layers allocation when extracting WOIs from a code-stream.*

| | *WOI size*: 256x256 | | | | *WOI size*: 512x512 | | | | *WOI size*: 1024x1024 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LAYER | CPI | ROC | CoRD | LAYER | CPI | ROC | CoRD | LAYER | CPI | ROC | CoRD |
| 0.025 bps | **28.68** | **28.68** | **28.68** | **28.68** | 27.20 | 27.20 | 27.20 | **27.24** | **25.89** | 25.84 | 25.84 | **25.89** |
| 0.1 bps | 31.10 | **31.18** | **31.18** | **31.18** | 30.26 | 30.33 | 30.33 | **30.38** | 29.35 | **29.41** | **29.41** | 29.40 |
| 0.25 bps | 33.57 | **33.66** | **33.66** | **33.66** | 32.94 | 32.99 | 32.99 | **33.00** | 32.31 | 32.34 | 32.34 | **32.41** |
| 0.5 bps | 36.01 | 36.12 | **36.14** | 36.07 | 35.50 | **35.60** | 35.51 | **35.60** | 34.97 | 35.03 | 34.93 | **35.04** |
| 0.75 bps | 37.58 | 37.51 | 37.58 | **37.61** | 37.18 | 37.20 | 37.22 | **37.29** | 36.63 | 36.68 | 36.64 | **36.77** |
| 2 bps | 44.67 | 44.81 | 44.75 | **45.03** | 44.69 | 44.91 | 44.83 | **44.95** | 45.11 | 45.11 | 45.14 | **45.22** |

LEGEND: *The first column represents the bit-rate at the quality layer boundary. The coding performance is expressed in dB in each cell.*
RESULTS: *image SPOT1 of the remote sensing corpus, gray scaled, size 2400x2400.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 32x32 code-blocks, no precincts, RESTART coding variation, LRCP progression. Lossy compression.*
IMPLEMENTATION: *BOI v1.2*

### 6.4.5 Computational complexity reduction of the coder

When CPI, ROC, or CoRD are used in the encoding process, the PCRD stage defined in EBCOT is completely unnecessary. This implies that it is not needed to compute the distortion of each coefficient, the rate-distortion slopes are no longer required, and the application of the Lagrange multiplier is not used. However, the optimal PCRD method takes only about 5% of the complete encoding process, and although for CPI, ROC and CoRD this percentage is even smaller, from this point of view, the overall computational complexity reduction is meaningless.

The main advantage of CPI, ROC and CoRD is that they perform the encoding of coding passes and the rate control simultaneously. This implies that only the coding passes included in the final code-stream are encoded. In terms of computational complexity, CPI, ROC and CoRD achieve the same speed-up because they just encode those coding passes included in the final code-stream.

Table 6.8: *Computational complexity evaluation of CPI/ROC/CoRD compared to PCRD in the encoding process.*

| *bps* | *CODING TIME* | *SPEED UP* | | *#CODING PASSES ENCODED* | *INCLUDED* |
|---|---|---|---|---|---|
| 0.0625 | 6 % | 16.6 | PCRD: | 23307 | 1387 |
| | | | CoRD: | 1431 | 1430 |
| 0.125 | 10 % | 10 | PCRD: | 23307 | 2550 |
| | | | CoRD: | 2552 | 2551 |
| 0.25 | 15 % | 6.6 | PCRD: | 23307 | 4508 |
| | | | CoRD: | 4309 | 4308 |
| 0.5 | 25 % | 4 | PCRD: | 23307 | 7136 |
| | | | CoRD: | 6697 | 6696 |
| 1 | 35 % | 2.9 | PCRD: | 23307 | 10507 |
| | | | CoRD: | 10205 | 10204 |
| 2 | 50 % | 2 | PCRD: | 23307 | 15126 |
| | | | CoRD: | 14679 | 14678 |

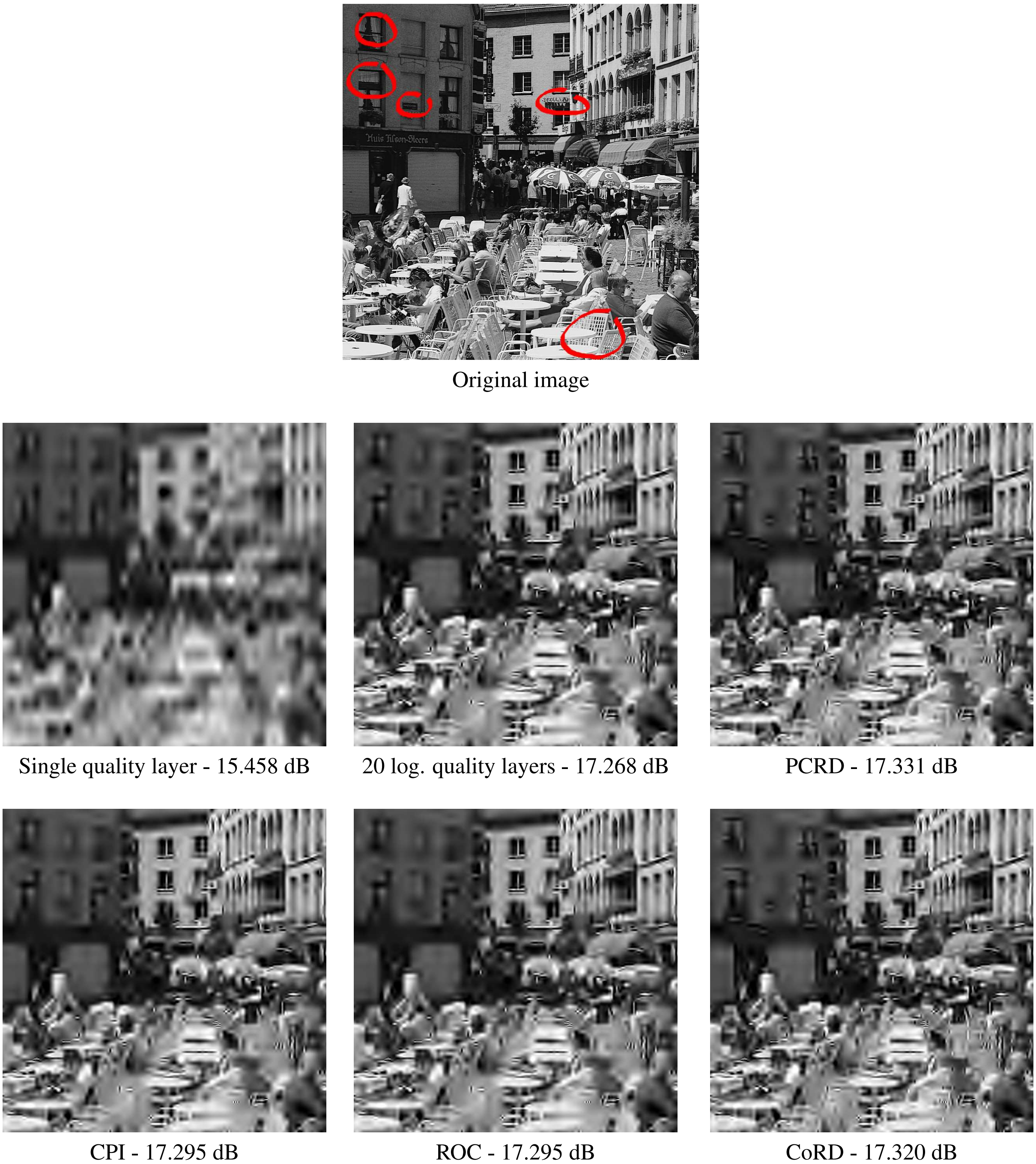
RESULTS: *average of corpus ISO 12640-1, gray scaled, size 2048x2560.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, RESTART coding variation. Lossy compression. Targeted bit-rates.*
IMPLEMENTATION: *BOI v1.2*

Table 6.8 provides the average encoding time for the images of the corpus ISO 12640-1 at different bit-rates. Column *CODING TIME* shows the percentage time needed by the sample date coding stage of BOI using CoRD, compared to the time spent by the PCRD method ($\frac{CoRD}{PCRD}$). The speed-up factor is also given: for very large compression factors (very low bit-rate), CoRD provides the best improvements. Considering that the encoding stage represents more than 60% of the whole compression process [21], CPI/ROC/CoRD contribute to reduce the JPEG2000 computational complexity efficiently. The average number of coding passes encoded for the optimal PCRD method and CoRD, and the number of coding passes included in the final codestream, are also given.

### 6.4.6 Visual comparison

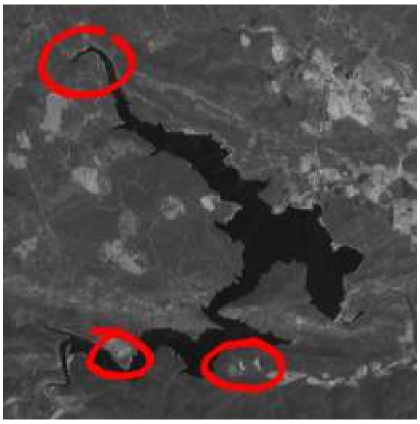
Regarding the qualitative analysis, Figures 6.5, 6.6, 6.7 provide a visual comparison of different regions of three images encoded at different bit-rates. The areas where we observe the largest differences are emphasized by a red line in the original image.

At very low bit-rates, some differences are noticed among the proposed rate control methods and the use of quality layers. Note that, in the Cafeteria image, small details of the windows are only observed in the PCRD and CoRD images. However at low and medium bit-rates, the differences are practically not noticeable. The largest differences occur with the single quality layer code-stream, which contains high degradations.

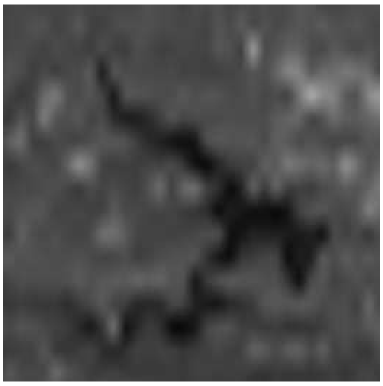



BIT-RATE: *0.015625 bps (compression factor 512:1)* AREA: *828x828*
RESULTS: *Cafeteria image of the corpus ISO 12640-1, gray scaled, size 2048x2560.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, RESTART coding variation. Lossy compression. Targeted bit-rate for PCRD.*
IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI/ROC/CoRD - BOI v1.2*

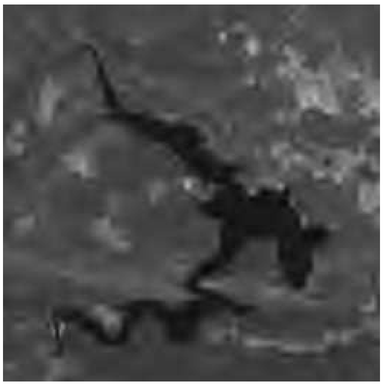
Figure 6.5: *Visual comparison among CPI, ROC, CoRD and the use of quality layers at a very high compression factor.*



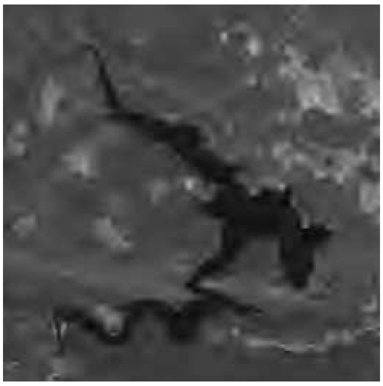
Original image



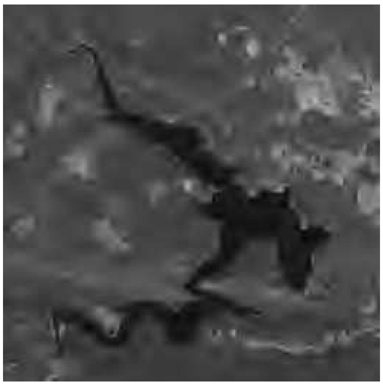
Single quality layer - 25.505 dB



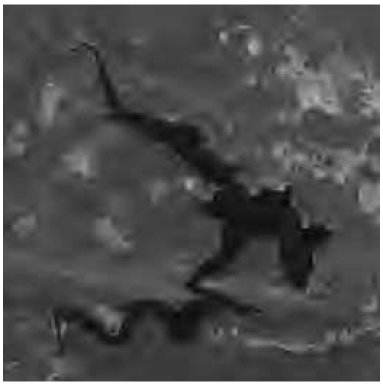
20 log. quality layers - 28.722 dB



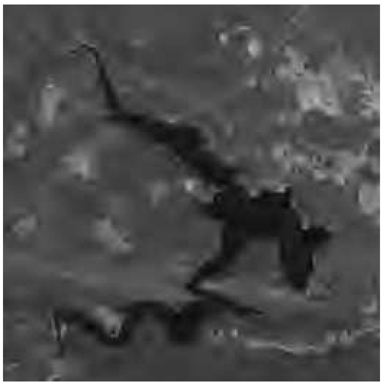
PCRD - 28.785 dB



CPI - 27.779 dB



ROC - 28.779 dB



CoRD - 28.781 dB

BIT-RATE: *0.0625 bps (compression factor 128:1)* AREA: *260x260*
RESULTS: *SPOT1 image of the remote sensing corpus, gray scaled, size 2400x2400.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, RESTART coding variation. Lossy compression. Targeted bit-rate for PCRD.*
IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI/ROC/CoRD - BOI v1.2*

Figure 6.6: *Visual comparison among CPI, ROC, CoRD and the use of quality layers at a high compression factor.*

Original image

PCRD - 41.550 dB

Single quality layer - 33.383 dB

20 log. quality layers - 41.491 dB

CPI - 41.487 dB

ROC - 41.483 dB

CoRD - 41.513 dB

BIT-RATE: *0.5 bps (compression factor 16:1)* AREA: *1800x615*
RESULTS: *Orchid image of the corpus ISO 12640-1, gray scaled, size 2048x2560.*
JPEG2000 CODING PARAMS.: *9/7 DWT 5 levels, derived quantization, 64x64 code-blocks, RESTART coding variation. Lossy compression. Targeted bit-rate for PCRD.*
IMPLEMENTATIONS: *PCRD/LAYERS - Kakadu v4.5, CPI/ROC/CoRD - BOI v1.2*

Figure 6.7: *Visual comparison among CPI, ROC, CoRD and the use of quality layers at a medium compression factor.*

# Chapter 7

# Conclusions

## 7.1 Summary and discussion

Quality scalability is an essential mechanism of image coding systems. Combined with other mechanisms, it allows the extraction of windows of interest (WOI) or the complete image from a code-stream at different bit-rates, achieving the same quality as if the image was encoded at those bit-rates. This is fundamental to interactively transmit the image over the network or to allow the truncation of the code-stream.

In JPEG2000, quality scalability is achieved embedding quality layers in the code-stream through the rate control method used in the encoding process. The allocation strategy determines the number and bit-rate of quality layers, and can be evaluated in terms of coding performance by decoding code-stream segments at different bit-rates. The optimization of the coding performance achieved by the quality layers allocation is studied in [107], concluding that the rate control method PCRD, defined in EBCOT [93] and used as a guideline in the JPEG2000 standard, already optimizes the quality scalability of a code-stream.

However, the use of quality layers might cause two drawbacks. On the one hand, already encoded code-streams containing a single or few quality layers have a lack of quality scalability, which may penalize the coding performance by at most 10 dB when the code-stream has to be truncated or interactively transmitted. On the other hand, the efficiency achieved with the use of quality layers when transmitting WOIs is not optimal and it can be widely improved, as is described in [98, 99] when using the JPIP protocol defined in the Part 9 of the standard [46].

The purpose of this thesis is to develop a rate control method that provides quality scalability for WOIs, or the complete image, even if the code-stream contains a single or few quality layers. First we evaluate the rate control methods proposed in the literature in order to disclose if some of them could be used to achieve our purpose. The main difficulty is that when the code-stream is already encoded, the original image is not available and therefore the rate control can not use distortion measures based on it. Given this restriction, we classify the rate control methods found in the literature in two main categories: deterministics and model-based, distinguishing whether if they use measures based on the original image or not.

Almost all the rate control methods are deterministic or based on mixed approaches. The best results in terms of coding performance are obtained by deterministic methods [93, 17, 95, 116, 110, 26, 49], while in terms of computational complexity the best results are obtained by mixed approaches [110, 103, 116]. However, these rate control methods can not be used to achieve our purpose due to the use of measures based on the original image. Besides, the single model-based rate control method which does not use any measure based on the original image [65], suitable for our purposes, penalizes the coding performance more than 1 dB when it is compared to deterministic rate control methods, restraining its use.

We propose three rate control methods devised to supply quality scalability to code-streams, without regard to the number of quality layers they contain. The first one is named Coding Passes Interleaving (CPI) and is based on a rate-distortion model implicitly used in classical coding systems such as [83, 77], interleaving coding passes from the highest bit-plane to the lowest bit-plane. The main drawback of CPI is that it obtains an irregular coding performance that is, in some cases, 0.5 dB worse than the one obtained with the optimal PCRD method.

The second rate control method is based on a Reverse subband scanning Order and coding passes Concatenation (ROC), which usually enhances the results of CPI achieving a regular coding performance among bit-rates. The main idea behind ROC is to use an interleaving-based algorithm similar to CPI but changing the scanning order followed by the algorithm under some circumstances. The main disadvantage of ROC is that the rough change on the scanning orders penalize, in some cases, the coding performance. This causes that, for some corpus of images, ROC does not improve the results of CPI.

The third rate control method is based on the Characterization of the Rate-Distortion slope of code-blocks (CoRD). The approach of CoRD is different from CPI and ROC, using a novel rate-distortion model motivated by the results of CPI and ROC, and derived from assumptions based on [103]. This model uses a special characteristic of the rate-distortion of code-blocks within a subband that we name balloon effect. The balloon effect exposes the distribution of the significant coefficients encoded at the different coding passes of a code-block, considering code-blocks within a subband. This allows a fair estimation of the rate-distortion slope and, therefore, the achievement of a competitive coding performance that improves the results obtained with CPI and ROC. Compared to the use of quality layers when decoding segments of a code-stream, CoRD is slightly better than the best allocation strategy evaluated in our experimental results. When extracting WOIs from a code-stream, CoRD also obtains better results than the ones obtained with the use of quality layers.

CoRD has several applications: it can embed quality layers to a code-stream, it can change the number and allocation of quality layers, it can truncate a single quality layer code-stream without penalizing its quality, or it can extract WOIs at different bit-rates. Because CoRD has negligible costs in terms of computational complexity, it can be used to control interactive image transmissions, even if the code-stream contains a single or few quality layers, enhancing the quality of the retrieved WOIs. The main drawback of CoRD is that, in order not to decode any segment of the code-stream, it needs the RESTART coding variation of JPEG2000, which allows the identification of coding passes lengths.

Besides, CoRD can also be used in the encoding process reducing the computational complexity of the Tier-1 stage. Only three rate control methods [116, 110, 103] achieve speed-ups as high as the one obtained with CoRD. However, CoRD needs to maintain the image in memory, along with some status map of the MQ-coder, like the methods proposed in [116, 110]. This could restrict its use to applications without constrained memory resources.

From the point of view of our classification, CoRD is a model-based rate control method, and compared to deterministic rate control methods, it obtains a very high competitive results in terms of coding performance and in terms of computational complexity. For the three image corpus evaluated in this thesis, CoRD is, in average, only 0.05 dB worse than the optimal PCRD method.

---

In addition to all the development presented in this thesis, we realize that in many applications the use of quality layers is already a good (or the best) solution. However, we hope that in some cases CoRD can be useful. We like to think that it provides a new mechanism to the standard.

## 7.2 Future work

Our study and development of rate control methods devised to supply quality scalability to already encoded code-streams is likely finished with the development of CoRD. However, this method should be enhanced to improve the results achieved in lossless compression and it should be implemented to manage three-dimensional data sets. Apart from this, our future work is focused on two different purposes: the application of CoRD to different scenarios, and the study of two issues that have appeared throughout the development of this thesis. The following points summarize them:

- Applications of CoRD:

1. *CoRD applied to JPIP*: CoRD provides an excellent framework to extract WOIs from a code-stream, achieving competitive results in terms of coding performance. We think that its application to interactive image transmissions would probably obtain similar results to the ones obtained in [98, 99].

2. *CoRD applied to region of interest coding*: although Part 1 [42] and Part 2 [44] of the standard already supply methods of Region Of Interest (ROI) coding, several variations have appeared in the literature proposing slightly modifications to these methods, for instance in [104, 56, 86, 55]. Some other methods of ROI coding are based on the prioritization of code-blocks through the use of quality layers [79, 109]. CoRD also provides a suitable algorithm to encode ROIs in a similar way.

- Related issues:

1. *Flexible order of coding passes*: from our point of view, one of the most surprising issues of ROC and CoRD methods is that they concatenate the coding passes of type MRP with the coding passes of type CP. It would be interesting to see if a better coding performance is achieved when the coding system allows a flexible order of the coding passes, this is, instead of concatenating coding passes of type MRP with CP, swap their order.

2. *Evaluation of CoRD with distortion measures different than MSE*: throughout the development of this thesis and in all the experimental sections, we always use the MSE as the distortion measure. However, in the literature there appear papers focused on the optimization of the visual quality [58] rather than on the MSE. It would be interesting to evaluate CoRD using visual distortion measures.

# Bibliography


[1] M. Adams. (2006, Oct.) JasPer Project Home Page. [Online]. Available: http://www.ece.uvic.ca/˜mdadams/jasper

[2] M. Aguzzi, M. G. Albanesi, and M. Martelli, "A new algorithm for the embedding of a prediction mechanism into the JPEG2000 coding chain," *ELSEVIER Pattern Recognition*, vol. 39, pp. 1492–1508, June 2006.

[3] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[4] A. Aminlou and O. Fatemi, "Very fast bit allocation algorithm, based on simplified R-D curve modeling," presented at the Proc. IEEE International Conference on Electronics, Circuits and Systems, vol. 1, Dec. 2003, pp. 14–17.

[5] A. Aminlou and O. Fatemi, "A novel efficient rate control algorithm for hardware implementation in JPEG2000," presented at the Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, Mar. 2005, pp. 21–24.

[6] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[7] E. Ardizzone, M. L. Cascia, and F. Testa, "A new algorithm for bit rate allocation in JPEG2000 tile encoding," presented at the Proc. IEEE International Conference on Image Analysis and Processing, Sept. 2003, pp. 658–661.

[8] J. Askelof, M. L. Carlander, and C. Christopoulos, “Region of interest coding in JPEG 2000,” *ELSEVIER Signal Processing: Image Communication*, vol. 17, no. 1, pp. 105–111, Jan. 2002.

[9] F. Auli-Llinas, J. Bartrina-Rapesta, J. R. Paton-Martin, J. Monteagudo-Pereira, and J. Serra-Sagrista, “J2K: Introducing a novel JPEG2000 coder,” presented at the Proc. SPIE International Conference on Visual Communications and Image Processing, July 2005, pp. 1763–1773.

[10] F. Auli-Llinas, J. Monteagudo-Pereira, J. Bartrina-Rapesta, J. Paton-Martin, and J. Serra-Sagrista, “Novel JPEG2000 implementation applied to GIS scenarios,” presented at the Proc. SPIE International Conference on Satellite Data Compression, Communications and Archiving, Aug. 2005, pp. 159–168.

[11] F. Auli-Llinas and J. Serra-Sagrista, “Low complexity JPEG2000 rate control through reverse subband scanning order and coding passes concatenation,” *IEEE Signal Processing Letters*, Apr. 2007, to be published.

[12] F. Auli-Llinas, J. Serra-Sagrista, J. L. Monteagudo-Pereira, and J. Bartrina-Rapesta, “Efficient rate control for JPEG2000 coder and decoder,” presented at the Proc. IEEE Data Compression Conference, Mar. 2006, pp. 282–291.

[13] S. Battiato, A. Buemi, G. Impoco, and M. Mancuso, “JPEG2000 coded images optimization using a content-dependent approach,” *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, pp. 400–408, Aug. 2002.

[14] A. Bilgin, J. Sementilli, F. Sheng, and M. Marcellin, “Scalable image coding using reversible integer wavelet transforms,” *IEEE Transactions on Image Processing*, vol. 9, no. 11, pp. 1972–1977, Nov. 2000.

[15] Center for Ecological Research and Forestry Applications. (2006, Oct.) CREAF. Cerdanyola del Valles 08290 (Spain). [Online]. Available: http://www.creaf.uab.es

[16] V. Chande and N. Farvardin, “Joint source-channel coding for progressive transmission of embedded source coders,” presented at the Proc. IEEE Data Compression Conference, Mar. 1999, pp. 52–61.

[17] T.-H. Chang, L.-L. Chen, C.-J. Lian, H.-H. Chen, and L.-G. Chen, “Computation reduction technique for lossy JPEG2000 encoding through EBCOT tier-2 feedback processing,” presented at the Proc. IEEE International Conference on Image Processing, vol. 3, June 2002, pp. 85–88.

[18] T.-H. Chang, C.-J. Lian, H.-H. Chen, J.-Y. Chang, and L.-G. Chen, “Effective hardware-oriented technique for the rate control of JPEG2000 encoding,” presented at the Proc. IEEE International Symposium on Circuits and Systems, vol. 2, May 2003, pp. 684–687.

[19] Y.-W. Chang, H.-C. Fang, C.-J. Lian, and L.-G. Chen, “Novel pre-compression rate-distortion optimization algorithm for JPEG 2000,” presented at the Proc. SPIE International Conference on Visual Communications and Image Processing, Jan. 2004, pp. 1353–1361.

[20] F. Chebil and R. Kurceren, “Pre-compression rate allocation for JPEG2000 encoders in power constrained devices,” presented at the Proc. SPIE International Conference on Image and Video Communications and Processing, Jan. 2006.

[21] K.-F. Chen, C.-J. Lian, H.-H. Chen, and L.-G. Chen, “Analysis and architecture design of EBCOT for JPEG-2000,” presented at the Proc. IEEE International Symposium on Circuits and Systems, vol. 2, May 2001, pp. 765–768.

[22] P. A. Chou, T. Lookabaugh, and R. M. Gray, “Optimal pruning with applications to tree-structured source coding and modeling,” *IEEE Transactions on Information Theory*, vol. 35, no. 2, pp. 299–315, Mar. 1989.

[23] C. Chrysafis and A. Ortega, "Line-based, reduced memory, wavelet image compression," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 378–389, Mar. 2000.

[24] J. C. Dagher, A. Bilgin, and M. W. Marcellin, "Resource-constrained rate control for motion JPEG2000," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1522–1529, Dec. 2003.

[25] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *The Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 245–267, 1998.

[26] W. Du, J. Sun, and Q. Ni, "Fast and efficient rate control approach for JPEG2000," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 4, pp. 1218–1221, Nov. 2004.

[27] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 230–239, Apr. 2004.

[28] M. Dyer, D. Taubman, S. Nooshabadi, and A. K. Gupta, "Concurrency techniques for arithmetic coding in JPEG2000," *IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications*, vol. 53, no. 6, pp. 1203–1212, June 2006.

[29] EPFL, C. R. France, and Ericsson. (2006) Java implementation of JPEG 2000. [Online]. Available: http://jpeg2000.epfl.ch

[30] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, pp. 399–417, 1963.

[31] H.-C. Fang, Y.-W. Chang, T.-C. Wang, C.-J. Lian, and L.-G. Chen, "Parallel embedded block coding architecture for JPEG 2000," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 9, pp. 1086–1097, Sept. 2005.

[32] T. J. Flohr, M. W. Marcellin, and J. C. Rountree, “Scan-based processing with JPEG-2000,” presented at the Proc. SPIE Applications of Digital Image Processing XXIII, vol. 4115, Dec. 2000, pp. 347–355.

[33] J. E. Fowler. (2006, Apr.) QccPack - Quantization, Compression, and Coding library. [Online]. Available: http://qccpack.sourceforge.net

[34] D. L. Gall and A. Tabatabai, “Subband coding of digital images using symmetric kernel filters and arithmetic coding techniques,” presented at the Proc. IEEE International Conference Acoustics, Speech, and Signal Processing, Apr. 1988, pp. 761–764.

[35] M. Grangetto, E. Magli, and G. Olmo, “A syntax-preserving error resilience tool for JPEG 2000 based on error correcting arithmetic coding,” *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 807–818, Apr. 2006.

[36] Group on Interactive Coding of Images. (2006, Oct.) BOI software. [Online]. Available: http://www.gici.uab.es/BOI

[37] A. K. Gupta, S. Nooshabadi, D. Taubman, and M. Dyer, “Realizing low-cost high-throughput general-purpose block encoder for JPEG2000,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 843–858, July 2006.

[38] J. S. Houchin and D. W. Singer, “File format technology in JPEG 2000 enables flexible use of still and motion sequences,” *ELSEVIER Signal Processing: Image Communication*, vol. 17, no. 1, pp. 131–144, Jan. 2002.

[39] *Digital compression and coding for continuous-tone still images*, ISO/IEC Std. 10 918-1, 1994.

[40] *Graphic technology - Prepress digital data exchange - Part 1: CMYK standard colour image data (CMYK/SCID)*, ISO/IEC Std. 12 640-1, 1997.

[41] *JPEG-LS lossless and near-lossless compression for continuous-tone still images*, ISO/IEC Std. 14 495-1, 1999.

[42] *Information technology - JPEG 2000 image coding system - Part 1: Core coding system*, ISO/IEC Std. 15 444-1, Dec. 2000.

[43] *Information technology - Lossy/lossless coding of bi-level images*, ISO/IEC Std. 14 492, 2001.

[44] *Information technology - JPEG 2000 image coding system - Part 2: Extensions*, ISO/IEC Std. 15 444-2, Apr. 2004.

[45] *Information technology - Coding of audio-visual objects: ISO base media file format*, ISO/IEC Std. 14 496-12, June 2005.

[46] *Information technology - JPEG 2000 image coding system: Interactivity tools, APIs and protocols*, ISO/IEC Std. 15 444-9, Dec. 2005.

[47] Joint Photographic Experts Group (JPEG). (2006, Oct.) JPEG homepage. [Online]. Available: http://www.jpeg.org

[48] JPEG, "Call for contributions for JPEG 2000 (JTC 1.29.14, 15444): Image coding system," ISO/IEC JTC1/SC29/WG1, Tech. Rep. N505, Mar. 1997.

[49] T. Kim, H. M. Kim, P.-S. Tsai, and T. Acharya, "Memory efficient progressive rate-distortion algorithm for JPEG 2000," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 181–187, Jan. 2005.

[50] O. M. Kosheleva, B. E. Usevitch, S. D. Cabrera, and E. Vidal, "Rate distortion optimal bit allocation methods for volumetric data using JPEG 2000," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2106–2112, Aug. 2006.

[51] P. Kulkarni, A. Bilgin, M. W. Marcellin, J. C. Dagher, T. Flohr, and J. Rountree, "Reduced memory multi-layer multi-component rate allocation for JPEG2000," presented at the Proc. SPIE International Conference on Image and Video Communications and Processing, Jan. 2005, pp. 139–150.

[52] D. T. Lee, "JPEG 2000: Retrospective and new developments," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 32–41, Jan. 2005.

[53] J. Li and S. Lei, "An embedded still image coder with rate-distortion optimization," *IEEE Transactions on Image Processing*, vol. 8, no. 7, pp. 913–924, July 1999.

[54] X. Li, "Improved wavelet decoding via set theoretic estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 108–112, Jan. 2005.

[55] Z. Li-bao, "Region of interest image coding using IWT and partial bitplane block shift for network applications," presented at the Proc. IEEE International Conference on Computer and Information Technology, Sept. 2005, pp. 624–628.

[56] L. Liu and G. Fan, "A new JPEG2000 region-of-interest image coding method: Partial significant bitplanes shift," *IEEE Signal Processing Letters*, vol. 10, no. 2, pp. 35–38, Feb. 2003.

[57] Z. Liu and L. J. Karam, "Mutual information-based analysis of JPEG2000 contexts," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 411–422, Aug. 2005.

[58] Z. Liu, L. J. Karam, and A. B. Watson, "JPEG2000 encoding with perceptual distortion control," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1763–1778, July 2006.

[59] M. Long, H.-M. Tai, and S. Yang, "Quantization step selection schemes in JPEG2000," *IEE Electronics Letters*, vol. 38, no. 12, pp. 547–549, June 2002.

[60] T. Lu and P. Chang, "Significant bit-plane clustering technique for JPEG2000 image coding," *IEE Electronics Letters*, vol. 40, no. 17, pp. 1056–1058, Aug. 2004.

[61] S. Mallat, “A theory of multiresolution signal decomposition: The wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, July 1989.

[62] M. Marcellin and D. Taubman, Private communication, June 2005.

[63] M. W. Marcellin, M. A. Lepley, A. Bilgin, T. J. Flohr, T. T. Chinen, and J. H. Kasner, “An overview of quantization in JPEG 2000,” *ELSEVIER Signal Processing: Image Communication*, vol. 17, no. 1, pp. 73–84, Jan. 2002.

[64] T. Masuzaki, H. Tsutsui, T. Izumi, T. Onoye, and Y. Nakamura, “Adaptive rate control for JPEG2000 image coding in embedded systems,” presented at the Proc. IEEE International Conference on Image Processing, vol. 3, June 2002, pp. 77–80.

[65] T. Masuzaki, H. Tsutsui, T. Izumi, T. Onoye, and Y. Nakamura, “JPEG2000 adaptive rate control for embedded systems,” presented at the Proc. IEEE International Symposium on Circuits and Systems, vol. 4, May 2002, pp. 333–336.

[66] A. Nguyen, V. Chandran, S. Sridharan, and R. Prandolini, “Interpretability performance assessment of JPEG2000 and Part 1 compliant region of interest coding,” *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 808–817, Nov. 2003.

[67] C. Parisot, M. Antonini, and M. Barlaud, “High performance coding using a model-based bit allocation with EBCOT,” presented at the Proc. of EURASIP XI European Signal Processing Conference, Sept. 2002.

[68] C. Parisot, M. Antonini, and M. Barlaud, “Scan-based quality control for JPEG2000 using R-D models,” presented at the Proc. of EURASIP XI European Signal Processing Conference, Sept. 2002.

[69] C. Parisot, M. Antonini, and M. Barlaud, “Stripe-based MSE control in image coding,” presented at the Proc. IEEE International Conference on Image Processing, vol. 2, Sept. 2002, pp. 649–652.

[70] G. Pastuszak, "A high-performance architecture for embedded block coding in JPEG 2000," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 9, pp. 1182–1191, Sept. 2005.

[71] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 11, pp. 1219–1235, Nov. 2004.

[72] K. Peng and J. C. Kieffer, "Embedded image compression based on wavelet pixel classification and sorting," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1011–1017, Aug. 2004.

[73] W. Pennebaker and J. Mitchell, *JPEG still image data compression standard.* New York: Van Nostrand Reinhold, 1993.

[74] R. Puri and K. Ramchandran, "Multiple description source coding through forward error correction codes," presented at the Proc. 33rd Asilomar Conference on Signals, Systems and Computers, vol. 1, Oct. 1999, pp. 342–346.

[75] X. Qin, X.-L. Yan, X. Zhao, C. Yang, and Y. Yang, "A simplified model of delta-distortion for JPEG2000 rate control," presented at the Proc. IEEE International Conference on Communications, Circuits and Systems, June 2004, pp. 548–552.

[76] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard," *ELSEVIER Signal Processing: Image Communication*, vol. 17, no. 1, pp. 3–48, Jan. 2002.

[77] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[78] V. Sanchez and M. K. Mandal, "Efficient channel protection for JPEG2000 bitstream," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 4, pp. 554–558, Apr. 2004.

[79] V. Sanchez, A. Basu, and M. K. Mandal, "Prioritized region of interest coding in JPEG2000," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 9, pp. 1149–1155, Sept. 2004.

[80] T. Sanguankotchakorn and J. Fangtham, "A new approach to reduce encoding time in EBCOT algorithm for JPEG2000," presented at the Proc. IEEE Conference on Convergent Technologies for Asia-Pacific Region, vol. 4, Oct. 2003, pp. 1338–1342.

[81] D. Santa-Cruz, R. Grosbois, and T. Ebrahimi, "JPEG 2000 performance evaluation and assessment," *ELSEVIER Signal Processing: Image Communication*, vol. 17, no. 1, pp. 113–130, Jan. 2002.

[82] J. Serra-Sagrista, F. Auli-Llinas, F. Garcia-Vilchez, J. Gonzalez, and P. Guitart, "Review of coding techniques applied to remote sensing," presented at the Proc. International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Sept. 2004, pp. 647–654.

[83] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Image Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[84] P. Sherwood and K. Zeger, "Progressive image coding on noisy channels," *IEEE Signal Processing Letters*, vol. 4, no. 7, pp. 189–191, July 1997.

[85] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 5, pp. 1445–1453, Sept. 1988.

[86] A. Signoroni, F. Lazzaroni, and R. Leonardi, "Exploitation and extension of the region-of-interest coding functionalities in JPEG2000," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 818–823, Nov. 2003.

[87] A. Skodras, C. Christopoulos, and T. Ebrahimi, "JPEG2000: The upcoming still image compression standard," *ELSEVIER Pattern Recognition Letters*, vol. 22, no. 12, pp. 1337–1345, Oct. 2001.

[88] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, Sept. 2001.

[89] M. Slattery and J. Mitchell, "The Qx-coder," *IBM Journal of Research and Development*, vol. 42, no. 6, pp. 767–784, Nov. 1998.

[90] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998.

[91] H.-M. Tai, M. Long, and S. Yang, "New results on rate control in JPEG2000," presented at the Proc. IEEE World Automation Congress, vol. 13, June 2002, pp. 351–355.

[92] D. Taubman, "High performance scalable image compression with EBCOT," presented at the Proc. IEEE International Conference on Image Processing, vol. 3, Oct. 1999, pp. 344–348.

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

[94] D. Taubman, "Remote browsing of JPEG2000 images," presented at the Proc. IEEE International Conference on Image Processing, Sept. 2002, pp. 229–232.

[95] D. Taubman, "Software architectures for JPEG2000," presented at the Proc. IEEE International Conference on Digital Signal Processing, vol. 1, July 2002, pp. 197–200.

[96] D. Taubman. (2006, Oct.) Kakadu software. [Online]. Available: http://www.kakadusoftware.com

[97] D. Taubman, E. Ordentlich, M. Weinberger, and G. Seroussi, "Embedded block coding in JPEG 2000," *ELSEVIER Signal Processing: Image Communication*, vol. 17, no. 1, pp. 49–72, Jan. 2002.

[98] D. Taubman and R. Prandolini, "Architecture, philosophy and performance of JPIP: Internet protocol standard for JPEG2000," presented at the Proc. SPIE International Conference on Visual Communications and Image Processing, June 2003, pp. 791–805.

[99] D. Taubman and R. Rosenbaum, "Rate-distortion optimized interactive browsing of JPEG2000 images," presented at the Proc. IEEE International Conference on Image Processing, Sept. 2003, pp. 765–768.

[100] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image compression fundamentals, standards and practice*. Norwell, Massachusetts 02061 USA: Kluwer Academic Publishers, 2002.

[101] D. S. Taubman and M. W. Marcellin, "JPEG2000: Standard for interactive imaging," *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1336–1357, Aug. 2002.

[102] J. Tian and R. . Wells, "A lossy image codec based on index coding," presented at the Proc. IEEE Data Compression Conference, Mar. 1996, p. 456.

[103] K. Vikram, V. Vasudevan, and S. Srinivasan, "Rate-distortion estimation for fast JPEG2000 compression at low bit-rates," *IEE Electronics Letters*, vol. 41, no. 1, pp. 16–18, Jan. 2005.

[104] Z. Wang and A. C. Bovik, "Bitplane-by-bitplane shift (BbBShift)- a suggestion for JPEG2000 region of interest image coding," *IEEE Signal Processing Letters*, vol. 9, no. 5, pp. 160–162, May 2002.

[105] P. Westerink, J. Biemond, and D. Boekee, "An optimal bit allocation algorithm for sub-band coding," presented at the Proc. IEEE International Conference Acoustics, Speech, and Signal Processing, Apr. 1988, pp. 757–760.

[106] J. W. Woods and T. Naveen, "A filter based bit allocation scheme for subband compression of HDTV," *IEEE Transactions on Image Processing*, vol. 1, no. 3, pp. 436–440, July 1992.

[107] X. Wu, S. Dumitrescu, and N. Zhang, "On multirate optimality of JPEG2000 code stream," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2012–2023, Dec. 2005.

[108] Z. Wu, A. Bilgin, and M. Marcellin, "An efficient joint source-channel rate allocation scheme for JPEG2000 codestreams," presented at the Proc. IEEE Data Compression Conference, Mar. 2003, pp. 113–122.

[109] H. Yang, M. Long, and H.-M. Tai, "Region-of-interest image coding based on EBCOT," *IEE Proceedings- Vision, Image and Signal Processing*, vol. 152, no. 5, pp. 590–596, Oct. 2005.

[110] Y. Yeung and O. C. Au, "Efficient rate control for JPEG2000 image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 335–344, Mar. 2005.

[111] Y. Yeung, O. C. Au, and A. Chang, "An efficient optimal rate control scheme for JPEG2000 image coding," presented at the Proc. IEEE International Conference on Image Processing, vol. 3, Sept. 2003, pp. 761–764.

[112] Y. Yeung, O. C. Au, and A. Chang, "Efficient rate control technique for JPEG2000 image coding using priority scanning," presented at the Proc. IEEE International Conference on Multimedia and Expo, vol. 3, July 2003, pp. 277–280.

[113] Y. Yeung, O. C. Au, and A. Chang, "Successive bit-plane rate allocation technique for JPEG2000 image coding," presented at the Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, Apr. 2003, pp. 261–264.

[114] W. Yu, "Integrated rate control and entropy coding for JPEG 2000," presented at the Proc. IEEE Data Compression Conference, Mar. 2004, pp. 152–161.

[115] W. Yu, J. Fritts, and F. Sun, "An efficient packetization algorithm for JPEG2000," presented at the Proc. IEEE International Conference on Image Processing, vol. 1, Sept. 2002, pp. 208–212.

[116] W. Yu, F. Sun, and J. E. Fritts, "Efficient rate control for JPEG-2000," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 5, pp. 577–589, May 2006.

[117] W. Zeng, S. Daly, and S. Lei, "An overview of the visual optimization tools in JPEG 2000," *ELSEVIER Signal Processing: Image Communication*, vol. 17, no. 1, pp. 85–104, Jan. 2002.

[118] Y.-Z. Zhang and C. Xu, "Analysis and high performance parallel architecture design for EBCOT in JPEG2000," presented at the Proc. IEEE International Conference on Image Processing, vol. 3, Sept. 2005, pp. 996–999.

Francesc Aulí Llinàs

Bellaterra, October 2006