DOCTOR OF PHILOSOPHY

Colour image coding with wavelets and matching pursuit

Ryszard Maciol

2013

Aston University

Colour Image Coding with Wavelets and Matching Pursuit

Ryszard Paweł Macioł

Doctor Of Philosophy

Aston University

April 2012

Aston University

Colour Image Coding with Wavelets and Matching Pursuit

Ryszard Paweł Macioł

Doctor Of Philosophy, 2012

Thesis Summary

This thesis considers sparse approximation of still images as the basis of a lossy compression system. The Matching Pursuit (MP) algorithm is presented as a method particularly suited for application in lossy scalable image coding. Its multichannel extension, capable of exploiting inter-channel correlations, is found to be an efficient way to represent colour data in RGB colour space. Known problems with MP, high computational complexity of encoding and dictionary design, are tackled by finding an appropriate partitioning of an image. The idea of performing MP in the spatio-frequency domain after transform such as Discrete Wavelet Transform (DWT) is explored. The main challenge, though, is to encode the image representation obtained after MP into a bit-stream. Novel approaches for encoding the atomic decomposition of a signal and colour amplitudes quantisation are proposed and evaluated. The image codec that has been built is capable of competing with scalable coders such as JPEG 2000 and SPIHT in terms of compression ratio.

Keywords: Matching Pursuit, Sparse Approximations, Lossy Compression, Colour Image Coding, Wavelets, Run Length Encoding

Contents

List of Figures

List of Tables

Notation and Abbreviations

$\mathcal{H}$	Hilbert vector space
$\mathbb{N}$	set of all natural numbers: $\{1, 2, \ldots\}$
$\langle f, g \rangle$	inner product in $\mathcal{H}$
$\|f\|$	norm (length) of vector in $\mathcal{H}$: $\|f\| = \sqrt{\langle f, f \rangle}$
$dim(\mathcal{H})$	dimension of $\mathcal{H}$
$f \otimes g$	tensor product of two 1D vectors that results in 2D matrix
f, g, g_γ	signals from $\mathcal{H}$
$\mathcal{D} \subset \mathcal{H}$	subset of $\mathcal{H}$
$\lceil a \rceil$	the smallest integer greater than or equal a
$\lfloor a \rfloor$	the greatest integer smaller than or equal a
round(a)	the nearest integer to a rounded up
$\overline{\mathcal{D}}$	closure of set $\mathcal{D}$ in $\mathcal{H}$
a, a_i, c, c_i	real coefficients (amplitudes)
$\mathbf{a}$	vector of amplitudes
$Span(\mathcal{D})$	linear span - a set of all linear combinations of elements from $\mathcal{D}$
sgn(a)	sign function: 1 if $a > 0$, 0 if $a = 0$ and -1 if $a < 0$
$R(D)$	Rate as a function of distortion
AC	Alternating Current
BAC	Binary Arithmetic Coding
BP	Basis Pursuit
bpp	bits per pixel
CIE	Commision Internationale de l'Eclarige
CMYK	Cyan, Magenta, Yellow and BlacK
CR	Compression Ration
CRT	Cathode Ray Tube
CSOT	Composite Spatial Orientation Tree
$\mathbb{C}$WT	Complex Wavelet Transform
DC	Direct Current
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DHT	Discrete Hartley Transform
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimal Truncation

EZW	Embedded Zerotree Wavelets
GBP	Greedy Basis Pursuit
GIF	Graphics Interchangeable Format
HSI	Hue, Saturation, Intensity
HSIM	Hue Similarity Index Metric
HSV	Hue, Saturation, Value
HVS	Human Vision System
IQA	Image Quality Assessment
IQM	Image Quality Metric
IRCT	Irreversible Colour Transform
ITU-R	International Telecommunication Union Radiocommunication Sector
ITU-T	International Telecommunication Union Telecommunication Sector
JNCD	Just Noticeable Colour Difference
JPEG	Joint Photographic Experts Group
KLT	Karhunen-Loève Transform
LIP	List of Insignificant Pixels
LIS	List of Insignificant Sets
LSP	List of Significant Pixels
M-SSIM	Mean Structural Similarity Index Metric
MERGE	Multi-pass Embedded Residual Group Encoding
MMP	Multi-channel Matching Pursuit
MP	Matching Pursuit
MSE	Mean Squared Error
NTSC	National Television System Committee
OMP	Orthogonal Matching Pursuit
OOMP	Optimised Orthogonal Matching Pursuit
PAL	Phase Alternating Line
PCRD	Post Compression Rate Distortion
PLQ	Precision Limit Quantisation
PNG	Portable Network Graphics
PSNR	Peak Signal-to-Noise Ratio
RCT	Reversible Colour Transform
RGB	Red, Green, Blue
RLE	Run Length Encoding
RMSE	Root Mean Squared Error
ROI	Region of Interest
SHSIM	Structure and Hue Similarity Index Metric
SNR	Signal-to-Noise Ratio
SOT	Spatial Orientation Tree
SPECK	Set-Partitioning Embedded BloCK
SPIHT	Set-Partitioning in Hierarchical Trees
SSIM	Structural Similarity Index Metric

Acknowledgements

First of all, I would like to thank Prof. Ian Nabney for his invaluable support and infinite patience. His guidance helped me a lot to improve not only as a researcher but also as a person. I am also very grateful to Dr Yuan Yuan for introducing me into research and inspiring me with an exciting topic of sparse representations of images. Also thanks to Dr Laura Rebollo-Neira and James Bowley for many inspiring and fruitful discussions.

I am very much indebted to Dr Ben Tocher and Dr Laszlo Hetey for a review of the final thesis and plenty of useful comments. Thanks also to James Bowley, my ex-house mate James and Alex for reading some of my writing.

Many thanks to all the people around me during this exciting and sometimes hard time of PhD studies: to my sister and my mother for all the moral support during difficult moments and also to Dr Diar, Dr Alex, Dr Jan, Dr Ben, Maciek, Marcin, Michael and Mitchell for always listening and helping me to maintain my motivation.

1 Introduction

1.1 Motivation

In the era of multimedia, compression is critical for the storage and transmission of video and image data. A wide range of applications includes video broadcasting for digital TV, and image and video distribution over the Internet.

The focus of this thesis is on *digital still images* which are displayed as two-dimensional matrices (*bitmaps*) of values called *pixels*. Typically, pixels are represented as integer values in a range [0..255] for grayscale images thus requiring 8 bits per pixel. In colour imaging, three 8-bit values are used to represent one colour pixel. For display purposes these values usually represent Red, Green and Blue channels. Such a representation is called the *raw format* and the number of bits needed to represent one pixel value is referred to as *bit* or *colour depth*. A static colour picture of size 1600×1200 requires 5.49 MB of memory in a raw format. Two hours of colour video in NTSC standard definition (SD) format (720×480) with 30 raw frames per second would need more than 200 GB. The scale of these numbers indicates that reduction of data is not only desirable but essential for the multimedia industry to function. Therefore videos and images are displayed as bitmaps (frames) but transmitted and stored as compressed *bit-streams*.

A compression system that consists of an *encoder* (compressor) that maps a bitmap into a bit-stream and a *decoder* that performs the reverse operation is referred to as a *codec*. Compression methods can be classified as *lossless*, *near lossless* or *lossy*. Lossless techniques preserve data ideally, near lossless up to rounding errors while lossy introduce

distortion that although mathematically significant can still be acceptable or even unnoticeable for a human observer. The efficiency of image compression method is characterised by the compression ratio defined as:

$$CR = \frac{RawSize}{CompressedSize} : 1 = \frac{c \times W \times H}{CompressedSize} : 1, \tag{1.1}$$

where c is bit depth, W is image width and H is image height. It is more convenient, especially in lossy compression, to use *bit-rate* instead since it represents the average number of bits needed to represent one pixel and is measured in bits per pixel [bpp]:

$$R = \frac{CompressedSize}{W \times H} \text{ [bpp]}. \tag{1.2}$$

Depending on the size of a compressed bit-stream, low, medium and high bit-rates can be distinguished. Low bit-rate image coding refers to bit-rates below 0.5 [bpp], medium to $0.5 - 1$ [bpp], and high to greater than 1 [bpp]. Bit-rates 0.5 and 1.0 correspond to compression ratios: $CR_{0.5} = 16$, $CR_{1.0} = 8$ for grayscale and $CR_{0.5} = 48$, $CR_{1.0} = 24$ for three-channel colour images.

The lower the bit-rate the higher is the compression but more distortion is introduced. For typical still images lossless methods can achieve a bit-rate of 3 bpp while lossy methods can reach less than 0.5 bpp without visible distortion [42]. Recognisable representations can be achieved at rates as low as 0.05 bpp, which is extremely useful when the communication channel capacity is limited or the resolution of the target device is much lower than the size of original image.

For the evaluation of lossy methods an appropriate measure of distortion, denoted here by D, is required. Mean Squared Error (MSE) is commonly used despite being heavily criticised due to its poor correlation with human perception of visual data [123]. The topic of metric selection is touched on throughout this thesis with a special concern over colour data. Since distortion depends on the bit-rate the problem of lossy compression could be viewed as minimisation of the function $R(D)$ called the *rate-distortion curve*.

It has to be remembered that *rate-distortion optimisation* is only one of the issues that need to be taken into account when designing video and image codecs. Modern applications often require a flexible construction of the encoded stream in order to be able to recover partial information without necessarily decoding the whole stream. For example, the fore-mentioned reconstruction at lower resolution or quality may be desired. This is referred to as *scalable* or *progressive* coding and is supported by most of the latest image and video coding standards (e. g. H.264 and JPEG 2000).

Other features desired from modern codecs include (see [65, 114]): low complexity of encoder and decoder, robustness to channel errors, the possibility of performing image operations directly in a bit-stream, region of interest coding. The potential for hardware implementation and parallelisation can be also of interest especially when a sequential algorithm is computationally complex. Moreover, in terms of computational and memory complexity codecs can be classified as either *symmetric* or *asymmetric*. In the symmetric case encoder and decoder have similar complexity while for the asymmetric case they

differ significantly. For example, for scalable coding of TV or Internet broadcasting a fast decoding is critical, while encoding can be slower as it will be done only once. On the other side, applications such as video conferencing require both real time decoding and encoding.

High compression and construction of bit-streams that fulfil industrial demands is a challenge which involves applying techniques from applied mathematics and computer science as well as psychophysics. This thesis looks at the mathematical aspects of scalable image coding on the examples of current image compression standards and recent advances in signal processing. This leads to new ideas of encoding signal representations into bit-streams. The practical issues of implementing codecs are also addressed. Lossy compression for asymmetric systems which require fast decoding is considered with emphasis on low bit-rates and static still colour images. However, most of the presented methods can be used as a part of video codecs.

Historically, the development of image and video coding standards was always a process of putting together advances in understanding human vision, signal representation and theory of information and coding. For example, in 1992 static Huffman coding [52] and Discrete Cosine Transform (DCT) were included in the JPEG standard [55]. The DCT transform was examined by researchers in the 1970s as a way of representing an image in the frequency domain in the same spirit as the Discrete Fourier Transform (DFT) [56]. Due to advances in wavelet methods and the theory of coding in the 1980s and 1990s, JPEG 2000 [115] was based on Discrete Wavelet Transform (DWT) and a version of Binary Arithmetic Coding (BAC) called the MQ-coder [114]. At the turn of the 21st century more flexible tools for signal analysis based on redundant transforms started to be in the research focus of signal and image processing.

1.2 Outline

The research reported in this thesis analyses and evaluates the use of signal representation methods called *sparse approximations* in image coding. A special emphasis is placed on colour images. A novel colour image codec is proposed, implemented, described and evaluated. Both the algorithms used to obtain a sparse approximation of the signal and the proposed idea of encoding this approximation into a bit-stream can be viewed as a form of generalisation of well-known concepts such as *image transform*, *significance map* and *bit-plane coding*. They are the basis of well established methods such as JPEG standards (JPEG and JPEG 2000) and SPIHT.

Chapter 2 introduces the main concepts in image compression and coding. The transform coding is introduced using JPEG standards as examples. The problem of selecting an appropriate transform for image representation is highlighted in relation to the construction of the human vision system. The reasons behind choosing particular wavelets to decompose still images are given. Then a comprehensive outline of wavelet-based coding methods is provided as the state-of-the art in scalable image coding. Bit-plane coding of significance maps is introduced together with an outline of these methods. After presenting

the main concepts for single-channel data (grayscale) their extension into colour imaging is given. In Chapter 3 the problem of sparse approximation of grayscale and colour images is formulated and an algorithm to solve it, called Matching Pursuit (MP), is selected from a range of methods available in signal processing. The proposed implementation of MP is presented in Chapter 4. The problem of designing dictionaries which define MP as a data transformation is carefully studied and the relation between the structure of a dictionary and complexity of encoder is analysed. Chapter 5 deals with the quantisation and encoding of an image approximation into a scalable bit-stream. An in-depth analysis of the quantisation effect is performed. Then a novel coding method that utilises the concepts from database index coding is proposed. In Chapter 6, different choices of colour spaces and optimisation criteria during the approximation process are compared and the proposed codec is evaluated in comparison to state-of-the-art methods. Chapter 7 concludes the thesis and lists the problems that require further investigation.

1.3 Contributions

This thesis considers common problems with sparse approximations such as efficient implementation and the choice of the dictionary. Extensions of these problems into multi-channel data are studied and a general method of encoding atomic decomposition into a bit-stream is proposed.

The author considers the following main contributions of this work:

- Extending the use of MP in the transform domain into colour images (Chapter 4).
- Proposing and evaluating a novel colour-amplitude quantisation scheme (Chapter 5).
- Proposing a new method of encoding multi-channel atomic decompositions with detailed analysis (Chapter 5).

Other contributions include:

- Comparison of different sparse approximation methods Basis Pursuit (BP), Matching Pursuit (MP) and Orthogonal Matching Pursuit (OMP) when applied to lossy image compression (Chapter 3).
- Detailed analysis and evaluation of the idea of MP in the spatio-frequency domain, proposed in [131], using DWT and DCT (Chapter 4).
- Evaluating the effect of signal partitioning into blocks in the DWT domain (Chapter 4).
- Design of dictionaries for grayscale and colour MP proposing an efficient dictionary that minimises coherence (Chapter 4).
- Implementation of single and multi-channel MP with uBlas (Chapter 4).

- Adapting proofs of convergence from [118] to Quantised MP for single and multi-channel signals (Chapter 5).
- Using a standard paired t-test for statistical comparison of the average performance of two different compression methods on a set of images (Chapters 2, 4 and 6).
- Comparison of the proposed method against the SPIHT and JPEG 2000 standards employing recent advances in objective image quality assessment (Chapter 6).
- Analysis of different norms as atom selection criteria for MP (Chapter 6).
- Evaluating usefulness of different colour spaces for MP-based colour image coding. Comparison of a single-channel MP in YC_bC_r colour space against multi-channel performed directly in RGB (Chapter 6).

1.4 Published Work

R. Maciol and Y. Yuan and I. T. Nabney. Colour image coding with Matching Pursuit in the Spatio-Frequency Domain. In *Proc. of the International Conference on Image Analysis and Processing*, volume I, pages 306-317, 2011.

R. Maciol and Y. Yuan and I. T. Nabney. Grayscale and colour image codec based on Matching Pursuit in the Spatio-Frequency Domain. Technical report, Aston University, availabale at: *http://eprints.aston.ac.uk/15194/*, 2011.

2 Modern Lossy Image Compression

Compression is only possible due to the presence of redundancy in data. Three types of redundancy can be distinguished in relation to images [45, p.526-534]:

Inter-pixel redundancy which refers to similarities between nearby pixels.

Psycho-visual redundancy which refers to human perception of the images.

Coding redundancy which refers to the modelling of the symbol distributions and Shannon's entropy coding.

For colour data there is also **inter-channel redundancy** which refers to correlations and dependencies between colour planes.

Reduction of statistical correlations by exploiting inter-pixel, inter-channel and coding redundancies can be done in a lossless way. In practice, lossless techniques such as PNG, GIF or lossless JPEG 2000 achieve compression ratios only up to 3 for standard test images of size 512×512. The key to achieve greater compression is to discard information that is imperceptible to the Human Vision System (HVS). Lossy methods such as JPEG, JPEG 2000 or SPIHT exploit the properties of the HVS allowing us to achieve compression ratios of up to 20 without noticeable distortion. Moreover, wavelet-based JPEG 2000 and SPIHT can provide recognisable image at ratios as high as 100 [42, ch.5]. This chapter outlines the problems of developing and evaluating image coding methods using examples of codecs such as JPEG, JPEG 2000 or SPIHT. The focus is on the scalable lossy coding of still colour images.

The chapter starts with an outline of the HVS in Section 2.1 and highlighting its features that are useful for lossy compression of visual information. Section 2.2 focuses on human perception of colours outlining models used to define colours and represent digital images. A general framework of the image codecs considered in this work is described in Section 2.3. A methodology for comparing codecs is introduced including a review of distortion metrics used for Image Quality Assessment (IQA). A discussion of the pros and cons of using Mean Squared Error (MSE) is followed by an outline of the state-of-the-art distortion metrics. A detailed description is provided for the methods used in Chapters 4-6 for the evaluation of the proposed coding and decomposition methods. Section 2.4 briefly outlines the data transforms used in lossy compression to exploit inter-pixel and psycho-visual redundancies. Section 2.5 addresses problem of encoding transformed data into bit-streams and the main methods of scalable coding are reviewed. Section 2.6 analyses colour transforms and extensions of coding methods presented in Section 2.5 to colour data. Methods for exploiting inter-channel redundancies for colour image compression alternative to colour transforms are also discussed. Section 2.7 summarises the introduced algorithms and concepts.

2.1 The Human Vision System

Processing of *visual stimuli*, which results in *seeing*, is done by the HVS in a few stages. At the first stage a visual signal is captured by the eye lenses. Then it is acquired inside the eye with the use of the two types of *photoreceptors* to be finally processed by the complex network of brain neurons. There are two types of photoreceptors at the image acquisition stage namely *rods* and *cones* located on the inner surface of an eye called the *retina*. Rods are responsible for vision at low luminance levels e. g. in darkness (*scotopic vision*) while cones deal with daylight colour vision (*photopic vision*). Signals captured by the photoreceptors are processed through the network of brain neurons called *cortical cells*, located in part of the brain called the *visual cortex*. A visual signal acquired by approximately 130 million photoreceptors is transmitted to the brain by only 1 million *ganglion cells* without loss of meaningful information [33, ch.1]. This emphasises that already in the first stage of processing in the HVS a *sparse representation* of the input signal is of interest. More than 30 groups of cortical cells denoted as V1, V2, etc. can be distinguished in the visual cortex. Thanks to the communication happening in the network of those cells people can interpret the content of the viewed scene under a wide range of conditions.

Photoreceptors transmit electrical signals depending on the strength of light that has reached them. In general, they act as filters that do not respond below some threshold of the input signal amplitude and get saturated above the upper threshold. The chemical reaction behind this process varies with a type of photoreceptor and determines its spectral response characteristic. Ganglion cells combine the responses of the groups of photoreceptors, typically by taking positive/negative input of one cone/rod and summing it with negative/positive input of the surrounding photoreceptors. In terms of image processing

the ganglion cell acts as the edge detector. Moreover, after this stage signals are processed as frequency-modulated rather than amplitude-modulated at the stage of acquisition by photoreceptors. In further stages of processing visual signals: inside V1, cortical cells act as filters that respond to: various oriented edges, spatial and temporal frequencies, locations and combinations of the above. Hence the interest in representing images in terms of frequencies, spatial locations and edges.

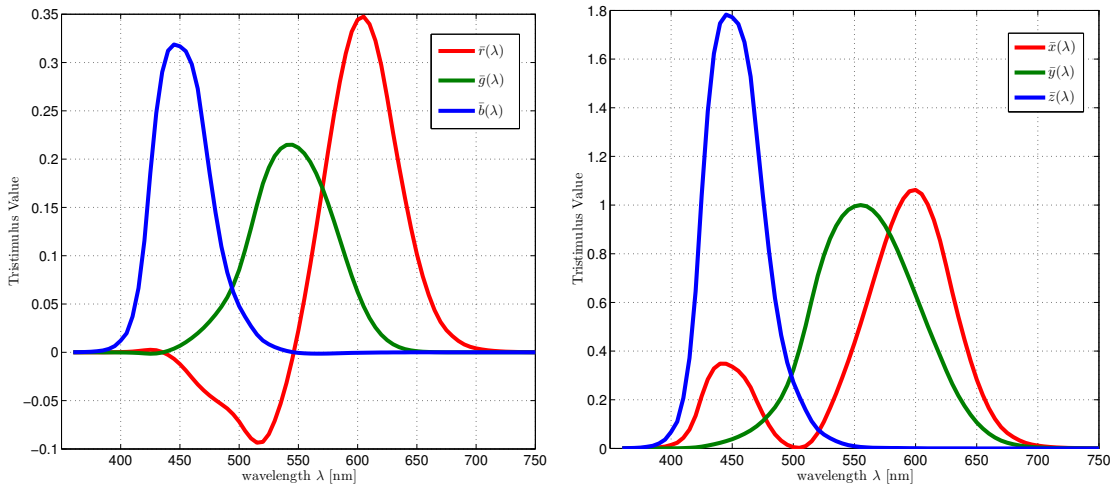
Colour perception is another aspect of vision that can be explained by the physical structure of the HVS. We perceive differently the light at different wavelengths due to a presence of three types of cones of varying spectral responses. L, M, and S cones can be distinguished responding to long, medium and short wavelengths respectively. The area on the retina that has the densest concentration of cones and reduced number of rods is called *fovea*. It occupies an area of 2° angle on the surface of the retina and corresponds to the sharpest spatial and colour vision. The term 2° *CIE standard colorimetric observer* refer to testing human perception of the visual stimuli acquired by the photoreceptors on the fovea. In the next section we look more closely at the nature of colour, human perception of colours and its relation to digital imaging.

2.2 Colour Models

Defining and managing colours is of great importance in all modern multimedia applications. However, the perception of colour is extremely subjective and personal. Colour emerges by interaction of the three components: light source(s), observed object(s) and the HVS [33]. A visual stimuli is an effect of interaction between light and the observed object and can be physically characterised by a function $\Phi(\lambda)$, called *spectral distribution*. Defining colours means building a model that predicts the average human perception of a given stimuli.

2.2.1 Defining Colours: CIE Standards

Any colour perceivable by a human can be matched by a combination of the three primaries of different intensities: this is referred to as trichromatic theory of colour vision. The original experiments, performed independently by Wright (1929) and Guild (1931) on a set of human observers, have matched a perception of the light at every wavelength to a combinations of three primary wavelengths that correspond to Red (700.0nm), Green (546.1nm) and Blue (435.6nm) light. The table of these values for wavelengths from the visible spectrum: $\lambda_l = 380$nm to $\lambda_h = 700$nm defines the CIE RGB colour space for the 2° CIE standard colorimetric observer. Figure 2.1a shows the colour matching functions: $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ and $\bar{b}(\lambda)$. The coordinates (tri-stimulus values) R, G and B of the colour of a



(a) CIE RGB (b) CIE XYZ

Figure 2.1: Colour matching functions.

visual stimuli characterised by a spectral distribution $\Phi(\lambda)$ are:

$$\begin{aligned} R &= \int_{\lambda_l}^{\lambda_h} \Phi(\lambda)\bar{r}(\lambda)\,\mathrm{d}\lambda, \\ G &= \int_{\lambda_l}^{\lambda_h} \Phi(\lambda)\bar{g}(\lambda)\,\mathrm{d}\lambda, \\ B &= \int_{\lambda_l}^{\lambda_h} \Phi(\lambda)\bar{b}(\lambda)\,\mathrm{d}\lambda. \end{aligned} \tag{2.1}$$

CIE RGB can be transformed by linear transformation from Equation (2.2) into another space: CIE XYZ. Hereby negative values are avoided and the Y coordinate measures the brightness of the stimuli. This results in a new set of colour matching functions $x(\bar{\lambda})$, $y(\bar{\lambda})$, $z(\bar{\lambda})$ plotted in Figure 2.1b. The Y-component in CIE XYZ space is called *luminance* and corresponds to spectral response of the HVS that is a balanced combination of L,M and S cone responses. Numerically, most of the luminance information comes from the green component in the CIE RGB. Indeed, cones responses are stronger in the medium-wave (green) region of the visible spectrum and the HVS is the most sensitive to green light.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \frac{1}{0.17697} \begin{pmatrix} 0.49000 & 0.31000 & 0.20000 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00000 & 0.01000 & 0.99000 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \tag{2.2}$$

2.2.2 Device Dependent Colour Spaces

CIE XYZ or CIE RGB define colours referring to the average human perception of visual stimuli forming *device-independent* colour spaces. Devices such as cameras, scanners, printers and digital displays represent image data as a composition of three values (e. g. R, G and B for display or C, M, Y and K for printing) that are translated to analogue signals. R, G and B values on different devices correspond to the different visual stimuli

that also dependent on lighting condition. Digital images are represented using coordinates in *device-dependent* colour spaces what creates a problem with exchanging image data between devices.

For applications including computer display it is usually acceptable to use RGB data calibrated for a typical monitor. Hence, Microsoft and HP have developed the sRGB colour space [112] to simplify the process of exchanging colour data. The viewing conditions correspond to a typical CRT monitor and dim viewing environment simulating a typical office.[1]

Under such conditions the RGB values used by sRGB are related to absolute CIE XYZ values by the following linear transform:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 3.2410 & -1.5374 & -0.4986 \\ -09692 & 1.8760 & 0.0416 \\ 0.0526 & -0.2040 & 1.0570 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{2.3}$$

RGB values outside a range $[0, 1]$ are simply clipped, which results in narrowing of the range of colours (gamut) that can be represented by RGB compared to CIE XYZ. Nevertheless, for most of the applications RGB gamut is wide enough.

Then the values $C \in \{R, G, B\}$ are transformed into the values $C' \in \{R', G', B'\}$ using non-linear transform expressed by Equation (2.4). This transform is called *gamma correction* and is introduced to compensate for non-linearities of the display devices. It has to be noted that at the same time it compensates for non-uniformities of human perception depending on the brightness. In this sense gamma-corrected RGB values form a more-less perceptually uniform colour space. This is, for example, important in measuring colour image distortion (see Section 2.3.2).

$$C' = \begin{cases} 12.92 \times C & \text{if } C < 0.00304 \\ 1.055 \times C^{1.0/2.4} - 0.055 & \text{if } C \geq 0.00304. \end{cases} \tag{2.4}$$

To obtain 8-bit values, normalisation to the range $[0, 255]$ and rounding to the nearest integer is performed. The sRGB form a link between absolute colour definitions using CIE XYZ and numerical data stored digitally. The colour images used as raw data throughout this thesis are gamma corrected sRGB images which will be referred to just as *RGB images*.

2.2.3 Luma-chroma Colour Representation

The RGB colour model, models (to some extent) the image acquisition process performed by the HVS and is convenient and common for display devices. However, according to the so called *opponent colour theory*, signals transmitted to later stages of the HVS include combinations of L, M and S cone responses. Those signals can be defined [33, p.18] as:

(1) L+M+S forming luminance (see Section 2.2.2),

(2) L-M+S forming red-green opponent signal,

[1] In CIE standards the light source is defined by a so called illuminate, D_{65} here.

(3) L+M-S forming yellow-blue opponent signal.

The opponent signals are chrominance components. The HVS is sensitive to a wider spectrum of luminance than chrominance frequencies. Moreover, separating the brightness and colour information is a form of a decorrelating transform and thus reduces difficulties with handling noise. Therefore, for image representation, compression and transmission colour models that separate chrominance (colour) and luminance information are of interest.

YUV

An example of such a model is YUV, that has been originated in analogue TV broadcasting standards, NTSC in the United States and PAL in Europe. Video data stored as RGB are transformed using a linear transformation to the space that can provide more efficient transmission as an analogue signal. Transformation from RGB to the YUV system is given by Equation (2.5):

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ U &= 0.492(B - Y) \\ V &= 0.877(R - Y). \end{aligned} \tag{2.5}$$

The coefficients 0.299, 0.587 and 0.114 determine the amounts of Red, Green and Blue needed to produce reference white light. It has to be noted that the Y channel in YUV and similar colour spaces is not the same as the luminance component Y in CIE XYZ when the transform is applied to RGB values from semi-uniform colour space such as sRGB. Therefore we shall refer to it as *luma* or Y-channel [93, ch.8].

HSV and HSI

Other colour models close to human perception and very popular in image processing nowadays are HSV and HSI [110]. Colour data are represented by its Hue, Saturation and Intensity which is analogous to the way how artists produce colours. At first they select the Hue and then modify its Saturation and Brightness (Intensity) to achieve the desired effect. Due to the properties of the transformation algorithm from RGB: HSV is referred as the *hex-cone model* while HSI as the class of *triangle models*. The hex-cone model transforms the RGB cube through a non-linear transform into a hex-cone while the triangle model into a pyramid. The I value in the general triangle model (HSI) is expressed as a weighted sum of R, G and B:

$$I = w_r R + w_g G + w_b B \tag{2.6}$$

Different weights w_r,w_g,w_b define a different transform hence we talk about the class of triangle models. If $w_r = w_g = w_b = \frac{1}{3}$ then the achromatic (gray) points are placed in the centre of an equilateral triangle. I value in this case can be seen as the projection of R, G, B data onto the diagonal of RGB cube. Taking w_r,w_g,w_b as in Equation (2.5) will result in $I = Y$ from the NTSC standard.

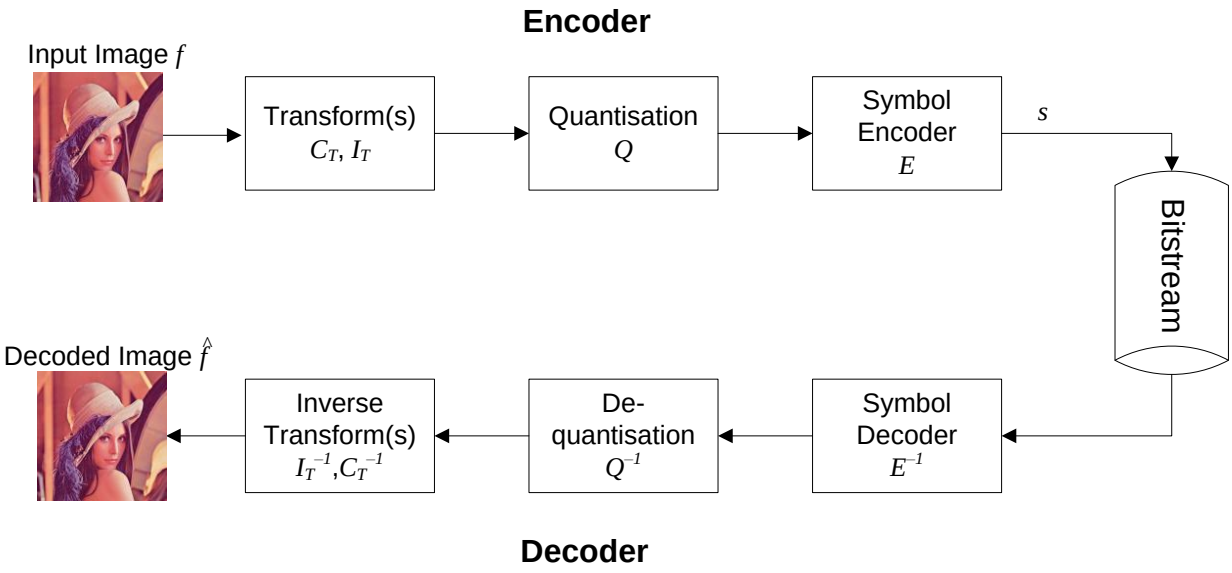


Figure 2.2: Lossy image coding system.

Both, YUV and HSI/HSV exhibit similar decorrelating properties. In this work we prefer chroma-luma transforms for compression as it is used by the standards. The HSV model is utilised in Chapter 6 for image quality evaluation (see also Section 2.3.3).

2.3 Design and Evaluation of Image Codecs

2.3.1 Codec Structure

Image codecs, especially lossy, are designed as hybrid systems with a few steps of processing which exploit a different type of redundancy. Usually three main stages can be distinguished: *transform*, *quantisation* and *entropy coding*. This structure, shown in Figure 2.2, is the basis for most modern lossy image codecs.

It can be observed for still images that pixels close to each other typically do not differ much, although, they do not tend to have equal values. This suggests the presence of statistical correlations and dependencies. The same is true of colour channels in the RGB colour space. Inter-pixel and inter-channel redundancies can be reduced by applying the appropriate decorrelating transform. In addition the transform is designed to be capable of reducing psycho-visual redundancies by exploiting properties of the HVS. Coding standards use two types of decorrelating transforms: colour space and image transforms. Colour space transformations, such as those introduced in the previous section, operate on each pixel independently transforming R, G, B:

$$C_T : (R, G, B) \rightarrow (x, y, z),$$

while image transforms operate on each of the colour planes of size $H \times W$:

$$I_T : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{H \times W}.$$

The transform itself does not give any compression, but a more convenient representation of the image. At the lossy step of *quantisation*, denoted as Q, the floating point values are converted into symbols from a finite set. After quantisation the transformed data can be seen as the symbols from a finite alphabet usually with many repetitions. Such data can be further processed by the symbol (entropy) encoder denoted by E to exploit the coding redundancies. Encoding of an image f is a composition of multiple stages that results in an output stream s:

$$s = E\, Q\, I_T\, C_T(f). \tag{2.7}$$

A decoding process is a composition of reverse operations (denoted as $*^{-1}$) and can be expressed as follows:

$$\hat{f} = C_T^{-1} I_T^{-1} Q^{-1} E^{-1}(s). \tag{2.8}$$

In principle C_T and I_T are reversible transforms i. e: $C_T^{-1} I_T^{-1} I_T C_T(f) = f$ (in practice up to floating point errors). Encoding E, as operating on the integers, is absolutely reversible. Quantisation Q introduces an error ϵ_n for each transformed coefficient c_n:

$$\epsilon_n = |Q^{-1}Q(c_n) - c_n|. \tag{2.9}$$

While in JPEG standards the stages of compression are well separated we will see that some of the steps can be combined. The codec proposed in this thesis is an example in which the components are not entirely distinct.

2.3.2 Rate-Distortion Evaluation

The compression performance of a codec is measured by comparing an image $\hat{f}$ distorted by the process of encoding and decoding against the original one: f. An Image Quality Metric (IQM) which requires both input and distorted images to be known is called a *full-reference* IQM. The most commonly used metric is the Mean Squared Error (MSE):

$$MSE = \frac{1}{HW} \sum_{x=1}^{H} \sum_{y=1}^{W} \Big(f(x,y) - \hat{f}(x,y) \Big)^2. \tag{2.10}$$

The Peak Signal to Noise Ratio (PSNR), derived from the MSE (Equation (2.11)), provides a convenient representation of the MSE in decibels (dB) on a logarithmic scale:

$$PSNR = 10 \log_{10} \left(\frac{{I_{MAX}}^2}{MSE} \right) \text{ [dB]}. \tag{2.11}$$

I_{MAX} denotes a dynamic range of signal f, which is 255 for 8-bit grayscale images.

Equation (2.10) defines MSE for one channel image. Mathematically, the simplest extension to colour images would be an average over the three channels defined as follows:

$$RGB\text{-}PSNR = 10 \log_{10} \left(\frac{3 \cdot {I_{MAX}}^2}{MSE_r + MSE_g + MSE_b} \right). \tag{2.12}$$

As mentioned in Section 2.2, sensitivity of the HVS is different for Red, Green and Blue channels. Adding weights for each channels is a common technique to better model this

difference. Weighted PSNR (W-PSNR) can be defined as:

$$W\text{-}PSNR = 10\log_{10}\left(\frac{{I_{MAX}}^2}{w_r MSE_r + w_g MSE_g + w_b MSE_b}\right). \tag{2.13}$$

By taking weights to be equal to the squares of the first row of Equation (2.5) we get the PSNR calculated for the Y-channel in luma-chroma colour. It is denoted here as Y-PSNR and widely used in colour image and video quality assessment. The idea behind Y-PSNR is to concentrate on the luminance data to which the HVS is the most sensitive. However, the colour information is almost completely discarded.

Despite their high popularity as a standard way of evaluating image compression methods MSE and PSNR are poorly correlated with the human perception of image distortion. This drawback is even more serious in the case of colour data. Another disadvantage of PSNR is that it does not give an absolute scale comparable for different images. Typically at 20 dB the output image is recognisable but has a poor quality, at 30 dB quality is acceptable while at 40 dB there is no visible difference to the reference image. However, it happens often that for some images 20 dB provides acceptable images while for others there are visually annoying artefacts at 30 dB. Figure 2.3 visualises this situation on two standard test images: *Lenna* and *Baboon*. On the highly textured *Baboon* image distortion, although comparable in terms of MSE, is not as clearly visible as on *Lenna*.

An argument for using PSNR is that it is derived from the Euclidean norm in an inner product space and is simple to calculate. Many optimisation problems are formulated as MSE minimisation (equivalently PSNR maximisation). For example, MSE is used when designing a quantiser as a measure of quantisation error (see Equation (2.9)). Therefore it can serve as a first indicator when comparing the compression performance of different algorithms.

2.3.3 Image Quality Assessment

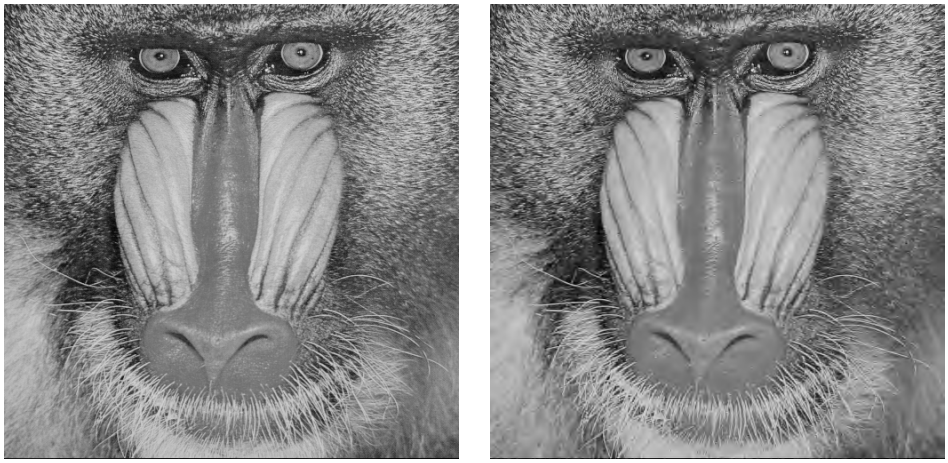
Critiques of MSE with a review of recent promising methods for image quality assessment can be found in [123]. The survey [89] reviews more than 100 metrics. To statistically verify performance of the new metrics, subjective evaluation experiments [92, 104] need to be conducted. Here, we give an outline of full-reference metrics that have been tested and are well established in the field of IQA. We will those metrics for evaluation of compression methods in Chapters 4-6.

One of the most promising techniques is based on combining results of separately comparing structure, luminance and contrast of reference and distorted image. The Structural Similarity Index Metric (SSIM) [122] is defined for a single-channel image patch f_i as:

$$SSIM_i \;=\; l(f_i, \hat{f}_i)^\alpha c(f_i, \hat{f}_i)^\beta s(f_i, \hat{f}_i)^\gamma \tag{2.14}$$

The three components $l(f_i, \hat{f}_i)$, $c(f_i, \hat{f}_i)$, $s(f_i, \hat{f}_i)$ model luminance, contrast and structural distortion respectively. In general, the exponents α, β and γ represent relative importance of each component: they are taken here to be $\alpha = \beta = \gamma = 1$ as in [122]. SSIM is computed locally for N image patches (windows) sliding over the whole image resulting in N values

Original image (left) against compressed with JPEG 2000 at 0.05 bpp (right).
PSNR=27.23 dB, M-SSIM=0.7552.



Original image (left) against compressed with JPEG 2000 at 0.70 bpp (right).
PSNR=27.13 dB, M-SSIM=0.7996.

Figure 2.3: Visualisation of different qualities corresponding to similar PSNR.

SSIM_i for $i = 1, 2, \ldots, N$. If f_i is a window of size $B \times B$ and its distorted version is $\hat{f}_i$ then we can define the mean m_{f_i}, standard deviation σ_{f_i} and cross-correlation $\sigma_{f_i\hat{f}_i}$ within this window as

$$m_{f_i} = \frac{1}{B^2}\sum_{x=1}^{B}\sum_{y=1}^{B} f_i(x,y), \quad \sigma_{f_i} = \frac{1}{B^2}\sum_{x=1}^{B}\sum_{y=1}^{B}\left(f_i(x,y) - m_{f_i}\right)^2,$$

$$\sigma_{f_i\hat{f}_i} = \frac{1}{B^2}\sum_{x=1}^{B}\sum_{y=1}^{B}(f_i(x,y) - m_{f_i})(\hat{f}_i(x,y) - m_{\hat{f}_i}). \tag{2.15}$$

The luminance distortion depends on mean values:

$$l(f_i, \hat{f}_i) = \frac{2m_{f_i}m_{\hat{f}_i} + \theta_l}{m_{f_i}^2 + m_{\hat{f}_i}^2 + \theta_l}, \tag{2.16}$$

the contrast distortion on standard deviations:

$$c(f_i, \hat{f}_i) = \frac{2\sigma_{f_i}\sigma_{\hat{f}_i} + \theta_c}{\sigma_{f_i}^2 + \sigma_{\hat{f}_i}^2 + \theta_c}, \tag{2.17}$$

and the structural information on standard deviations and cross-correlation $\sigma_{f_i\hat{f}_i}$

$$s(f_i, \hat{f}_i) = \frac{2\sigma_{f_i\hat{f}_i} + \theta_s}{\sigma_{f_i}^2 + \sigma_{\hat{f}_i}^2 + \theta_s}. \tag{2.18}$$

The Mean SSIM (M-SSIM) metric is obtained by taking the average over all SSIM_i:

$$M\text{-}SSIM = \frac{1}{N}\sum_{i=1}^{N} SSIM_i \tag{2.19}$$

Small positive constants θ_* in denominators of the Equations (2.16)-(2.18) provide numerical stability when the values σ or m are close to 0. Disadvantage of MSSIM as IQM is that it requires adjusting parameters such as block-size, sliding window overlap and choice of the constants in Equations (2.16)-(2.18). In [124] constants were chosen to be: $\theta_c = \theta_s = (K_1 \cdot L)^2$ and $\theta_l = (K_2 \cdot L)^2$ with $K_1 = 0.01$, $K_2 = 0.03$ and $L = 255$, being dynamic range of 8-bit grayscale images. The block size was 8×8, sliding pixel by pixel. Moreover in practice due to the blocking effect each block was convolved with 11×11 circular-symmetric Gaussian weighting function with standard deviation of 1.5 samples before calculation of $l(f_i, \hat{f}_i)$, $c(f_i, \hat{f}_i)$, $s(f_i, \hat{f}_i)$. In practice this is implemented by adding appropriate weights when calculating means, standard deviations and cross-correlations in Equation (2.15). For more details see [124] and [123]. Within these settings, which we also shall use in this thesis, M-SSIM has been reported to correlate much better with human opinions about image quality than MSE [92, 123, 124].

A natural way to extend Equation (2.14) for colour images is to use the single-channel M-SSIM in a colour space that models human vision such as HSI. This approach was realised in [106] which introduced Structure and Hue Similarity Metrics (SHSIM) based on the M-SSIM from Equation (2.19). For the Hue channel (M-HSIM) a simplification of Equation (2.19) expressed by Equation (2.20) is used:

$$HSIM = \frac{2m_f m_{\hat{f}}}{{m_f}^2 + {m_{\hat{f}}}^2}. \tag{2.20}$$

The SHSIM metric is defined as follows:

$$SHSIM = \frac{\alpha SSIM + \beta HSIM}{\alpha + \beta}. \tag{2.21}$$

The weights α and β were selected to maximise correlation with available subjective evaluation data [104]. The best correlation has been found for $\alpha = 1.0$ and $\beta = 0.2$.

We perform R-D evaluation of compression methods in subsequent chapters using different IQA methods. However, we will still extensively use MSE especially when formulating the optimisation problems. For example, well-known image transformations that will be introduced in Section 2.4 are designed with MSE in mind as the distortion measure. For grayscale images we take into consideration SSIM and PSNR while for colour images we will also analyse performance using Y-PSNR and the methods just introduced based on SSIM, namely SSIM of luminance channel and HSSIM.

2.3.4 Comparing Lossy Compression Methods

To fairly compare the different lossy compression techniques we have chosen 12 standard test images of different sizes and characteristics which are shown in Appendix A. We are interested in a development of a general purpose method so the average performance is of interest. Standard statistical tool to compare average performance for our case (a small sample size) is based on a *paired t-test*, [21, p.322] which analyses a mean and variance of paired differences assuming that they are drawn from the normal distribution. The main idea is that when comparing the two methods A and B for A to be considered better than B it has to be consistently better for most of the data. We can also measure whether tuning some parameters of one method gives statistically significant improvement. Using paired t-test the distribution of differences between methods across a set of images is assessed. In the literature, it is not uncommon that the new ideas for image compression are presented with one image as a target. Using basic statistics we are more likely to avoid tuning a method for a single image.

A comparison procedure to test whether performance of method A is better than B in terms of distortion measure D is as follows:

(1) Measure distortions D_A^i and D_B^i for methods A and B and images $i = 1, 2, \ldots, S$ (in our case $S = 12$).

(2) Calculate differences $\Delta^i = D_A^i - D_B^i$ and estimate mean $\hat{\mu} = \frac{1}{S}\sum_{i=1}^{S} \Delta^i$ and standard deviation: $\hat{\sigma} = \frac{1}{S-1}\sum_{i=1}^{S} (\mu - \Delta^i)^2$.

(3) Check, using Kolmogorov-Smirnoff normality test (K-S test), whether Δ^i are drawn from the normal distribution.

(4) Perform t-test to check whether mean difference is greater than zero what suggests at given confidence level that the method A performs better than B.

We will use t-test to compare different decomposition methods for a fixed number of coefficients (atoms) in Chapter 4 and to compare a proposed algorithm with compression standards at fixed bit-rates in Chapter 6. If we consider fixed rate the values D^i of PSNR are not typically normally distributed for a set of images. In fact this is the case for any IQM considered in Section 2.3.3. The same PSNR may correspond to completely different rates as it was highlighted in Figure 2.3 (p.15). However the differences Δ^i can be considered as approximately normally distributed i. e. we observed that they always pass K-S test. Results of t-test: p-value and a confidence interval for a mean difference can tell us whether the method A can be considered to be superior to B and whether this is of practical importance.

A procedure described above is a case of hypothesis testing. Small p-values (here less than 0.05) indicate rejection of the null hypotheses. We can test three null hypotheses: (1) $\mu = 0$, (2) $\mu > 0$, (3) $\mu < 0$. We shall always report p-values for the hypothesis that $\mu = 0$. If we are interested in testing the null hypotheses that the mean performance of the method A is better than B ($\mu > 0$) provided that the mean estimate $\hat{\mu} > 0$ then we can obtain p-value for this case as: $p_{\mu>0} = 1 - p_{\mu=0}/2$.

2.4 Transform coding

Once a methodology to compare codecs has been introduced the next step is to actually build the codec from blocks described in Section 2.3.1. This section starts with image transforms. We are interested in transforms that:

(1) provide a sparse representation of an image;

(2) decorrelate pixel data;

(3) are general for a wide range of images;

(4) can be implemented using fast algorithms.

Mathematically, the idea behind transform coding is to represent element of linear space in the basis in which vector coordinates are decorrelated. The statistically optimal, in the sense of MSE, linear decorrelating transform is the Karhunen-Loève Transform (KLT) [95, ch.2]. Computation of KLT requires estimation of the covariance matrix and hence it is signal-dependent. Two types of signal-independent decorrelating transforms which attempt to approximate KLT for image data, namely: Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are reviewed in the next sections.

2.4.1 Discrete Cosine Transform

The family of sinusoidal unitary transforms that are data independent and can be computed efficiently has been constructed and described in [56]. It is shown in [56] that the DCT of type II (DCT-II) defined by Equation (2.22) is a good general estimate of KLT for a Markov source of first order with very high correlation. Such a model reflects the high correlations between adjacent pixels [95, ch.2].

$$a_k = C_k \sum_{i=0}^{W-1} f_i \cos\left(\frac{(2k+1)\pi i}{2W}\right), \quad k = 0, 1, \ldots, W-1, \tag{2.22}$$

where $f = [f_0, f_1, ..., f_{W-1}]$ is the input signal and $a = [a_0, a_1, \ldots, a_{W-1}]$ is the transformed output. The normalisation factors C_k equal:

$$C_k = \begin{cases} \sqrt{\frac{1}{W}} & k = 0, \\ \sqrt{\frac{2}{W}} & k = 1, 2, \ldots, W-1. \end{cases} \tag{2.23}$$

The 2D version used for images is given by Equation (2.24).

$$a_{kn} = C_k C_n \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} f_{xy} \cos\left(\frac{(2k+1)\pi x}{2H}\right) \cos\left(\frac{(2n+1)\pi y}{2W}\right). \tag{2.24}$$

Now the input image f is an $H \times W$ matrix: $f = [f_{xy}]$ with $x = 0, 1, \ldots, H-1$ and $y = 0, 1, \ldots, W-1$. The output is also an $H \times W$ matrix: $a = [a_{kn}]$ with $k = 0, 1, \ldots, H-1$ and $n = 0, 1, ..., W-1$ and C_k are given by Equation (2.23). The properties of DCT-II make it especially interesting for image compression tasks. Therefore the lossy part of the

first still-image compression standard: JPEG [55] is based on the DCT transform defined by Equation (2.24) performed on non-overlapping square blocks of fixed size. A default block-size was selected to be $W = H = 8$. The DCT is the real part of the complex Discrete Fourier Transform (DFT) [56] and can be seen as an image representation in the frequency domain. There are two reasons for performing block-wise DCT. Firstly, small blocks are faster to process and require less memory. Secondly, although the lower frequencies contain most of the important data, irregularities, like edges, are concentrated in high frequencies. In order to preserve some information about location of edges a spatio-frequency representation is required. In case of DCT, performing the transform locally, i. e. on small blocks, gives a desired spatio-frequency representation (see Figure 2.4a, p.21). We consider only a discrete number of frequency *bands* in this way. The most top-left corner represents the lowest frequency data that are formed by the most top-left coefficients of DCT for each block. The horizontal frequencies increase from left to right while vertical from top to bottom. One of the problems with this approach is that the use of block transforms introduces visible blocking artefacts. For very high compression ratios blocking is so severe that DCT-based methods such as JPEG are unsuitable for low bit-rate image coding.

2.4.2 Discrete Wavelet Transform

Another way to obtain image representations that are well localised in both space and frequency is to use the Discrete Wavelet Transform (DWT). DWT, contrarily to DCT, is capable of localising the high frequency information [5]. Moreover, it does not operate on blocks thus avoiding blocking artefacts at low bit-rates. The spatio-frequency representations of an image by both DCT and DWT are visualised in Figure 2.4. The highest the frequency the more precisely it is localised.

The DWT is applied as a sequence of low-pass and high-pass filtering operations alternating in vertical and horizontal directions. The result is a set of 2D subband signals as shown in Figure 2.5. Subband 1, often called the approximation subband, represents the lowest frequency information. Higher frequency subbands can be grouped into the *scales* of decomposition. In Figure 2.5 the 2 scales can be distinguished: one for the highest frequency subbands 5-7 and the second one for subbands 2-4.

In the spatial domain the DWT can be defined as a discrete convolution with digital filters:

$$\begin{aligned} a_{2n} &= \sum_{i=-L_L}^{L_L} l_i f_{2n-i} \\ a_{2n+1} &= \sum_{i=-L_H}^{L_H} h_i f_{2n+1-i} \end{aligned} \tag{2.25}$$

where f_n for $n = 0, 1, ..., N-1$ is an input signal of length N, a_{2n+1} stores the high-pass filtered subband and a_{2n} the low-pass, h_i are high-pass filter coefficients and l_i are low-pass filter coefficients. The low-pass filter in Equation (2.25) is of length $2L_L + 1$ and the high-pass of length $2L_H + 1$. The forward transform is done using analysis filters while the reverse is by synthesis filters. Note that Equation (2.25) is a generalisation

i	0	±1	±2	±3	±4
h_i	0.788485	−0.418092	−0.040689	0.064538	0
l_i	0.852698	0.377402	−0.110624	−0.023849	0.037828

Table 2.1: CDF 9/7 analysis filters coefficients.

of Equation (2.22) that defines DCT. For DCT the convolution was done with discrete cosines with the same number of non-zero elements (*support*) as input signal. In case of DWT short-support filters that define *compactly supported wavelets* are of interest. Short filters are more likely to localise the feature they are designed to detect and require less computations. An important property especially for singularities detection and image compression is wavelet *regularity*: n-regular wavelets are orthogonal to polynomials of order up to $n-1$. It means that smooth structures that can be well approximated by polynomials will result in negligible values in high bands in the transform domain.

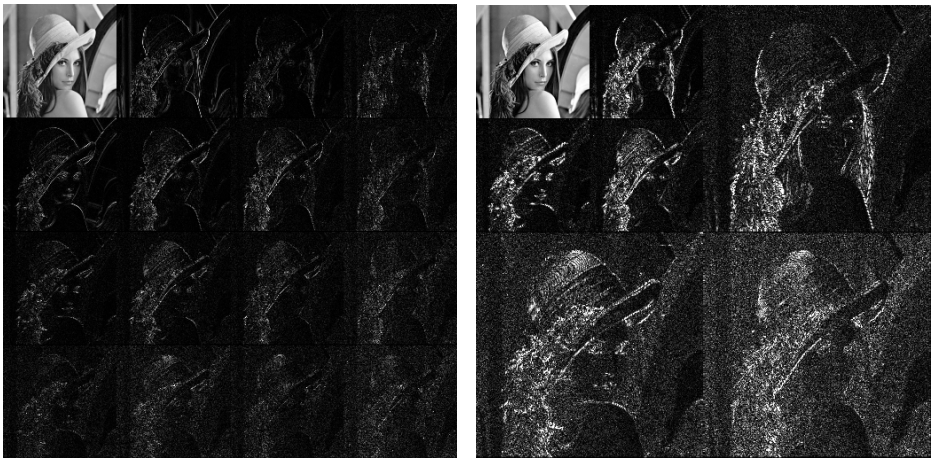
The choice of wavelet filters for image compression has been a subject of intensive research [5, 48, 49]. CDF (Cohen-Daubechies-Feauveau) 9/7 filters [5] have been experimentally shown to give the best R-D performance among short-support filters [48, 49]. Moreover they are highly regular ($n = 4$) and hence became adopted by the JPEG 2000 standard [115]. Coefficients for CDF analysis filters are irrational numbers with approximate numerical values given in Table 2.1 [5]. The synthesis filters coefficients $\widetilde{h_i}$, $\widetilde{l_i}$ used to reconstruct an image are related to h_i and l_i according to Equation (2.26) for $i = 0, \ldots, \max(L_L, L_H)$:

$$\begin{aligned} \widetilde{h_i} &= (-1)^i l_i, \\ \widetilde{l_i} &= (-1)^i h_i. \end{aligned} \tag{2.26}$$

One of the practical problems during wavelet analysis of images is the treatment of image boundaries. The theoretical framework of wavelet analysis has been derived for infinite signals. In practice a signal f_n is defined for $n = 0, 1, \ldots, N-1$: to get the transformed signal a_n for $n = 0, 1, \ldots N-1$ the samples $f_{-1}, f_{-2}, \ldots, f_{-L}, \ldots, f_{-L_H+1}$ as well as $f_N, f_{N+1}, \ldots, f_{N+L_L}, \ldots, f_{N+L_H-1}$ are needed. In signal processing *zero-padding*, i. e. assuming unknown samples are 0, is common. However, for images the best visual performance is typically achieved by *symmetric periodic extension* rather than zero-padding (see also Section 4.1.1 on p.54). For example, a finite sequence $\{1, 2, 3\}$ becomes $\{\ldots, 3, 2, 1, 2, 3, 2, 1, \ldots\}$ after extension.

2.4.3 Complex Wavelets

The DWT, although highly successful in a range of signal processing applications including image compression, suffers from some fundamental shortcomings [64, 101]. The most serious issues for application in image compression are inevitable aliasing and lack of directionality. Aliasing in the case of images can be observed in the form of ringing artefacts. It is not a result of the DWT itself but it is caused by processing the wavelet coefficients for example by quantisation which is an essential step before encoding coefficients into a bit-



(a) 4×4 block-wise DCT. (b) 2-scale 2D-DWT.

Figure 2.4: Subband decomposition of *Lenna*, scaled and contrast adjusted for display purposes.

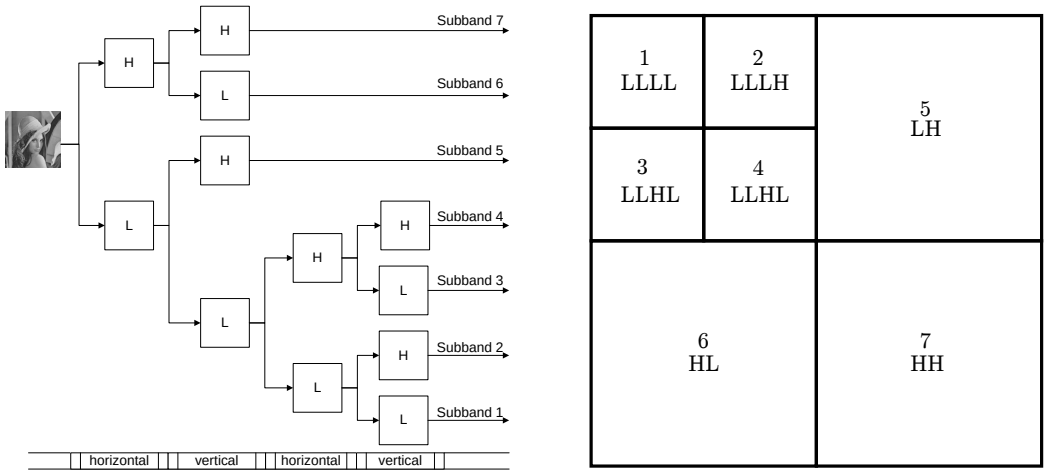


Figure 2.5: Wavelet subband image decomposition.

stream. Further, the patterns represented by DWT that arise from applying consecutive vertical and horizontal filtering have the form of the checker-board oriented simultaneously along different directions [101].

One of the possible solutions is based on performing wavelet decomposition in a similar fashion to the DFT on a complex analytic signal rather than on the original real valued input. In fact the complex Fourier basis in 2D is formed of directed plane waves and quantisation of Fourier coefficients will not introduce aliasing. In principle, by applying the Complex Wavelet Transform ($\mathbb{C}$WT) we expect to keep the advantages of wavelets including regularity and good localisation of singularities while eliminating major deficiencies.

Unfortunately the problem which has occurred now of designing a pair of real and complex wavelets at the same time is not trivial. In a review article [101] the solutions in the discrete case were classified into two categories:

(1) Find wavelets that form orthonormal or bi-orthogonal bases in complex Hilbert space.

(2) Design ℂWT as a redundant transform.

The first set of ideas appears to impose very strong constraints and after applying it we have to face again the same issues as with the real DWT. The second idea overcomes the disadvantages of the DWT on the cost of 2^d times redundancy, where d is the signal dimension. This means 4 times redundant representation in application to images. An example realisation is the Dual-Tree ℂWT from [64] which has already found promising applications in image compression [62, 63, 128]. We shall review this in Chapter 4 when exploring the idea of sparse image representation combining wavelets and non-linear approximations.

2.5 Encoding Transform Coefficients

2.5.1 Quantisation

The floating point output of the transform has to be mapped into symbols from a finite alphabet that can be transmitted to the coder. A *scalar quantiser* maps the numbers $a_n \in \mathbb{R}$ into L disjoint intervals $I_i = [t_i, t_{i+1})$, for $i = 0, 1, \ldots, L$.

JPEG standards use *Uniform Dead-zone Scalar Quantisation* [41] defined by Equation (2.27) (quantisation) and Equation (2.28) (de-quantisation):

$$Q_n = Q(a_n) = \operatorname{sgn}(a_n) \left\lfloor \frac{|a_n|}{\Delta} \right\rfloor, \tag{2.27}$$

$$Q^{-1}(Q_n) = \operatorname{sgn}(Q_n)(|Q_n| + \delta)\Delta. \tag{2.28}$$

The parameter $\delta \in [0, 1)$ defines the position of the de-quantised value inside the interval I_i: if $\delta = 0.5$ then we have quantisation to the *mid-point* of the interval. $\Delta_{(i)}$ is the width of the intervals I_i which in uniform case is the same for all i. Intervals I_i can be identified by the symbols Q_n from an alphabet of size L.

These symbols or their differences can be encoded directly or a *significance map* can be sent to the encoder. A significance map is a binary map that indicates whether to quantise a transform coefficient to 0 or not [102]. Usually the entries in significance maps are specified in terms of *bit-planes*. If coefficients are represented in a binary form: $a_n = \sum_{i=-\infty}^{m} a_n^{(i)} 2^{-i}$, then the values $a_n^{(i)}$ form $(m-i)$th bit-plane assuming all the coefficients are less than 2^{m+1}. By sending coefficients by bit-planes from the most to least significant bit the high coefficients are encoded first. This is a base of the concept of *scalable image coding* outlined in Section 2.5.4. The next section introduces information theoretical concepts used to map quantised coefficients into bits exploiting statistical (coding) redundancies.

2.5.2 Symbol Encoding

The average number of bits needed to encode symbols from a given alphabet is bounded from below by *Shannon's entropy*. For an L-symbol alphabet with probabilities of symbols $\{p_i\}_{i=1,\ldots,L}$ Shannon's entropy is defined as:

$$E = -\sum_{i=1}^{L} p_i \log_2(p_i). \tag{2.29}$$

Estimations of the probabilities p_i define a data model according to which the *entropy coder* maps symbols into bits. No prior knowledge about data i. e. $p_i = \frac{1}{L}$ for all symbols results in a maximal entropy of $\log_2(L)$, which when L is a power of 2 is exactly the number of bits needed to encode each of the symbols.

Imagine we have a sequence of K symbols from an alphabet of size L. To encode it we can map each of the symbols individually to a binary codeword or consider the whole message. In the latter case we would need a model for L^K messages. If data in the sequence are statistically independent then we can process them symbol after symbol and still achieve entropy bounds.

Variable Length Coding

The first idea is to assign a binary codeword for each symbol so that a fewer bits are spent on more likely symbols. The constraint is that decoder has to know where the codeword ends to be able to recover a symbol. This is resolved by using prefix-free codes with the property that none of the codewords is a prefix of any other. Huffman coding algorithm [52] constructs the most efficient of such codes. In fact it only approaches the entropy rate when probabilities are powers of 2 not lower than 2^{-L}. The waste is caused by the requirement that the codewords are at least one bit. Imagine a situation (*binary case*) that we have just two symbols 0 and 1 with p_1 close to 0 which means a lot of repeating 0s. The Huffman code will assign one bit for each symbol although the entropy can be arbitrary small.

Arithmetic Coding

Mapping the whole sequence of symbols into a binary message is an alternative that allows a fractional number of bits per symbol. The idea is usually explained as mapping the message to the real number in the interval $I = [0, 1]$. After each symbol in the sequence is encoded, I is subdivided according to probability distribution of symbols. In the end of encoding the whole message is mapped to some number from I. This idea is called arithmetic coding and can be effectively implemented using finite precision arithmetic. The case of 2-symbol alphabet is referred to as Binary Arithmetic Coding (BAC). Arithmetic coding of the sequence of symbols generated from a given distribution: $\{p_i\}_{i=1,\dots,L}$ asymptotically reaches a theoretical bound of the Shannon's entropy [126].

Run Length Encoding

Usually if data are dependent and correlated we try to exploit it before applying entropy coding. For example if there are long runs of repeating values in the sequence we can signal a symbol followed by the number of its consecutive occurrences which is referred as Run Length Encoding (RLE). In the binary case the run lengths of more probable symbol follow geometric distribution and the optimal Huffman codes to encode them are called Golomb codes [44].

162	162	162	161	162	157	163	161		1284	7	−5	2	−1	1	−2	1
162	162	162	161	162	157	163	161		5	−1	0	1	−1	1	0	0
162	162	162	161	162	157	163	161		3	1	−2	2	−1	−1	2	−2
162	162	162	161	162	157	163	161		0	−5	2	1	−2	0	2	−2
162	162	162	161	162	157	163	161	→	0	2	−1	−1	2	−2	2	−1
164	164	158	156	161	160	159	160		−1	3	−1	−1	1	1	−2	2
161	161	163	158	160	162	159	156		−5	−4	2	0	−1	2	−1	0
159	159	156	157	159	159	156	157		6	3	−2	0	0	1	−1	1
Input data									Rounded transformed data							

Figure 2.6: Example of block-wise DCT transform from JPEG for the most top-left block of luminance channel of *Lenna*.

Entropy coding can be done using static or adaptive models. In static methods the model (distribution) is fixed for all symbols in the sequence while in adaptive it changes by adapting to the changing distribution by exploiting any additional information that arrives with a new symbol.

The next sections describes how these symbol coders are used in practice to encode DCT and DWT coefficients.

2.5.3 Encoding DCT Coefficients: JPEG

For the JPEG, the output after transform is formed from 64 coefficients per each image block of size 8×8. The lowest frequency coefficient a_{00}, called DC (from Direct Current) represents simply the scaled mean value of the block. The remaining 63 coefficients, are called AC (from Alternating Current) components. Figure 2.6 shows rounded output of block-wise DCT: the most top-left value is DC component. Uniform scalar quantisation [41, ch.5] using quantisation tables [55] is performed for each coefficient a_{kn} according to Equation (2.30):

$$Q(a_{kn}) = \text{round}\left(\frac{a_{kn}}{q_{kn}}\right), \tag{2.30}$$

q is a fixed quantisation table that is adjusted depending on the required quality level. Quantised coefficients $Q(a)$ from each block are then encoded using combination of RLE and entropy coding. Figure 2.6 shows the rounded output of block-wise DCT: the most top-left value is the DC component. Data are scanned in zig-zag order from the lowest to highest frequency, for our example: 1284,7,5,3,-1,-5,2,0,1,0 etc., quantised and sent block by block. Either static Huffman coding specified by a standard [55] or arithmetic coding can be used [121, Sec.4]. Different Huffman codes, specified by the standard, are used for DC and AC values.

2.5.4 Scalable Image Coding

The JPEG standard was highly successful at the time it was released in 1992 [55]. However, with quickly evolving multimedia technologies new requirements for image and video

coding standards have been raised. For example, for JPEG 2000, in addition to improved R-D performance at low bit rates, the requirements included: precise rate control and generation of a scalable bit stream [17].

A scalable codec can encode many versions of the image at different qualities or/and resolutions into the same bit-stream. With JPEG limited support for scalable coding is provided. It is referred to as modes of progression and requires prior specification of the type of progression and desired quality parameters. Moreover JPEG does not provide direct rate control (i. e. control of the output file size). The JPEG 2000 standard and the other wavelet coders provide, on top of direct rate control two types of scalability: SNR and spatial. SNR scalability allows an image to be decoded at different qualities while spatial scalability allows decoding at different resolutions. Both scalability features are achieved using special markers (i. e. symbols) added to a bit-stream. Change of progression type can be achieved without re-encoding thanks to the structure of a stream which is formed from packets and markers that indicate all the necessary information [18].

The feature of a bit-stream related to progressiveness and scalability is the generation of the embedded bit stream. Embedded bit-stream refers to fully scalable data. Knowing the first N bits of the compressed stream allows decoding at any target number of bits K up to N bits with the best possible quality (lowest distortion) by taking the first K bits of the known stream. This feature is characteristic for EZW and SPIHT algorithms. Also MP decomposition introduced in the next chapter naturally leads to an embedded bit-stream generation. Data transmission can be stopped at any point resulting in the maximal quality in terms of MSE for the transmitted amount of data.

EBCOT [114], the encoding algorithm from JPEG 2000, does not provide a fully embedded bit stream. However, with the use of markers both spatial and quality progressiveness are provided while EZW and SPIHT can only offer SNR scalability. EBCOT also does not exploit all the redundancies between subbands. However, the generated bit-stream is robust to transmission errors thanks to coding done on separate units called code-blocks which can be important for mobile and Internet applications. This is one of the reasons why EBCOT has been selected as a standard rather than the less complex SPIHT.

JPEG 2000 was a big step forward in image coding. Key factors for the improved performance were the use of wavelets and coding the coefficients by bit-planes. Moreover the nature of DWT and bit-plane coding naturally results in a scalable bit-stream [65]. A few ways of encoding DWT transformed data are reviewed in the next sections.

2.5.5 Encoding DWT Coefficients

EBCOT: JPEG 2000

In baseline JPEG 2000 specified in [115] each wavelet scale is independently quantised using a uniform scalar quantiser with quantisation step adapted to subbands. Further parts of the standard allow more advanced quantisation methods. Quantised wavelet coefficients are rearranged to form an input for the entropy coder. Each subband is split

into non-overlapping rectangles. Three such rectangles from the subbands at the same scale form a *precinct*. Precincts are typically divided into 64×64 *code-blocks* which form a basic unit to be encoded by a symbol coder. Code-blocks are scanned in raster order and by bit-planes of wavelets coefficients starting from the most significant bit. Each bit-plane defines a significance map which is encoded using a binary arithmetic coder [18].

Each bit is encoded in one of three passes, namely: significance propagation, magnitude refinement and clean-up. Different passes use different contexts specified by the standard [115] for arithmetic coder. The bit-stream is refined using Lagrange optimisation to find truncation points for each code-block. This approach is referred to as Post-Compression Rate Distortion (PCRD) optimisation [116].

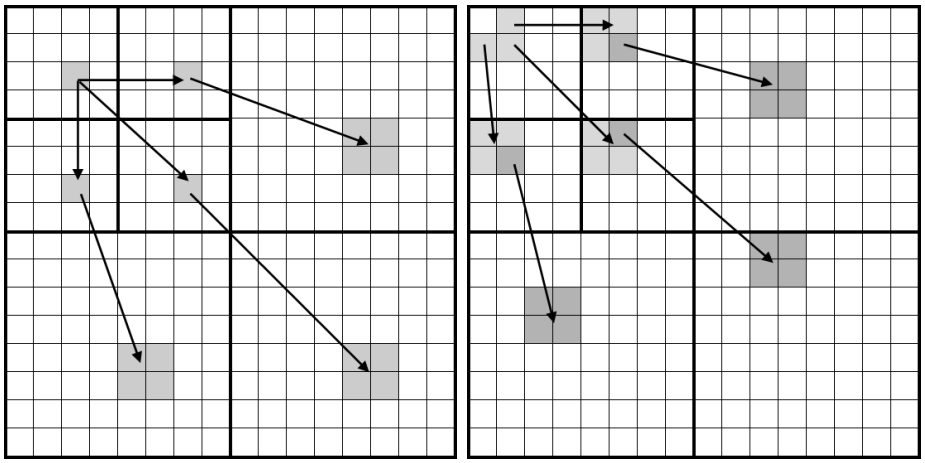
The shortcomings of the coding incorporated into JPEG 2000 are its complexity and that it does not exploit correlations and dependencies between subbands. Each code-block is coded completely independently [18]. Methods that make use of inter-band redundancies between wavelet coefficients, namely EZW, SPIHT and SPECK, are described in the next sections.

EZW

The Embedded Zerotree Wavelet algorithm (EZW) has been developed by Shapiro [102] as an efficient way of encoding significance maps. Shapiro's method utilises similarities between subbands of the same orientation together with the concentration of image information at low frequencies. This is realised with the introduction of the zerotree structure as shown in Figure 2.7. Wavelet coefficients a_{xy} are successively bit-plane coded based on a significance map. Coefficients a_{xy} are scanned in a fixed but arbitrary order with the important constraint of scanning from lower to higher frequency. For each scanned coefficient, the encoder sends one of four types of symbol: the sign of the significant coefficient (+ or -), the zerotree symbol ZT indicating that all descendants of the scanned coefficient are insignificant or the isolated zero symbol IZ which says that there is a significant descendent. Owing to the fact that high coefficient values are unlikely to appear at higher frequencies, a lot of coefficients at later passes are not scanned. This, together with adaptive arithmetic coding of the four symbols +, -, ZT and IZ, makes the effective compression possible.

SPIHT

SPIHT is a coding algorithm proposed in 1996 by Said and Pearlman [99]. Similarly to EZW, it utilises the spatial correlations between wavelet subbands using a tree data structure called the Spatial Orientation Tree (SOT). The algorithm also uses bit-plane coding of wavelet coefficients. However SPIHT traverses through the tree (i.e. scans the coefficients a_{xy}) more efficiently than EZW. This is done by using a slightly different tree structure (see Figure 2.7) and an additional partitioning step that is not present in EZW. Significance bits generated by SPIHT can be arithmetic coded but here, contrarily to EZW, this does not give any significant improvements in coding performance. Nevertheless,



(a) EZW (b) SPIHT

Figure 2.7: Tree structures used in zerotree algorithms.

SPIHT outperforms EZW even without entropy coding. Due to its simplicity and excellent R-D performance SPIHT was among the candidates considered during the standardisation process of JPEG 2000. Although, was not selected, it is still one of the most efficient existing techniques of lossy coding of still images and is often used as a benchmark when evaluating new methods.

SPECK

An extension of the idea of zero-tree coding, called Set-Partitioning Embedded BloCK (SPECK), has been proposed in [88]. The SPECK coder groups the pixels in rectangular blocks exploiting the tree structure of wavelet decomposition. The idea is to find areas in the subbands with high energy rather than single pixels and encode them first. Two lists are maintained during coding: List of Significant Pixels (LSP) and List of Insignificant Sets (LIS), contrarily to SPIHT where an additional list of insignificant pixels (LIP) is also maintained. Sets in LIS are quadrisected to localise significant blocks inside an image in the transform domain.

The R-D performance of SPECK is comparable to SPIHT. It is mentioned here as it exploits dependencies between different subbands and finds significant coefficients by quad-tree partitioning rather than just scanning them in some order like EZW and SPIHT. The sparse approximation methods that will be introduced in Chapter 4, can be used to represent the important areas in an image as atoms rather than single coefficient values.

2.6 Colour Compression

This section presents a range of ideas used to encode RGB images, classified into three categories:

- methods based on decorrelating transforms of colour planes - Section 2.6.1;

Figure 2.8: R, G, B channels of *Goldhill*.
Inter-channel correlations: $\rho_{R,G} = 0.94$, $\rho_{R,B} = 0.90$, $\rho_{G,B} = 0.97$.

Figure 2.9: Y, U, V channels of *Goldhill* (U and V adjusted for display purposes).
Inter-channel correlations: $\rho_{Y,U} = -0.09$, $\rho_{Y,V} = 0.25$, $\rho_{U,V} = 0.34$.

- methods operating directly in RGB colour space - Section 2.6.2;
- extensions to colour data of presented wavelet methods - Section 2.6.3.

2.6.1 Decorrelating Colour Channels

R,G and B channels are known to be highly correlated and dependent [94]. This is visualised on the example of *Goldhill* image in Figure 2.8. Figure 2.9 shows that less correlations between channels exist in luma-chroma space than in RGB.

JPEG 2000 defines two colour transforms: the Reversible Colour Transform (RCT) for lossless coding and the Irreversible Colour Transform (IRCT) for lossy mode. The RCT transform is defined by Equation (2.31) and IRCT by Equation (2.32).

$$\begin{aligned} Y &= \lfloor \tfrac{R+2G+B}{4} \rfloor \\ C_b &= B - G \\ C_r &= R - G \end{aligned} \tag{2.31}$$

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.500 \\ 0.500 & -0.41869 & -0.08131 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.32}$$

A similar transformation to IRCT is used with JPEG to improve the compression ratio although it is not imposed by the standard. Both are referred to as $\mathrm{YC_bC_r}$ colour spaces.

Decorrelation results in concentrating the most important information into one channel. The main idea behind the use of chroma-luma colour spaces for compression is to allow more loss of chroma information. The HVS, as explained in Section 2.2, is more sensitive to high frequencies in luminance. Y component, after transformation to luma-chroma colour

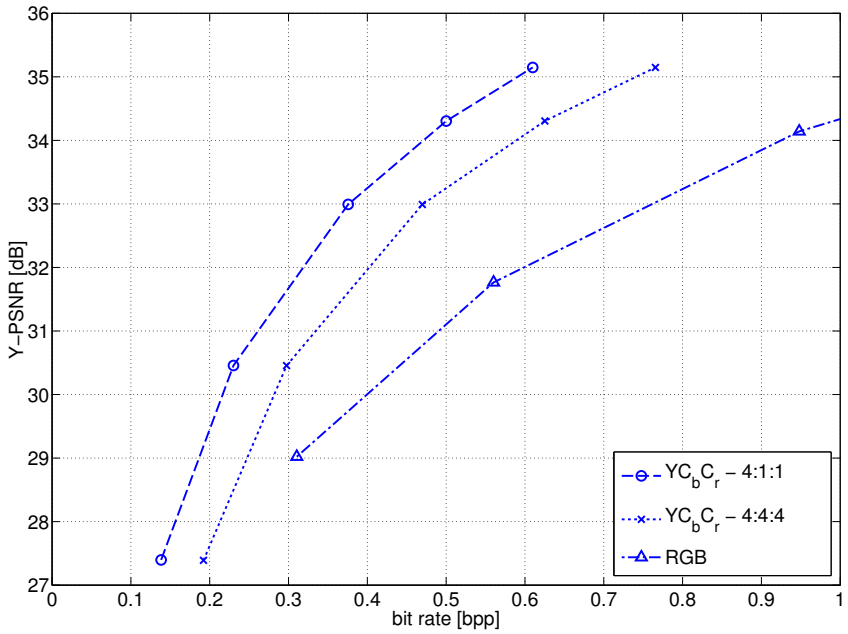


Figure 2.10: Evaluation of decorrelating transform from RGB to YC_bC_r for *Lenna* compressed using JPEG.

space, contains the most of the luminance information. In lossy coding standards C_b and C_r channels are typically sub-sampled. This can be seen as low-pass filtering of chroma (colour) information and gives a significant improvement of compression ratio without visual degradation of an image for most applications. Depending on a technical details of sub-sampling a few modes can be distinguished. The 4:2:0 mode refers to sub-sampling by 2 both horizontally and vertically, 4:2:2 to sub-sampling only in the horizontal direction by a factor 2, 4:1:1 to sub-sampling only in the horizontal direction by a factor 4, 4:4:4 mode does not involve any sub-sampling (see [42, ch.2]). In JPEG 2000 chroma sub-sampling is realised by setting the highest frequency wavelet coefficients to 0 [42]. Following the same property of insensitivity of human vision to high frequency chroma information, a coarser quantisation for C_b and C_r channels is normally implemented in JPEG codecs. It is realised by using a different quantisation table for chroma channels. In this way more chroma high frequency information is discarded by a coarse quantisation.

The difference in compression performance between different sub-sampling methods in YC_bC_r and direct encoding of RGB data is shown in Figure 2.10 for the JPEG codec. Performing coding in YC_bC_r colour space rather than in RGB can give a reduction of bit stream size by up to $\theta = 33\%$ (from about 0.9 bpp to 0.6 bpp) without loss of quality. Further reduction of the size can be achieved by sub-sampling the chroma channels which is effective for low and medium bit rates. When higher quality colour reproduction is needed the mode without sub-sampling should be used to avoid colour errors.

Colour space transformations are simply matrix transforms in three-dimensional linear space. It should be mentioned that transforms that separate the Y-channel with weights from the first row of matrix transform from Equation (2.32), preserves compatibility with the monochrome TV systems [103]. Skipping this constraint gives more flexibility in

design of the optimal colour transformation [50, 61]. With development of new image and video coding standards some research effort has been put into evaluating other linear transformations. Motivations for this include not only the possibility of obtaining higher compression for the same quality but also computational efficiency [91] as well as numerical accuracy for near lossless [61] and integer realisation [90, 91] for lossless compression.

The problem is analogous to searching for the optimal transform in transform coding, reviewed in Section 2.4. Now the number of dimensions is reduced to only three. The optimal decorrelating transform for a given colour image is again KLT. Since derivation of KLT depends on the data, approximations calculated basing on standard image sets are used or alternatively KLT coefficients can be sent with encoded data since the number of dimensions is low.

In [50] a comparative study of 11 commonly used colour transforms has been done. These include transforms from TV standards, JPEG standards and some KLT approximations. For lossy compression the highest performance, by mean of PSNR, for a given bit rate has been achieved for KLT approximations: KLT from [94], Discrete Cosine (DCT) and Discrete Hartley (DHT) Transforms. DCT and DHT are well known as good estimation of KLT for image data. The authors of [50] also presented a derivation of Reversible Transforms for lossless coding.

The new floating point colour transform $\mathrm{YS_bS_r}$ is proposed in [61] (Equation (2.33)) is based on the KLT.

$$\begin{pmatrix} Y \\ S_b \\ S_r \end{pmatrix} = \begin{pmatrix} 0.6460 & 0.6880 & 0.6660 \\ -1.0 & 0.2120 & 0.7880 \\ -0.3220 & 1.0 & -0.6780 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.33}$$

It outperforms $\mathrm{YC_bC_r}$ and other existing transforms by PSNR measured separately for each channel.

A colour transform is the simplest way to extend grayscale compression methods into colour data. However, from results achieved by [50], [61] and [91] it is clear that the performance improvement of the whole coding system just by decorrelating the colour components is limited. The gains in PSNR over $\mathrm{YC_bC_r}$ in [61] are minor and achieved at higher bit rates where it is likely that they will not be perceptible by the human eye for most applications. A review of some of the latest attempts to build efficient colour image codecs that goes beyond colour transformations is given in the next two subsections.

2.6.2 Directly Exploiting Inter-channel Correlations

First of all, instead of decorrelating data with the use of a matrix transformation, the functional dependency between channels can be estimated and only the estimation error encoded. The authors of [43] have proposed a method in which two channels of three-channel image (denote them by C_1, C_2) are approximated by a polynomial of order K of the base colour C_0 (see Equation (2.34)).

$$\begin{aligned} C_1 &= \textstyle\sum_{k=0}^{K} a_k {C_0}^k \\ C_2 &= \textstyle\sum_{k=0}^{K} b_k {C_0}^k. \end{aligned} \tag{2.34}$$

The chosen colour space was RGB. The image was divided into small blocks and linear regression (Least Squares fit of polynomial of order $K = 1$) was used to find coefficients: a_0, a_1, b_0 and b_1 in Equation (2.34). Differences between those coefficients were encoded. As a base colour the one with highest weighted correlation to the others was selected. Information about the selected colour has to be sent for each block. The green channel which contains the most luminance information, is selected most of the time. The Huffman code is used to encode the information about the base colour selection. A further step is taken in [39] where this approach is integrated into subband coders JPEG and JPEG 2000. A similar linear fit is performed in the frequency domain. This allows direct comparison of the proposed method with the decorrelating colour transforms employed in standard codecs. A bit allocation algorithm for colour components and subbands, proposed in [40], has been used. The authors claim to achieve compression improvement in the case of JPEG and visual superiority when integrating the proposed methods within the JPEG 2000 framework.

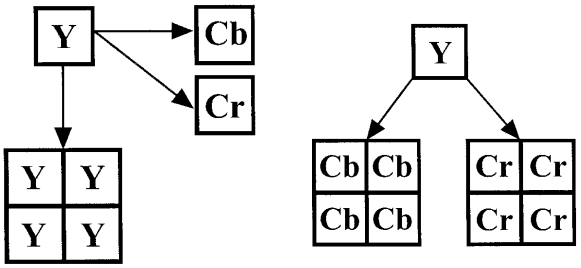
In [80,81] RGB channels are also encoded in the spatio-frequency domain after applying a DCT. Decorrelation of DCT blocks is done using a 3-level wavelet decomposition and optimal bit allocation similar to JPEG 2000. However, the proposed coder is reported to have inferior performance to JPEG 2000.

Another interesting, but in principle completely different, idea for perceptual improvement of image compression algorithms that works regardless of colour space has been proposed in [16] and described in detail in [15]. The idea is to tune existing codecs by allowing them to discard indistinguishable colours. Realisation of the idea has been done by changing the mechanism of rate control in JPEG 2000. The algorithm minimises the colour distortion measure from [14] instead of the MSE (Mean Squared Error). Results for tuned JPEG 2000 suggest that the same visual quality can be preserved with a reduction of size of up to 33%.

A different approach to combining decorrelation transforms with exploitation of remaining redundancies at later stages of processing was presented in [100]. It was analysed in [100] that there are still high inter-colour correlations after RCT or IRCT. The authors have proposed a coding scheme inspired by JPEG 2000 that additionally models correlations between channels in the YC_bC_r colour space. A slight improvement over JPEG 2000 has been reported. The next section reviews the ideas to extend a concept of zero-tree to colour images. In principle the idea is similar to [100]: perform colour decorrelating transform and the DWT and then link resulting data (sub-bands) into zero-tree in order to exploit remaining redundancies.

2.6.3 Colour Extensions of Wavelet-based Methods

Both EZW and SPIHT have been primarily analysed and designed only for grayscale images. EZW has been extended to colours as Colour EZW (CEZW) in [105]. A zerotree that connects colour channels in YUV space has been constructed. The important observation has been made that although YUV channels have little correlation in the wavelet domain,



(a) CEZW [105]. (b) CSPIHT [58].

Figure 2.11: Parent-node relation in colour codecs.

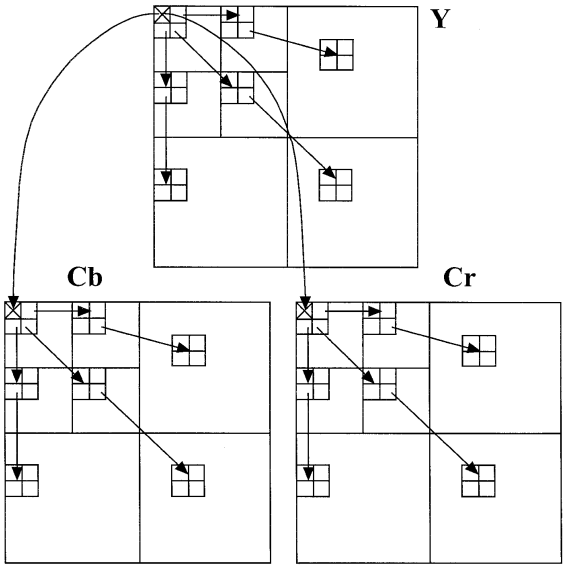


Figure 2.12: Spatial Orientation Tree (SOT) used in CSPIHT [58].

it is likely that sharp transitions (i. e. large values at higher frequencies) occur at the same locations for all channels. The structure of a tree where each luminance coefficient has 6 children as in Figure 2.11a has been proposed and implemented. SPIHT has been extended into colour image and video coding in [60]. An image codec is freely available at [111]. The proposed method encodes all three channels together using separate SOTs. Prior to the coding, a KLT is estimated and applied. The codec preserves the embedded nature of the bit-stream but does not fully exploit inter-channel correlations. As SPIHT performs much better than EZW for grayscale this extension for colours is only just better than [105] when evaluated using PSNR. This suggests usage of the improved tree structure shown in Figure 2.11b also with SPIHT.

The algorithm proposed in [58] builds SOT across all channels in a different way. The coder works in YC_bC_r colour space. The luma and chroma planes are linked through the lowest frequency subband of Y channel that has no offspring (see Figure 2.12). Exactly the same idea has been analysed by Khan and Ghanbari [59] in YUV colour space as *Composite SOT_B*. Moreover they have compared three methods of coding named after the structure of the used Colour SOT (CSOT):

- Independent CSOT, almost the same as [111];

- Composite CSOT_A, tree structure as in [105];
- Composite CSOT_B, tree structure as in [58].

The conclusion has been drawn that CSOT_B is the best. However reconstruction of chroma channels has been better for CSOT_A. The performance of Composite SOTs is always slightly better than for original SPIHT with KLT. Further improvement in the performance has been achieved in [127] by utilising KLT with the tree structure from [58, 59]. Authors have observed that after KLT the magnitudes of the components are different. Therefore wavelet coefficients are grouped by transformed colour components K_1, K_2, K_3 and then by subbands. At each pass only non empty groups are coded.

A further step based on the idea of coding blocks instead of single pixels has led to the family of algorithms represented by SPECK from [88]. The block-tree structure inspired by SPECK has been introduced for colour coding in [74, 75]. Proposed system is reported to have better R-D performance than CSPIHT [58] even though original SPECK performs similar or sometimes worse than SPIHT for grayscale [88]. Additionally block based methods are less memory and computationally complex. Nevertheless, the differences measured by PSNR are small.

2.7 Summary

In this chapter the most important ideas behind transform coding were introduced. These include decorrelating colour and image transforms such as KLT, DCT and DWT, quantisation and bit-plane coding of significance maps. Examples of JPEG standards and zero-trees from EZW and SPIHT were used to illustrate these concepts. The challenges of colour imaging have been pointed out. The basics of colour vision, the most popular colour models and their use in image compression were outlined. The framework of Image Quality Metrics (IQAs) has been introduced for application in systematic comparison of different coding methods.

It can be concluded that, although, invented in the early 1990s, the idea of coding wavelet coefficients by bit-planes and zero-trees adapted in EZW and SPIHT are among the most efficient existing scalable lossy image compression methods. Various ideas for colour compression were also reviewed, including:

(1) improvements of decorrelating transform [50, 61, 94],

(2) discarding indistinguishable colours [14–16],

(3) analysing and exploiting inter-channel redundancies in the spatio-frequency domain [39, 59, 75, 80, 81, 100].

To summarise, most of the presented colour coding algorithms operate in the decorrelated (mainly luma-chroma) colour space: YUV [59, 74, 75], YC_bC_r [58, 127] and KLT [60, 127]. RGB data were used directly in [39] with the intention to exploit correlations at later stages. Moreover, most of the methods operate in the wavelet domain (or DCT [81]) trying

to extend existing methods such as SPIHT or EZW to better exploit inter-correlations and inter-dependencies between colour channels by modifications in the structure of zero-trees.

The next chapters seek beyond wavelets for more efficient transforms and coding methods that preserve scalability and embedded bit stream construction. Moreover, we put special attention on the idea of exploiting inter-channel correlations directly on RGB data without any colour space conversion.

3 Sparse Approximations for Image Compression

Chapter 2 reviewed well-established methods of representing images. This included wavelets which form the basis of JPEG 2000, the industry standard for lossy image coding. Here, alternative and more general ways to represent signals are studied.

Image transforms like the DWT can be defined by sets of filters. Those filters are expected to represent well the features of an image perceived as important by people. It is known that filters oriented at different directions and well localised in space and in different spatial scales (bandpass) characterise simple cortical cells of the mammalian vision system [86]. The wavelet transform outlined in Chapter 2 have been designed with human perception in mind and partially satisfies those properties. Numerous work has been done in wavelet image de-noising [9–11] following Donoho's analysis of general capabilities of wavelets to recover noisy signals [24, 25, 27]. A range of issues, including lack of translation invariance [19], led to the development of signal denoising methods that use *overcomplete transforms* as an alternative, outperforming wavelets [13, 30].

Mathematically the DWT defines the transformation of a basis in a linear space. Sparse representations in redundant (overcomplete) dictionaries, introduced in this chapter, can be viewed as a generalisation of the basis transform. The original problem of the sparse representation has its origins in more general idea of *compressed sensing* of a given signal [26]. The goal is to recover a high-dimensional signal using many fewer components by measuring it (sensing) using a set of measurement vectors from the same space. There exists evidence [86] that human vision processes an image in a similar way. The number of neurons beyond optic nerves increases by an order of magnitude and only a small number

of them is active at the same time for a typical stimulus [6]. Highly sparse firing of neurons is a widespread phenomenon in a variety of brain activities [47].

The above observations suggest that the set of filters used by our vision system is much richer than in sets that define wavelet or Fourier basis. In the fore-mentioned work of Olhausen and Field [86], it has been shown that localised, oriented and bandpass filters can be obtained with a learning algorithm that maximises sparsity. Hence, sparse representations are of interest for a wide range of imaging applications from de-noising [12, 13, 30], de-blurring [31] and restoration (e. g. recovering missing pixels [70]), to image [36, 131] and video compression [20, 83, 130].

This chapter gives a brief overview of the optimisation problems of sparse representations and approximations. The main trends are introduced and selected examples of the algorithms are presented. Section 3.1 introduces the necessary notation and formulates related optimisation tasks. The main algorithms are introduced in Section 3.2 and Section 3.3. We argue in Section 3.4 that although the more sophisticated methods usually give a much sparser signal representations MP remains the method of choice for image and video coding, especially at low bit-rates. Then the idea of Multichannel MP (MMP) is introduced in Section 3.5 and recognised as an effective way to represent colour images and videos. Section 3.6 summarises advantages of MP in image coding application and outlines challenges related to necessity to encode sparse approximations into a bit-stream that are addressed in the next chapters.

3.1 Problem Specification

Consider a signal f as a function from a Hilbert space $\mathcal{H}$ represented as a sum of the basic waveforms g_i called *atoms* that come from an arbitrary set $\mathcal{D}$:

$$f = \sum_{i=1}^{\infty} a_i g_i. \tag{3.1}$$

The set $\mathcal{D} = \{g_i\}_{i \in \Gamma}$ of all the atoms is called a *dictionary* and the coefficients a_i are called the *amplitudes*. The set of all dictionary indices is denoted as Γ and if Equation (3.1) is used then $\Gamma = \mathbb{N}$. In order to be able to represent any function $f \in \mathcal{H}$ as a combination of dictionary elements, $\mathcal{H}$ has to be spanned by $\mathcal{D}$:

$$\overline{Span(\mathcal{D})} = \mathcal{H}. \tag{3.2}$$

This means that the set of all finite linear combinations of the elements from $\mathcal{D}$ is dense in $\mathcal{H}$. Importantly, the dictionary elements are normalised, i.e.:

$$\forall_{i \in \Gamma} \, \|g_i\| = 1. \tag{3.3}$$

The theory of sparse approximations can be formulated in a general Hilbert space (see [118, Sec. 8]). However, in practical signal processing, the data are represented in the form of finite-dimensional vectors. Therefore, it is assumed in this work that $dim(\mathcal{H}) = M$ and the size of the dictionary $\mathcal{D}$ is K, which means that we can use $\Gamma = \{1, 2, \ldots, K\}$.

If $\mathcal{D}$ is a basis in $\mathcal{H}$ (i.e. $K = M$ and g_i are linearly independent) then Equation (3.1) expresses a unique representation of the function f with $a_i = \langle f, g_i \rangle$ for all $i = 1, \ldots, M$. The KLT, DFT, DCT and DWT are basis transforms and can be also viewed as signal representations in a form given by Equation (3.1). In Chapter 2 we looked at basis transforms as matrix transformations or equivalently as sequences of filters. Equation (3.1) can be written as a matrix transform:

$$f = G\mathbf{a}, \tag{3.4}$$

by writing a dictionary in the form of a matrix G with K columns, each representing a dictionary element g_i of length M. The column vector $\mathbf{a}$ is a vector of K amplitudes a_i.

Here, we are interested in the case when $K > M$. Then G is a matrix representation of a *redundant* or *overcomplete* dictionary $\mathcal{D}$. In this case f has many representations over $\mathcal{D}$. For the purpose of compression a decomposition with a minimal number of significant coefficients a_i is desired. The number of non-zero entries in $\mathbf{a}$ is referred to as its L_0-norm and denoted as $\|\mathbf{a}\|_0$. In the case of lossy compression we are satisfied by the approximation $\hat{f}$ of f given by Equation (3.5):

$$\hat{f} = \sum_{i=1}^{N} a_i g_{\gamma_i} \tag{3.5}$$

for a given number of atoms N. In matrix notation Equation (3.5) implies that there are significant entries in the amplitude vector $\mathbf{a}$ only for a subset of indices $\Gamma_N \subset \Gamma$ defined as: $\Gamma_N = \{\gamma_i\}_{i=1,\ldots,N}$. The approximation defined by Equation (3.5) is referred to as *N-term approximation*. For the purpose of analysing greedy algorithms, repeated entries in Γ_N are allowed (i. e. it can happen that $i \neq j$ and $\gamma_i = \gamma_j$).

The optimisation problem that is a subject of the main focus in this work can be specified as follows. Find a subset (vector) of dictionary indexes Γ_N and a vector of amplitudes $\mathbf{a}$ such that the error:

$$e\left(\Gamma_N, \mathbf{a}_N\right) = \left\| f - \hat{f} \right\| \tag{3.6}$$

is minimal under the constraint of a fixed number of atoms:

$$\|\mathbf{a}\|_0 = N. \tag{3.7}$$

We shall use notation $\mathbf{a}_N$ for a vector that satisfies (3.7). We can see that we are interested in the minimum of the function $e\left(\Gamma_N, \mathbf{a}_N\right)$ over all possible N-term approximations. If the error is $e\left(\Gamma_N, \mathbf{a}_N\right) = 0$ then Equation (3.5) gives an exact reconstruction of the signal which is referred to as an *N-term representation* and the signal is said to be *N-sparse*.

It has to be noted here that theoretical analysis (e.g. [29,119,120]) considers either exact representation or approximation for a given error threshold ϵ that has a minimal number of atoms. In general the sparsest representation for a given error threshold ϵ is of interest. Then the problem at hand is to minimise $\|\mathbf{a}\|_0$ (L_0-norm) subject to $e\left(\Gamma_N, \mathbf{a}_N\right) \leq \epsilon$. This set-up with $\|\mathbf{a}\|_0$ as an objective function is very useful for denoising where we are interested in recovering an approximation $\hat{f}$ that is composed of as many components as

needed to obtain a version of the signal f with noise removed. However, in compression we are usually interested in minimising error for a given target size of a bit-stream (bit-rate). Intuitively we can expect that the number of atoms in decomposition is approximately proportional to the number of bits needed to encode this decomposition. Moreover, we want to be able to organise the stream in an embedded way as described in Section 2.5.4 so the most important atoms are encoded first. For now, let us outline the main problems and introduce a range of methods designed to look for sparse approximation. In Section 3.4 we will analyse issues with adapting those methods to solve a problem defined by Equations (3.6) and (3.7).

In general using the number of atoms $\|\mathbf{a}\|_0$ as either a constraint or an objective makes the problem non-convex. Therefore in either of settings sparse approximation and representation are combinatorial and NP-complex tasks [119]. To find a solution guaranteed to be optimal for a fixed number of atoms N, $\binom{K}{N}$ possibilities need to be searched which is typically infeasible. A wide range of algorithms can find a suboptimal solution. They fall into two main categories:

- greedy methods that iteratively build Γ_N and $\mathbf{a}_N$ like Matching Pursuit (MP) and its variations such as Orthogonal Matching Pursuit (OMP) and Optimised Orthogonal Matching Pursuit (OOMP);
- global optimisation methods like Basis Pursuit (BP) that attempts to translate the problem into a convex optimisation.

The second class of methods replaces the non-convex norm $\|\mathbf{a}\|_0$ by related convex function denoted as $\|\mathbf{a}\|_s$ which favours the components of $\mathbf{a}$ with many small values:

$$\|\mathbf{a}\|_s = \sum_{i=1}^{K} S(a_i). \tag{3.8}$$

In fact as pointed out in [29, sec.1.6] any symmetric, non-decreasing function S with non-increasing derivative for $x \geq 0$ favours a sparse solution. In [85] a few choices for $S(x)$ have been tried, including: $|x|$, $\log(1+x^2)$ or $-e^{-x^2}$. $S(x) = |x|$ deserves a special attention as it translates the problem into a linear programming task which is referred to as Basis Pursuit [12].

Both classes of methods are in use with satisfactory results for different signal processing applications. Next sections give an overview of MP (Section 3.2.1) and its enhancements OMP (Section 3.2.2) and OOMP (Section 3.2.3). BP is outlined in Section 3.3.1. We summarise Section 3.3.1 by highlighting a theoretical importance of OMP and BP. We then also consider greedy modification of BP: Greedy Basis Pursuit (GBP) in Section 3.3.2. In Section 3.4.1 we argue basing on experimental results that MP is the most promising choice for image coding at low bit rates.

3.2 Greedy Algorithms

3.2.1 Matching Pursuit

In 1993 Mallat and Zhang introduced a simple greedy technique for solving sparse approximation [72]. The Algorithm 3.1, called Matching Pursuit (MP), iteratively finds the approximation of a signal f by a sum of N atoms g_{γ_n} selected from a dictionary $\mathcal{D}$:

$$f \approx \sum_{n=1}^{N} \langle R^n f, g_{\gamma_n} \rangle g_{\gamma_n}. \tag{3.9}$$

For any dictionary that spans $\mathcal{H}$, the decomposition in Equation (3.9) obtained after N steps of Algorithm 3.1 converges to f as $N \to \infty$ [72]. At each iteration the atom most correlated with the actual signal residual $R^n f$ is selected and removed from $R^n f$. MP

Algorithm 3.1 Single-channel Matching Pursuit [72].

Initialisation: $R^1 f = f$.
for $n = 1$ to N **do**
  Find atom $g_{\gamma_n} \in \mathcal{D}$ such that:
  $|\langle R^n f, g_{\gamma_n} \rangle| \geq \alpha \sup_{g_\gamma \in \mathcal{D}} (|\langle R^n f, g_\gamma \rangle|)$.
  Update residual:
  $R^{n+1} f = R^n f - \langle R^n f, g_{\gamma_n} \rangle g_{\gamma_n}$.
end for

gives a suboptimal N-term solution to the sparse approximation problem.

We recall a few properties of MP. Firstly, the way the atoms are removed from a signal implies the energy conservation, i. e. after n iterations we have:

$$\left\| R^{n+1} f \right\|^2 = \|R^n f\|^2 + |a_n|^2, \tag{3.10}$$

with $a_n = \langle R^n f, g_{\gamma_n} \rangle$. This implies inductively a Parseval-like equality for the N-term MP decomposition:

$$\|f\|^2 = \sum_{n=1}^{N} |a_n|^2 + \left\| R^{N+1} f \right\|^2. \tag{3.11}$$

The equality from Equation (3.11) implies convergence of MP without requiring the selection of the best atom at the nth iteration. This is a sub-optimality parameter $\alpha \in (0, 1]$ from Algorithm 3.1 which gives some flexibility at the selection step. In theory, for an infinite dictionary $\mathcal{D}$, the best atom, i. e. the one that maximises the absolute value of the inner product $|\langle R^n f, g_\gamma \rangle|$, may not even exist. In a finite case the supremum (sup) in Algorithm 3.1 can be replaced with the maximum (max). The sufficient and necessary condition of convergence for any dictionary and any Hilbert space has been formulated in [118] and [67].

One of the theoretical results [118] is that parameter α can vary from iteration to iteration. Convergence is guaranteed to be preserved for sequences $\alpha_k \in (0, 1]$ such that $\sum_{k=1}^{\infty} {\alpha_k}^2 = \infty$, i. e. α_k are large enough so the series diverges. Although theoretically possible, choosing atoms in a suboptimal way significantly slows down the convergence. A marginal case of $\alpha = 1$ (selecting the best available atom) is referred to as the Full

Search MP (FSMP) [20, 35]. In the theory of approximations, the FSMP is referred to as a Pure Greedy Algorithm while MP in the form of Algorithm 3.1 as a Weak Greedy Algorithm [118]. It is known that in image compression FSMP gives significantly lower distortions than atom selection heuristics [20]. Therefore unless explicitly stated we refer further to FSMP simply as MP and shall always perform full search.

It has been proven in [72] that the decay of the error $R^{n+1}f$ in a finite dimensional space is exponential and a bound is given by the following formula:

$$\left\|R^{n+1}f\right\| \leq \|f\| \left(1-\alpha^2\beta^2\right)^{(n+1)/2}, \tag{3.12}$$

where β is the cosine of the maximum possible angle between functions from $\mathcal{H}$ and their closest dictionary elements:

$$\beta = \inf_{f\in\mathcal{H}} \sup_{g_\gamma\in\mathcal{D}} \frac{|\langle f, g_\gamma\rangle|}{\|f\|}. \tag{3.13}$$

This result implies that the first few atoms preserve most of the signal energy. In marginal case if there is an atom that exactly matches the signal: $\beta = 1$ and full search is performed ($\alpha = 1$) then the convergence is achieved in one step. In general the decay of error $\left\|R^{n+1}f\right\|$ is dependent on the structure of the dictionary described by parameter β and on the sub-optimality parameter α.

MP has been originally proposed for speech signal de-noising [72] but also found applications in EEG signal analysis [28], image de-noising [32], video [83] and image compression [36, 131]. MP with an appropriately chosen dictionary can well match the signal structures. Nevertheless it suffers several major shortcomings that include:

(1) High computational cost of finding the atomic decomposition.

(2) Non-guaranteed convergence in finite steps even in the finite-dimensional space (MP converges only asymptotically).

(3) Risk of finding a representation that is far from optimal which is a typical drawback of greedy techniques.

While finding decomposition (encoding) is slow composing an image back (decoding) requires just summing up the atoms which makes decoding as fast as DCT or DWT. Therefore MP is suitable for an asymmetric application in which one encodes the stream once and decodes it many times. This includes the storage of images and videos for progressive transmission.

3.2.2 Orthogonal Matching Pursuit

The convergence of the MP is not guaranteed in a finite number of steps. The atoms selected after N steps indicated by Γ_N are not only not guaranteed to be linearly independent but not even to be different meaning that the same atom can be selected many times. A simple numerical example when MP fails to recover a sparse representation was shown in [87]. This problem is solved by the method called Orthogonal Matching Pursuit (OMP) that guarantees the exact recovery of a signal in finite-dimensional space $\mathcal{H}$ with

no more steps than the dimension of the space [87]. A key change compared to MP is the requirement for the residual $R^{n+1}f$ to be orthogonal to the current n-term approximation.

$$\langle R^{n+1}f, g_{\gamma_i}\rangle = 0 \qquad \text{for } i = 0, 1, \ldots, n \tag{3.14}$$

The Algorithm 3.2 involves an additional step to preserve the property expressed by Equation (3.14). It can be achieved by updating amplitudes of already found atoms, which requires solving a system of linear equations of size $n \times n$ at the nth iteration. In fact, this is equivalent to solving least squares at each iteration. This can be performed recursively based on the solution from previous iteration [87].

The atoms $\{g_{\gamma_i}\}_{i=1,\ldots,n}$ that form the n-term decomposition are guaranteed to be linearly independent. The set of amplitudes $\mathbf{a}_n = \{a_i^n\}_{i=1,\ldots,n}$ minimises the mean squared error $e(\Gamma_n, \mathbf{a}_n)$ from Equation (3.6) at each iteration for an already fixed set of atoms Γ_n. The actual improvement over MP is based on a refinement of the values of amplitudes

Algorithm 3.2 Single-channel Orthogonal Matching Pursuit [87].

Initialisation: $R^1 f = f$, $\mathcal{D}^1 = \emptyset$.
for $n = 1$ to N **do**
  Find atom $g_{\gamma_n} \in \mathcal{D} \setminus \mathcal{D}^n$ such that:
  $|\langle R^n f, g_{\gamma_n}\rangle| \geq \alpha \sup_{g_\gamma \in \mathcal{D}\setminus\mathcal{D}^n} (|\langle R^n f, g_\gamma\rangle|)$.
  Calculate $\{b_i^n\}_{i=1,\ldots,n-1}$ so for all i it satisfies:
  $\langle g_{\gamma_n}, g_{\gamma_i}\rangle = \sum_{j=1}^{n-1} b_j^n \langle g_{\gamma_j}, g_{\gamma_i}\rangle$
  Update amplitudes:
  $a_n^n = \frac{\langle R^n f, g_{\gamma_n}\rangle}{1-\sum_{j=1}^{n-1} b_j^n \langle g_{\gamma_j}, g_{\gamma_n}\rangle}$
  $a_i^n = a_i^n - a_n^n b_i^n$, for $i = 1, \ldots, n-1$
  Update residual:
  $R^{n+1}f = f - \sum_{i=1}^{n} a_i^n g_{\gamma_i}$.
  $\mathcal{D}^{n+1} = \mathcal{D}^n \cup \{g_{\gamma_n}\}$.
end for

a_i^n at the cost of additional computational and memory overhead. A superscript n next to amplitudes indicates that all amplitudes are updated every iteration. The OMP is an improved version of MP that selects the next atom in the same fashion but recalculates amplitudes to reduce the error. An update equation analogous to Equation (3.10) for MP still holds:

$$\left\|R^{n+1}f\right\|^2 = \|R^n f\|^2 + |a_n^n|^2, \tag{3.15}$$

However, since amplitudes keep changing from iteration to iteration the Parseval equality (Equation (3.11)) no longer holds and hence the atom contribution to overall error cannot be explained any more just by the square of its amplitude.

In approximation theory OMP is known as a Weak Orthogonal Greedy Algorithm [118]. In practice, in initial iterations the behaviour of OMP is almost exactly the same as MP as atoms selected at first are often nearly orthogonal. Note that the first iteration of Algorithm 3.2 is exactly the same as for Algorithm 3.1.

3.2.3 Optimised Orthogonal Matching Pursuit

OMP does not guarantee that the update step chooses the atom that improves the approximation the most after the orthogonalisation expressed by Equation (3.14). One of the improvements has been introduced in [98] called Optimised Orthogonal Matching Pursuit (OOMP). OOMP guarantees that the next atom is chosen so that the updated approximation has minimal error. OMP selects the next atom γ_{n+1} and adds it to Γ_n:

$$\Gamma_{n+1} = \Gamma_n \cup \{\gamma_n\}.$$

Then it updates amplitudes independently

$$\mathbf{a}_{n+1} = \mathbf{a}_n \cup \{a_{n+1}^{n+1}\}.$$

OOMP updates both dictionary Γ_n and amplitudes $\mathbf{a}_n$ of n-term decomposition at the same time so that the error for $n+1$-term decomposition is minimised.

Several further refinements that remove or replace some non-optimal atoms have been proposed over the years [3, 4, 97]. As we shall see in the example given in Section 3.4, in practice improvements of OMP and OOMP over MP related to faster convergence only become clear for relatively many atoms. This fact favours MP for applications to low bit-rate image coding which is the main topic of this thesis. The numbers of atoms in decompositions considered in Chapters 4-6 are much lower than the actual signal dimension, for example 6000 atoms is only 2.29% of $512 \times 512 = 262144$ which is a typical size of test images. On the other side, slow convergence of MP prevents from adapting it at higher rates.

3.3 Global Optimisation Techniques

All refinements to MP such as OMP or OOMP remain greedy techniques and still can be trapped into a local minimum. The main concern is that sub-optimal atom choice in the first iterations will require to be recovered at later stages. A simple numerical example (Chen's example from [13]) can be given when OMP fails to recover a N-sparse signal requiring M steps even though N is much smaller than a dimension of space M. Global optimisation is an answer to the shortcomings of the greedy nature of iterative procedures such as MP and OMP, but at the cost of increased computational and memory complexity. Here, we give an example of the approach called Basis Pursuit that has been found especially useful for signal de-noising [13, 30, 31]. For application in scalable image coding, iterative methods such as MP or OMP are, as we shall see in Section 3.4.1, still of interest. It is important to obtain a decomposition as in Equation (3.5) in which the atoms are sorted by their importance which can be measured by the contribution to overall error. Hence, we also consider performing Basis Pursuit in iterative manner referred to as Greedy Basis Pursuit (GBP) [53].

3.3.1 Basis Pursuit

Basis Pursuit refers to the concept of translating a non-convex sparse approximation problem into linear programming [12]. For the exact signal reconstruction given by Equation (3.8) this is done by taking the L_1 norm instead of L_0:

$$\|\mathbf{a}\|_1 = \sum_{n=1}^{K} |a_n|. \tag{3.16}$$

Note that this is the same as taking function $S(x) = |x|$ in Equation (3.8). The optimisation problem solved by Basis Pursuit can be expressed as:

$$\min_{\mathbf{a}}\{\|\mathbf{a}\|_1\}, \tag{3.17}$$

with a constraint $f = G\mathbf{a}$ as has been expressed by Equation (3.4). Non-zero entries in $\mathbf{a}$ indicates selected atoms. The problem can be directly translated into a linear program as shown for example in [12]:

$$\min_{\hat{\mathbf{a}}}\{\mathbf{1^T}\hat{\mathbf{a}}\} \text{ subject to } [G, -G]\hat{\mathbf{a}} = f,\ \hat{a}_i \geq 0, \tag{3.18}$$

the result $\hat{\mathbf{a}}$ includes both positive and negative components of the actual amplitudes $\mathbf{a}$. Subtracting second half of $\hat{\mathbf{a}}$ from the first one gives the result $\mathbf{a}$.

The following modification of original BP formulation called Basis Pursuit De-Noising (BPDN) [13] is considered in application for image de-noising rather than exact reconstruction [30]. Then, the optimisation problem which is equivalent to convex perturbed linear programming [13] can be formulated as follows:

$$\min_{\mathbf{a}}\{\frac{1}{2}\left\|f - \hat{f}\right\| + \lambda \|\mathbf{a}\|_1\}, \tag{3.19}$$

where $\hat{f}$ is given by Equation (3.5). Factor λ in additive white noise model equals to $\sigma\sqrt{2\log K}$, [13] where σ is a parameter of noise and K is a size of dictionary [13]. For many applications BP and BPDN are more powerful than MP and OMP [13, 31]. The main advantage is that, since it has been translated to a convex a linear (BP) or perturbed linear (BPDN) programming optimisation problem, it is guaranteed to converge to a global optimum.

It has to be remembered here that we are finding a global minimum of a different optimisation task (either (3.18) or (3.19)) while the sparsest representation is of interest ((3.6)-(3.7)). The remarkable property of BP is that after changing optimisation objective it can recover N-sparse signals in cases when greedy methods fail. What is more, if N is sufficiently small comparing to a signal dimension M then BP finds the sparsest solution in L_0-norm sense. Numerous research has been devoted to study this phenomenon [26, 29, 119, 120]. In [119] the conditions have been given that guarantees success for both OMP and BP. Those conditions give the lower bound for N and a dictionary for which Pursuit algorithms succeed to solve exact recovery problem (see also [26]). In practice signals are hardly ever exactly sparse as pointed out in [119] and if they are there is a major challenge to identify appropriate dictionary.

3.3.2 Greedy Basis Pursuit

If the signal is not N-sparse in the dictionary used BP could be used to obtain a sparse approximation based on the fact that many coefficients a_n are small and could be neglected. The two major practical problems here include necessity to compute the full decomposition at first and selection the most important atoms afterwards. If we are focused on scalable lossy image compression we want to be able to target a fixed number of atoms so that we can stop computationally expensive decomposition process once we obtain satisfactory image quality. Moreover, we want to order atoms to support construction of embedded bit-stream so that the ones selected first contain most of the information. Greedy methods addresses both issues hence MP and its refinement is the method considered in image compression applications.

There has been some research effort to adapt global optimisation such as BP so it can work in situations where MP suits better than BP. Greedy Basis Pursuit where we can order atoms by their importance has been introduced in [53]. The idea is to exploit geometric properties of linear program as defined by Equation (3.18) in order to be able to iteratively select atoms to build up a signal decomposition. In the example involving speech signal authors of [53] achieved better sparsity than MP but the mean squared of the residual error decrease was slower. In the next section we try to apply GBP among other methods to down-sampled versions of still images obtaining results in similar spirit to [53].

3.4 Sparse Approximations of Images

3.4.1 Choice of Method

In order to use sparse approximations for image compression the crucial factors include the sparsity of decomposition and the possibility to encode it progressively as for wavelets codecs. Iterative greedy techniques are of interest in image and video coding as they naturally generate a progressive representation. Since the atom amplitude tends to decrease (i.e. the atoms are found from more to less important) it is possible to build a scalable image codec based on greedy methods [36, 131]. There is no advantage in global optimisation such as BP when it comes to scalable image coding as it optimises for a fixed error level or number of atoms.

The following experiment summarises properties of outlined methods and attempts to evaluate their application into image coding. Figure 3.1 shows a comparison of the outlined methods applied to represent images using the same dictionaries. In this example we used the lowest frequency subbands of size 16×16 after the wavelet transform as it represents the most of the image energy. Figure 3.1 considers grayscale *Lenna*. Results for all test images can be found in Appendix C. The main motivation for a choice of small-size signals is the computational complexity of BP and OMP. Also it has to be noted that it the lowest frequency subband can be seen as a down-sampled version of the original image. The input signals have been normalised by dividing each sample by the maximum absolute

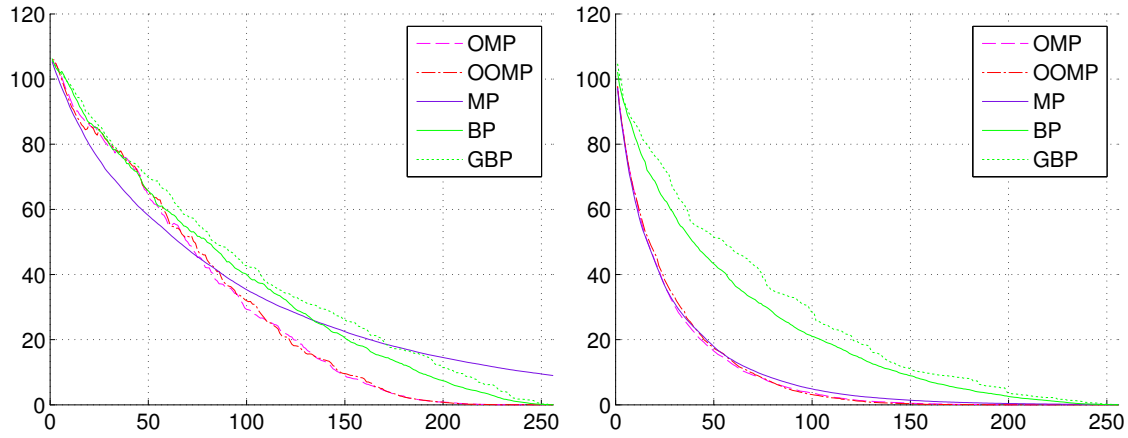


Figure 3.1: Comparison of different basis selection methods for the normalised and DC-shifted lowest frequency subband of 5-scale CDF 9/7 wavelet decomposition of grayscale *Lenna*, 16×16. RMSE (y-axis) as a function of number of atoms (x-axis). Random dictionary with $\times 4$-redundancy (left), $\mathcal{D}_{16}$ (right).

value and mean shifting so that Figure 3.2 shows meaningful PSNR values for image data. We apply MP, OMP, OOMP, BP and GBP in the most general form. The input is a 256-dimensional signal and hence a dictionary is represented by a matrix with 256 rows. Implementations of OMP and OOMP [98] are taken from [96] and GBP from [53]. BP was performed using Matlab's *linprog*. The author's implementation of MP was applied. For GBP, to avoid problems with ill-conditioned matrices a small perturbation has been added to the dictionary matrix as recommended in [53]. Two dictionaries were used: uniform random dictionaries with redundancy 4 (i.e. $K = 4M$) generated for each image in the same way as in [53] and our dictionary $\mathcal{D}_{16}$ designed for MP (see Chapter 4).

BP and OMP (OOMP) converge, as expected, in at most 256 steps, which can be seen in Figure 3.1. MP struggles to converge for a random dictionary. However, for $\mathcal{D}_{16}$ the error is negligible, though not zero, after 256 steps. It is worth noting here that 51 out of the 100 first atoms for the MP decomposition of *Lenna* in Chapter 4 are found in the lowest frequency sub-band. This means there are only 49 atoms across other 15 subbands emphasising the importance of the lowest frequencies for low bit-rates image coding.

BP, OMP and OOMP were applied to solve the sparse representation problem (exact recovery). For BP atoms were sorted by decreasing amplitude as since this is global optimisation method their order in a dictionary does not reflect contribution to a decomposition, For GBP, MP, OMP and OOMP atoms were ordered as they were found. The results obtained are in a similar spirit as analogous results presented for image blocks in [2, p.46]. BP and GBP solve a different optimisation problem therefore their mean square performance is inferior to MP and OMP in considered application. Interestingly, when using a random dictionary this gap become less clear as shown in Figure 3.1. Also in our experiment, GBP finds a decomposition that in the R-D sense (MSE) is inferior to just sorting coefficients found by BP by their absolute value. Final solutions are the same for BP and GBP up to numerical errors. Figure 3.3 shows performance measured by a L_1-norm which shows the expected gap between Basis Pursuit and MP-like methods. BP

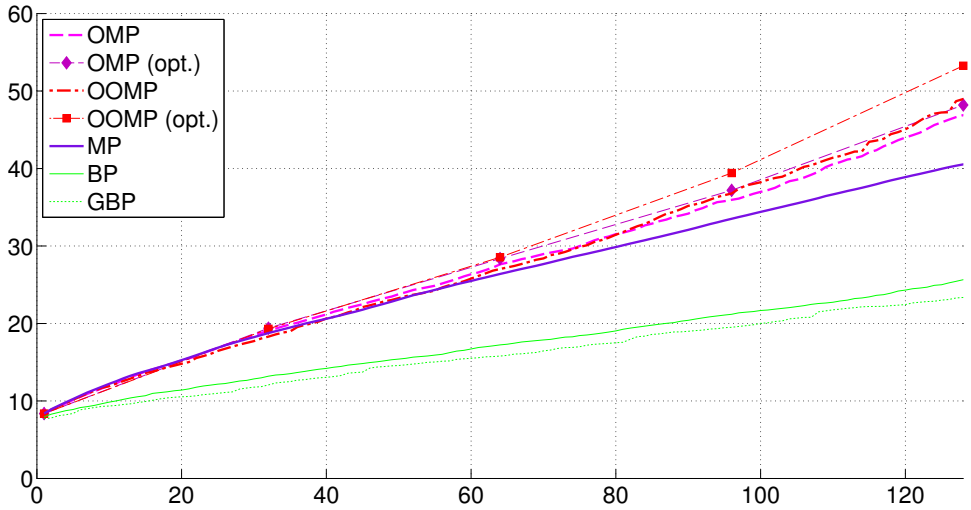


Figure 3.2: Comparisons with OMP targeting fixed number of atoms for the normalised and DC-shifted lowest frequency subband of 5-scale CDF 9/7 wavelet decomposition of grayscale *Lenna*. PSNR (y-axis)[dB] as a function of number of atoms (x-axis).

finds the solution with 25% lower L_1-norm.

Figure 3.2 visualises a case when OMP and OOMP are applied as approximation methods. We can see a clear improvement of OMP over MP and OOMP over OMP for more than 60 atoms. It has to be remembered that in this case we lose progressiveness. The problem can be summarised in the following way: if we use OMP and OOMP and obtain N_0-term and N_1-term decompositions ($N_1 > N_0$) N_0 atoms in N_0-term decomposition are the same as for N_1-term decomposition, however the amplitudes changed (often drastically). For the MP the next atoms are selected without changing previous ones, so that N_0-term decomposition is always a subset of N_1-term. This property together with a good performance at low rates makes MP the method of choice for our coding system. In Section 3.4.2 we will see additional benefits of MP when it comes to quantisation and encoding. If we would use BP de-noising and obtain N_0-term and N_1-term decompositions ($N_1 > N_0$) both decompositions will include not only different amplitudes but also different atoms.

3.4.2 Encoding Atomic Decompositions

It is known that sparse approximations and MP in particular provides a sparser representation of images than DCT or DWT [7]. In practical compression applications an additional issue is quantisation and encoding of the transform coefficients. MP is the most popular choice of a redundant transform in the field of image and video compression [36, 83, 131] regardless the fact that OMP gives a much sparser representation. Quantisation in the case of MP can be done inside a loop [84]. It means that it is possible to choose worse amplitude a_n at nth iteration and correct the error at later stages (see Chapter 5 for details). The advantage of OMP is based on refining the amplitudes which is in contradiction to the necessity of quantising them. Chapter 5 provides the evidence that even very coarse quantisation has a minor effect on the decomposition error. For the reasons above

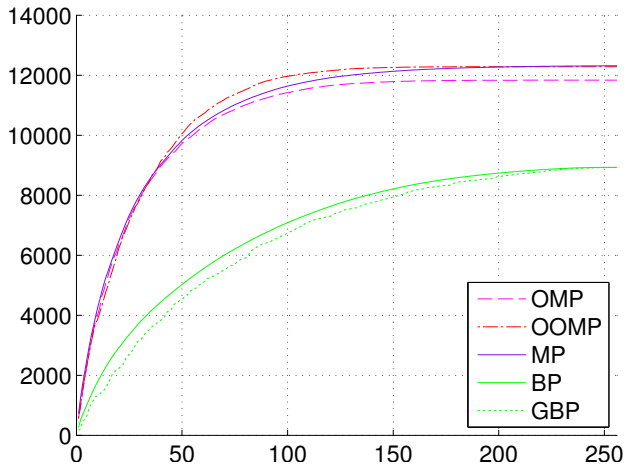


Figure 3.3: Sum of absolute values of amplitudes: $y(x) = \sum_{i=1}^{x} |a_i|$ (y-axis) as a function of number of atoms (x-axis).

plus our focus on low-bit rate coding and significantly lower computational and memory requirements MP is chosen in this work.

The first successful application of MP to 2D signals was video coding of motion compensation residuals in [82]. It has been shown that only a few atoms are required to represent well the motion prediction signal. Once the atoms were found, their amplitudes were uniformly quantised and the residual $R^n f$ was updated by a quantised value [84]. To identify their locations the atoms were grouped into blocks and the difference from a centre of a block was coded using Huffman coding. For reasons of memory and computational complexity, atom search at each iteration was done sub-optimally over an area of size 32×32 with the highest signal energy (Find Energy from [83]). Later [20] it has been shown that a full search gives significantly lower distortion and several speed-ups were proposed e.g. [57].

3.5 Multichannel Matching Pursuit

MP can be extended to decompose vector signals without losing the convergence property [67]. The atom that, according to some criterion, best matches all the components of the input signal is selected. The video codec described in [83] was also capable of coding colour data (see [82, p.27]). Atoms were selected in the YUV colour space. Image channels were decomposed separately. In this way the most of the atoms are assigned to Y channel mimic the idea of sub-sampling from JPEG and JPEG 2000 standards. We show in Chapter 4 that with the idea from [35] and [130] presented in this section we can achieve sparser representation. Multi-channel MP (MMP) for RGB images is summarised by Algorithm 3.3.

Algorithm 3.3 Full Search Multi-channel Matching Pursuit for RGB images.

Initialisation: $R^1 f^{(r)} = f^{(r)}, R^1 f^{(g)} = f^{(g)}, R^1 f^{(b)} = f^{(b)}$.
for $n = 1$ to N **do**
Find atom $g_{\gamma_n} \in \mathcal{D}$ that maximises the L_2-norm:
$\gamma_n = \max_{\gamma \in \Gamma} \sqrt{\langle R^n f^{(r)}, g_\gamma \rangle^2 + \langle R^n f^{(g)}, g_\gamma \rangle^2 + \langle R^n f^{(b)}, g_\gamma \rangle^2}$.
Update residuals:
$R^{n+1} f^{(r)} = R^n f^{(r)} - \langle R^n f^{(r)}, g_{\gamma_n} \rangle g_{\gamma_n}$.
$R^{n+1} f^{(g)} = R^n f^{(g)} - \langle R^n f^{(g)}, g_{\gamma_n} \rangle g_{\gamma_n}$.
$R^{n+1} f^{(b)} = R^n f^{(b)} - \langle R^n f^{(b)}, g_{\gamma_n} \rangle g_{\gamma_n}$.
end for

It has to be noted that theoretical results in [67] were provided for selecting the atom that maximises the absolute value of an inner product over all channels as expressed by Equation (3.20):

$$\gamma_n = \max_{\gamma \in \Gamma} \max_{r,g,b} \left(| \left\langle R^n f^{(r)}, g_\gamma \right\rangle |, | \left\langle R^n f^{(g)}, g_\gamma \right\rangle |, | \left\langle R^n f^{(b)}, g_\gamma \right\rangle | \right). \tag{3.20}$$

In [37], where Algorithm 3.3 was applied in the RGB colour space, atoms with maximal L_2 norm were selected. This corresponds to minimisation of the Mean Squared Error (MSE) over all channels [71, ch.9]. This approach will be also used here.

The data obtained after MP decomposition by the colour codec from [37] include the selected atom parameters and three channel amplitudes. The amplitudes are projected onto a diagonal of the RGB cube and then the distance from diagonal (H), direction (S) and position on the diagonal (I) are quantised and encoded. H and S values are quantised using uniform scalar quantisation and the I component using exponential quantisation from [38]. This is analogous to the representation of amplitudes in HSI colour space. Entropy coding was performed using the same adaptive arithmetic coding as for grayscale [34]. The proposed scheme has been compared with JPEG 2000 showing promising performance at low-bit rates. It has to be remembered that this codec, based on the grayscale codec from [34] extended to colour coding, is computationally extremely complex. The idea of performing the MP after the wavelet transform proposed in [131], used here and described in the next chapter is many times faster (in practice time is reduced from a few hours to a few seconds).

The general idea behind matching the same atom to all three channels is to explore the inter-channel correlations and dependencies of a typical image directly in RGB colour space. Most of the properties of single-channel MP, such as the Parseval-like equality from Equation (3.11), holds for MMP. The choice of an atom can also be suboptimal with a sub-optimality parameter $\alpha \in (0, 1]$. This flexibility brings here the potential to choose the best atom according to quality metrics more correlated than MSE with human visual perception. This topic is touched on in Chapter 6.

3.6 Summary

The main conclusion of this chapter is to highlight MP as a preferable method for sparse approximation of images in application to scalable image coding. Let us summarise its

main advantages. Firstly, the signal norm preservation provides scalability and easy rate control. Sorting atoms by magnitude of their amplitudes is equivalent to sorting them by their contribution to overall signal error. Secondly, the atom search is independent on previous iterations which allows flexibility in coefficients quantisation which can be performed in-loop and in an adaptive way. Quantisation is essential for applications in coding and when performed in-loop allows to recover introduced errors at later stages (see Chapter 5). Furthermore, the performance of MP at low bit rates has been shown to be as good as for OMP or BP. Finally, linear programming is typically of order of magnitude slower for realistic problems [13] and although it has been shown to perform better when applied on sound on image sub-band and considered dictionary it failed to provide representation suitable for scalable image coding.

Even though MP is the simplest among the reviewed sparse approximation methods and even after applying clever algorithmic optimisations its main shortcoming remains the high computational complexity of the atom finding process (encoder). The next chapter addresses the problems of an efficient implementation of MP as well as dictionary design. Atom quantisation and encoding with focus on colour image coding are tackled in Chapter 5.

4 Matching Pursuit in the Spatio-Frequency Domain

This Chapter focuses on implementation of the MP algorithm for grayscale and colour images, its complexity and the design of dictionaries. A dictionary, as already explained in Chapter 3, can be defined as a set of filters. We are interested in choosing a set of filters that gives the best coding performance for a wide range of images. We assume for a while that the number of atoms in a decomposition is on average directly proportional to the size of the stream. Quantisation and encoding of an MP decomposition is a topic in Chapter 5 where we also try to find the optimal size of the dictionary. Therefore in this chapter we evaluate dictionaries mainly by comparing distortion for a given number of atoms. It will be shown that the size and structure of a dictionary are critical parameters for the memory and computational complexity of the decomposition process, and also that with increasing dictionary size above a certain value reduction in distortion is negligible. To achieve a good spatio-frequency representation and consequently a sparser image approximation, MP is applied after the spatio-frequency transform. This means that a dictionary is defined by both a set of mother functions, called here *generators* or *bases*, and the choice of transform. This idea is inspired by [131] where preceding MP with DWT was first performed explicitly.

We start with a review of methods for performing MP in Section 4.1. Different variations of MP of the transformed data are considered and compared. Then the multi-channel extensions are analysed. The effects of different treatment of image borders, choice of image transform and removal of DC-component are evaluated.

Both general and application specific problems of dictionary design and selecting generators are tackled in Section 4.2. It starts with a general formulation of the problem in

signal processing and then refines it to separable dictionaries in the transformed domain. Then dictionaries are trained using the method from [76] and evaluated on the independent test data. Our implementation and its complexity are analysed in relation to dictionary design in Section 4.3. The Chapter is summarised and concluded by Section 4.4.

4.1 MP and Image Transforms

Applying MP directly to represent image pixels has very limited practical application due to the necessity of computing an enormous number of inner products. To overcome this limitation a range of ideas[1] to apply MP to smaller parts of the signal have been suggested.

A video codec proposed in [83] used a heuristic, called *Find Energy*, that finds the block inside a signal with maximal energy (L_2-norm) and then searches for the best atom only around the centre of this block. In practice for 4CIF video frames of dimension 704×576, Find Energy searched among 12×12 overlapping blocks while atom search was performed then inside a block of size 16×16.

A lot of research followed [83] trying to improve coding performance and speed the MP up. The idea of representing a dictionary as a set of band-filters analogously to wavelets was proposed in [22] where a Haar-like approximation of a dictionary from [83] was used for the calculation of inner products.

In other work on the structure of a dictionary [20] not only reduction of complexity but also improvement in performance have been achieved. It was also shown that heuristics, like Find Energy, generally give a higher distortion than a full search over the whole signal [20, 35, 36].

In [36] filters of footprints up to a quarter of the image size were applied to represent image features at different scales. The two 2D generating functions were used Gaussian, $\exp(-x^2-y^2)$, and its second derivative, $(4x^2-2)\exp(-x^2-y^2)$. A dictionary was designed so that the low frequencies are represented by scaled Gaussians. Rotation and scaling of the Gaussian second derivative captures image contours. Scaling was anisotropic which means that it can vary in vertical and horizontal direction. Due to these transformations the filters obtained were inherently non-separable. In order to make it computationally feasible, matching was done after performing a Fast Fourier Transform (FFT). Nevertheless the Full Search MP from [36] still remained computationally extremely demanding. In [35] it was reported to take more than 2 hours for decomposition of 128×128 grayscale *Lenna* to 29.91 dB. The full codec proposed in [36] had coding performance comparable to JPEG 2000 at low bit rates.

In our work, we use the 2D Discrete Wavelet Transform (2D-DWT) and perform MP on all subbands: this approach was originally proposed in [131] to preserve low complexity and to create a dictionary capable of capturing image features at different scales. One-scale subband decomposition was reported independently in [54] to be a straightforward way to improve performance of a given dictionary in video coding. The method from [131] is comparable in coding performance to the JPEG 2000 standard and has a practical com-

[1] Unless otherwise stated, the ideas reviewed here were applied to single-channel signals.

image	PSNR			M-SSIM		
	DCT	CDF 9/7	Haar	DCT	CDF 9/7	Haar
airplane512x512	25.64	26.56	24.80	0.7331	0.7869	0.7164
baboon512x512	20.75	21.01	20.59	0.3986	0.4287	0.3806
barbara720x576	22.90	23.52	22.18	0.6004	0.6519	0.5378
goldhill720x576	26.35	27.00	25.90	0.6135	0.6485	0.5922
house768x512	22.12	22.57	21.89	0.5133	0.5508	0.5009
lenna512x512	26.85	28.34	25.88	0.7197	0.7845	0.6790
lighthouse768x512	23.57	24.13	23.42	0.6452	0.6775	0.6416
motorcross768x512	19.57	20.10	19.22	0.3807	0.4227	0.3612
parrots768x512	28.93	30.51	28.29	0.8035	0.8549	0.7811
peppers512x512	26.21	27.68	25.34	0.6887	0.7510	0.6471
sailboat512x512	23.21	23.90	22.57	0.5898	0.6366	0.5610
sailboats512x768	26.81	27.64	26.68	0.7397	0.7788	0.7387
average	**24.41**	**25.25**	**23.90**	**0.6189**	**0.6644**	**0.5948**

Table 4.1: Comparison between image transforms for performing MP.

putational complexity (encoding takes a few seconds on the modern PC, see Section 4.3.1 for details). Extension into Multi-channel Matching Pursuit (MMP) has been proposed and applied to grayscale video coding [130] where it was performed on Groups of Pictures in temporal direction. In Section 4.1.1 and in Chapter 5 we will apply it to decompose RGB images.

4.1.1 MP Performed on Subbands

We provide here some arguments on why using a subband transform with MP can reduce complexity and improve coding performance. We also justify use of CDF 9/7 wavelets filters from JPEG 2000 standard.

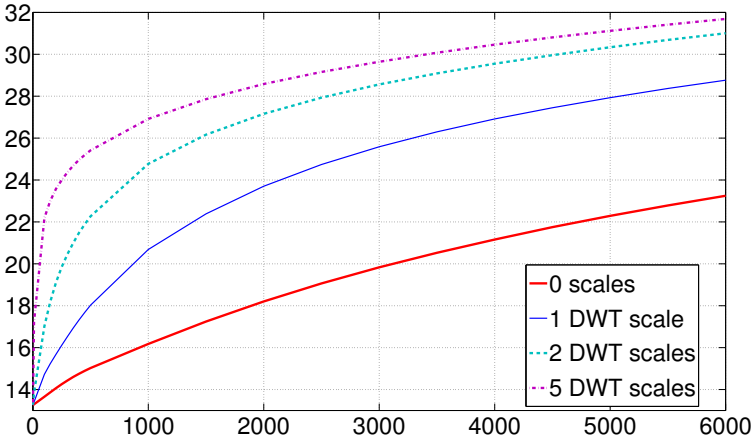


Figure 4.1: PSNR performance in dB (y-axis) for a given number of atoms (x-axis) using different numbers of wavelet scales (grayscale *Goldhill*).

The main argument is based on applying redundant dictionary to improve sparsity of the transformed data. Coding schemes such as SPIHT and EBCOT attempt to take advantage from sparse representation of image in the wavelet domain by directly quantising and coding wavelet coefficients. Encoding groups of coefficients rather than single numbers by matching functions from redundant dictionary using MP algorithm allows to achieve much sparser representation. We observed that by introducing redundancy we can half the number of atoms needed to represent an image to the same PSNR. For example, for *Lenna* image to achieve the same PSNR of 27.23 dB for 2000 quantised wavelet coefficients we need 1735 atoms if using a dictionary with 2 generators ($\mathcal{D}_2$), 1045 atoms using 4 generators and only 751 atoms using dictionary $\mathcal{D}_{16}$ that will be introduced in Section 4.2.2. These figures confirms a known fact [7, 36] that enriching a set of functions in a dictionary significantly improves the sparsity.

Additional argument for two-stages process involving image transform and MP is based on the fact that it can be computationally easier to match filters in the spatio-frequency domain. Filters applied locally in the spatio-frequency domain correspond to global structures in the image domain. To give an example for the discrete case, consider a dictionary entry with support W: $g(t) = 1/\sqrt{W}$ for $t = 1, 2, ..., W$. Its DCT (and DFT) is the Dirac delta $g(\omega) = [1]$ with support 1. Performing an inner product with such a short signal requires only one multiplication. It is known that for transforms like DCT, DFT or DWT long-support functions have short support after transformation, and hence MP is computationally more efficient in the transform domain. A similar idea was used indirectly in [36] where convolutions with filters in a dictionary were performed in the Fourier domain.

Let us look how this affects an update step of the MP algorithm. If the transform T is linear and preserves an inner product,

$$\langle f, g \rangle = \langle T\{f\}, T\{g\} \rangle \text{ for all } f, g \in \mathcal{H}, \tag{4.1}$$

then the MP decomposition of signal f (see Equation (3.9)) obtained in the transform domain is:

$$T\{f\} \approx \sum_{n=1}^{N} \langle T\{R^n f\}, T\{g_{\gamma_n}\} \rangle \, T\{g_{\gamma_n}\}. \tag{4.2}$$

If the inner product is preserved then selecting the best matching atom in the transform domain also maximally reduces an error in the spatial domain.

The natural question arises about the choice of the transform T. In [131] filters designed for video coding in the image domain were applied to wavelet subbands after performing 2D-DWT with CDF 9/7 filters from the lossy mode of JPEG 2000. 2D-DWT is not exactly an orthonormal transform but with appropriate normalisation (see Chapter 2) can be practically treated as such [102]. Thanks to the energy compaction property of the DWT, the atoms found in the wavelet domain in initial iterations have high amplitudes. Hence, they contribute more to the whole image energy resulting in higher PSNR for the same number of atoms. Figure 4.1 shows PSNR as a function of atom numbers and different number of wavelet scales. The results highlight that the dictionary D_{16} applied to wavelet subbands gives a representation which is sparser by a few orders of magnitude

(a) Haar, 5 scales, zero-padding,
PSNR=25.88 dB, M-SSIM=0.6790.

(b) DCT, 16x16 blocks,
PSNR=26.85 dB, M-SSIM=0.7197.

(c) CDF 9/7, 5 scales, zero-padding,
PSNR=27.89 dB, MSSSIM=0.7732.

(d) CDF 9/7, 5 scales, symmetric ext.
PSNR=28.34 dB, M-SSIM=0.7824.

Figure 4.2: Grayscale *Lenna* decomposed using 1000 atoms and a dictionary of 16 bases with different transforms and border treatments.

than if the same dictionary was applied in the image domain. Further, MP with DWT gives significantly better performance than with DCT as can be seen in Figure 4.2. It has to be noted, though, that the choice of suitable wavelet filters is critical. Highly regular filters such as CDF 9/7 mentioned in Chapter 2 are preferred. Figure 4.2 shows comparison between CDF 9/7 and Haar wavelets. The latter performs even poorer than DCT in terms of PSNR.

One of the technical problems with wavelets is the treatment of image borders while performing filtering operations. Generally, as mentioned in Chapter 2, the best results in imaging application are obtained by symmetric periodic extension. Figure 4.2 displays the different results obtained when using different wavelet filters and different border treatments. Indeed periodic symmetric extension is by far more effective than, for example, zero-padding.

The small number of 1000 atoms was chosen to visually highlight compression artefacts introduced by different set-ups. Table 4.1 collects the results of comparisons between

(a) DCT, 16x16 blocks,
PSNR=22.90 dB, M-SSIM=0.6022.

(b) CDF 9/7, 5 scales, symmetric ext.,
PSNR=25.38, M-SSIM=0.7027.

Figure 4.3: Grayscale *Lenna* decomposed using 1000 transformed coefficients.

Haar and CDF 9/7 wavelets against DCT. The difference in coding performance in terms of PSNR between the simplest Haar wavelets and CDF 9/7 from JPEG 2000 can be as high as 2 dB. On average the gap between Haar and CDF 9/7 filters is 1.5 dB while the gain achieved by the smooth wavelets over DCT is 0.84 dB on average. The gaps in performance are consistent across all images and hence statistically significant when analysed using methodology introduced in Section 2.3.4. Moreover, using non-smooth wavelets such as the Haar transform introduces even more annoying visual effects, in a form of non-uniform blocking, than applying block-based DCT, as can be seen in Figure 4.2 (b).

These conclusions are generalisation of the known facts about image transforms. The commonly used thresholding of the raw transformed coefficients (see Chapter 2) can be viewed as the Matching Pursuit in the transform domain with a dictionary that contains just one one-pixel-long basis (Dirac delta). In Figure 4.3 an image is composed only from raw transformed coefficients. Two known facts about the MP (see also for example [7] or [17]) are visualised.

Firstly, by comparing Figure 4.3 and Figure 4.2, it is clear that MP with a redundant dictionary gives a much sparser representation of an image than wavelets. To achieve the same PSNR=28.34 dB as for MP and a dictionary with 16 bases (Figure 4.2) 2450 raw wavelet coefficients are needed. Interestingly using raw wavelets coefficients to achieve the same PSNR results in lower M-SSIM (M-SSIM=0.7784 in this case). This suggest the potential for better visual appearance of images decomposed using MP since M-SSIM correlates better with human perception of image quality.

Secondly, the block-based transforms introduce annoying blocking artefacts and perform much poorer in an R-D sense than smooth wavelets. To provide a guideline about partitioning that minimises distortion and keeps the computational and memory cost of an encoder tractable some more comments have to be made regarding blocking. Processing fixed-size blocks makes program optimisation, including the potential for parallel and

(a) 4x4-block-DCT, 16 subbands, PSNR=25.64 dB, M-SSIM=0.6516.

(b) 32x32-block-DCT, 1024 subbands, PSNR=26.78 dB, M-SSIM=0.7259.

(c) CDF 9/7, 5 scales, 16x16 blocks, PSNR=28.10 dB, M-SSSIM=0.7776.

(d) CDF 9/7, 5 scales, 32x32 blocks, PSNR=28.28 dB, M-SSIM=0.7818.

Figure 4.4: Grayscale *Lenna* decomposed using 1000 atoms and a dictionary of 16 bases with different numbers of blocks in the DCT and DWT domain.

GPU implementations, easier. Many compression schemes based on redundant representations partition an image into non-overlapping blocks [8, 51, 109] speeding up the process but keeping the problems of blocking artefacts unsolved. For example, our MATLAB implementation designed for processing small fixed-size blocks can be in practice a few times faster than full search in wavelet subbands. Implementation details of the MP are left for Section 4.3. Here, the relation between block size and R-D performance is investigated.

Figure 4.4 shows the effect of using different block sizes for MP with DCT. Performing DCT on 4x4 blocks means transforming the image into 16 subbands, which is the same number as after 5-scale DWT. Hence the results are similar to Haar wavelets from Figure 4.2. Experiments show that the optimal PSNR is achieved when applying DCT on blocks of size 16×16 which, for 512×512 images, means subbands of size 32×32. A similar type of partitioning of DWT coefficients can also be used. On average for 10 images [2] and 6000 atoms full subband search gives: 31.34 dB, partitioning into non-overlapping

[2] We excluded *Barbara* and *Goldhill* as they cannot be split into equal blocks of size 32×32.

32×32 blocks: 31.19 dB and 16×16 blocks: 30.92 dB. The lower R-D performance could be compensated for by an increase in encoding speed. In fact a speed-up is only observed for partitioning into 16×16 blocks.

We have shown that applying MP after the spatio-frequency transform improves sparsity of decomposition. DCT and DWT were analysed but any pre-transform could be chosen. It has to be remembered that by using Fourier or wavelet basis we have to accept its shortcomings. In Section 2.4.3 Complex Wavelets ($\mathbb{C}$WT) have been introduced as the alternative to wavelets. The Dual-tree Complex Wavelet Transform was presented as possible implementation of the concept of complex wavelets to improve representation of directional features in images [101]. It could be used to represent images within an MP-like framework [62, 128]. A potential issue with transforms such as $\mathbb{C}$WT is that contrarily to DWT and DCT it is a redundant transform. In case of Dual-tree implementation from [62] four times redundancy is introduced. It was shown in [128] that complex wavelets provide less sparse representation than the DWT. The coding scheme introduced in Chapter 5 relies on the use of a basis transform. We assume that only one parameter, the atom index (see Section 5.2), accounts on extended data dimensionality from the use of redundant transformation. Therefore, in the general form introduced in Chapter 5, our codec is not expected to perform well when used with the redundant pre-transform. Nevertheless, some promising results have been recently presented with complex wavelets applied to grayscale image coding using known coding algorithms such as EBCOT from JPEG 2000 or SPIHT [128]. The main focus here is on representing multichannel images and encoding colour decomposition after Multichannel Matching Pursuit in the transform domain. Complex Wavelets are beyond the scope of this work but can be seen as an interesting alternative to DWT that preserves directional features in images. Further investigation is required to study a potential of Complex Wavelets to represent multi-channel images.

4.1.2 Multichannel MP with DCT and DWT

If we extend MP to multi-channel signals according to Algorithm 3.3 then most of the conclusions from the previous subsection transfer across directly. For example, the difference between colour MP in DCT and DWT is presented in Figure 4.5. The differences in PSNR values is similar to the grayscale image.

However, it is remarkable that we need only 1105 colour (RGB) atoms to achieve the same quality in terms of Y-PSNR as 1000 atoms in the grayscale case (Figure 4.2). We know that most of image information is stored in Y-channel. Multi-channel MP can recover it directly in RGB colour space. This confirms the ability of multi-channel algorithm to exploit correlations between RGB channels. However, if we increase the number of grayscale atoms there are more and more colour atoms needed to recover Y-channel. The results of an illustrative experiment are presented in Table 4.2 for the image *Lenna*. The Y-channel of the original colour image after RGB$\rightarrow$YCC transform, was decomposed with a single-channel MP and a specified number of atoms. Then MMP was iterated on the original RGB-colour image until Y-PSNR reached the value obtained for a single-channel.

(a) DCT, 16x16 blocks,
Y-PSNR=26.94 dB,
RGB-PSNR=26.43 dB,
Y-M-SSIM=0.7218.

(b) CDF 9/7, 5 scales, symmetric ext.,
Y-PSNR=28.34 dB,
RGB-PSNR=27.67 dB,
Y-M-SSIM=0.7833.

Figure 4.5: *Lenna* decomposed using 1105 atoms and 16-generators-dictionary.

For smaller numbers of atoms only around 10% more atoms are needed to represent both grayscale and colour (chroma) information. From Table 4.2 it is clear that this number increases with the number of iterations. The interpretation of this result is that from some stage in the application of MP there is mainly noise left to be decomposed resulting in less correlation between channels and less benefit from the multi-channel algorithm.

The advantage of the multi-channel algorithm is even more visible if we compare its performance against single-channel MP applied to each of the three channels independently. This approach was used in the colour extension of the historical video codec from [83], mentioned in Section 4.1.1. MP was applied in YC_bC_r colour space. The Find Energy heuristic was calculated for Y, C_b and C_r channels. The block with the highest energy was selected and then the best atom was found within this block (see Section 4.1 and [83]).

The same procedure can be adapted to our Full Search MP scheme (see Section 3.2.1). We simply pick an atom with globally the highest absolute value of the inner product instead of searching for the highest energy block. We shall compare sparsity obtained

Number of gray atoms	1000	2000	5000	10000	20000
Number of colour atoms	1105	2267	5940	13060	30230
Difference ratio (%)	10.50%	13.35%	18.80%	30.60%	51.15%
Y-PSNR	28.34	31.08	35.04	38.07	41.34
RGB-PSNR	27.66	30.02	33.21	35.64	38.58
Y-M-SSIM	0.7831	0.8417	0.9020	0.9352	0.9647
Grayscale M-SSIM	0.7845	0.8428	0.9025	0.9350	0.9635

Table 4.2: Number of colour atoms needed by the MMP with DWT (5 scales, CDF 9/7) to obtain the same quality decomposition of the Y-channel as single-channel MP for *Lenna*.

by single-channel MP in YC_bC_r against MMP. When this approach was applied to the *Lenna* image, decomposed into 6000 atoms: 4326 atoms were found in Y channel and only 816 in C_b and 858 in C_r. The Y-PSNR of such a decomposition was 34.36 dB compared to 35.08 dB obtained by multi-channel decomposition, and the Y-M-SSIM values are 0.8626 and 0.9025 respectively, favouring MMP in RGB colour space. If a single-channel decomposition is done directly in RGB space then we get 1806 red, 2366 green and 1828 blue atoms. As this approach ignores inter-channel correlations, it is not surprising that the quality cannot even compete with MP in YC_bC_r: we get RGB-PSNR=30.65 dB for single-channel MP in RGB space while MMP achieves 33.24 dB and single-channel MP in YC_bC_r space: 32.01 dB, for the same number of 6000 atoms. Using RGB-PSNR, the difference between MMP in RGB and MP in YCC is obviously much higher than using Y-PSNR as the first method minimises the joint error (hence maximises RGB-PSNR) while the second maximises Y-PSNR.

These experimental results confirm better decorrelating properties of the joint decomposition (MMP in RGB) over decorrelating colour space transform (MP in YCC). It has to be noted that since Multi-channel MP potentially generates more data per atom (three amplitudes rather than single one) quantisation and encoding should be taken into account to ensure a fair comparison of compression performance. This is analysed in Chapter 6 with use of multi-channel coding method introduced in Chapter 5. When coding is taken into account both methods performs similarly.

4.1.3 Zero-Mean Signals

The last point to discuss about the transform part of image codecs is the fact that it is convenient theoretically [102] to consider zero-mean signals before applying an image transform. The JPEG standard includes a processing step in which the value 128 is subtracted from each pixel in an 8-bit image. Here, by mean-shifting (or DC-shifting as referred to by JPEG) we mean subtracting the mean pixel value quantised to 8-bits from each channel. Such a value can be sent to the encoder as an 8-bit number and our signal becomes zero-mean up to quantisation. In fact mean-shifting is not only convenient but also gives a slight but statistically significant improvement in PSNR. For the codec proposed in this thesis, the comparison done for 1000 atoms on 12 standard test images shows that in the DWT domain the average PSNR equals 25.25 dB with mean-shift against 25.19 dB without. The same is true for colour images for which, with the same settings, we have: 24.42 dB against 24.34 dB. The general rule is the more wavelet scales the smaller the difference. This is due to the fact that the non-zero mean is only the lowest frequency subband which preserves the DC component. Nonetheless, difference is always statistically significant in our experiments, even if the sample is of only 12 images. For each single image mean-shift gave a small improvement ranging from 0.02 dB to 0.12 dB, with standard deviation of differences less than 0.04 dB for both grayscale and colour.

As the gain from using mean shifting seems negligible we provide the evidence that it is significant. Performing a paired-sample t-test gives p-values 0.0004 for grayscale and 0.0017

for colours with corresponding 95% confidence intervals within $[0.03, 0.09]$ and $[0.03, 0.12]$ respectively. For the same comparisons according to M-SSIM metric, we have, for grayscale $\overline{\text{M-SSIM}} = 0.6664$ against $\overline{\text{M-SSIM}} = 0.6626$ and for colour $\overline{\text{Y-M-SSIM}} = 0.6551$ against $\overline{\text{Y-M-SSIM}} = 0.6528$, with p-value for a paired-sample t-test less than 10^{-4} in both cases. Due to this minimal but consistent gain we always use mean-shifting before the transform step.

4.2 Design of the Dictionaries

In Section 4.1 we analysed the properties of different transform used in combination with Matching Pursuit. Problem of building a dictionary is often specified as general optimisation problem in the image domain. In practice, due to the size of the image data we have to restrict our search to a smaller subspace. In this section a few dictionaries are designed and evaluated.

4.2.1 Separable 2D-Dictionaries

A 2D dictionary of size K, in the most general form as introduced in Chapter 3, is defined as a set of K matrices:

$$\mathcal{D} = \{g_1, g_2, \ldots, g_K\}. \tag{4.3}$$

For image representation we are interested in translation-invariant dictionaries which means that $\mathcal{D}$ is generated by locating a subset of $B \ll K$ matrices $\{g_\lambda\}_{\lambda=1,2,\ldots,B}$ at each point in the image:

$$\forall_{i\in\{1,\ldots,K\}}\ g_i(x,y) = g_\lambda(x - t_x, y - t_y), \text{ for some } \lambda, t_x, t_y. \tag{4.4}$$

Approaches to dictionary design can be classified into two categories. In the first category, mother functions (i. e. generators g_λ) are selected according to some image model as in [36], reviewed in Section 4.1.1, where the Gaussian and its second derivative were chosen. The second category is based on training a dictionary on a set of images. In the seminal work of Neff and Zakhor [83] video residuals were decomposed using huge dictionaries. Functions that occur in decompositions of training data more often were kept while the others were removed.

Due to the complexity issues (see Section 4.3.1) we focus on dictionaries that are separable, i. e. each matrix g_λ can be represented as a tensor product of vectors. A dictionary is defined by b 1D filters ($B = b^2$) and has the following form:

$$\mathcal{D}^{(sep)} = \left\{ g_{\lambda_x} \otimes g_{\lambda_y} \right\}_{\lambda_x,\lambda_y=1,2,\ldots,b}. \tag{4.5}$$

From here on we identify a dictionary by the set of its separable generators g_{λ_i}, for $\lambda_i = 1, 2, \ldots, b$.

The dictionary $\mathcal{D}^{(0)}$ from [83] was separable and translation-invariant and contained $b = 20$ generators which corresponds to $B = 400$ separable 2D bases. Improvements over $\mathcal{D}^{(0)}$ based on the same top-down principle (removing bases from a bigger dictionary)

which was carried out in [20] resulting in a more efficient (both computationally and in terms of R-D) dictionary $\mathcal{D}^{(1)}$ consisting of 16 1D bases. Later, the contribution of [76] has shown that the even better results can be achieved by building a dictionary with a bottom-up approach, i. e. iteratively building up a dictionary by adding one basis at a time.

4.2.2 Basis Picking

The algorithm, proposed in [76], called Basis Picking creates a sequence of dictionaries: $\{\mathcal{D}_s\}_{s=0,1,2,\ldots,S}$ with increasing numbers of generators. We have: $\mathcal{D}_0 \subset \mathcal{D}_1 \subset \mathcal{D}_2 \subset \ldots \subset \mathcal{D}_S$ and at each step $s > 0$ one function, denoted as g_s, is removed from a large set of candidates and added to form a new dictionary:

$$D_{s+1} = D_s \cup \{g_s\}. \tag{4.6}$$

This implies that if $D_0 = \emptyset$ then the size of the s-th dictionary is s. In Monro's original experiments grayscale images were decomposed with a fixed number of atoms and the candidate set built from parametrised Gabor atoms of the form given by Equation (4.7) and one Dirac delta. Codebooks for both grayscale still images and video residuals have been trained and tested in [76, 79].

$$g_{(\sigma,f,w,\phi)}(t) = K_{(\sigma,f,w,\phi)} \exp\left(-\frac{\pi}{4\sigma}t^2\right) \cos\left(\frac{\pi f t}{w} + \phi\right). \tag{4.7}$$

All the bases g are sampled at $2w + 1$ points: $t = [-w, \ldots, 0, \ldots, w]$ and normalised by factor $K_{(\sigma,f,w,\phi)}$ to have unit norm as required by the MP algorithm [72]. Therefore they can be represented as vectors of length $l = 2w + 1$: $g = [g(1), g(2), \ldots, g(l)]$. In [76] the lengths (footprints) of bases in the candidate set varied from 1 to 15 (w from 1 to 7). The distribution of bases (filters) in the candidate set according to their footprint is given in Table 4.3. The values of the parameters σ, f, ϕ were taken to be: $\sigma \in \{1, 2, 4, 8, 12, 16, 20, 24\}, f \in \{0, 1, \ldots, w\}, \phi \in \{0, \pi/8, \pi/4, 3\pi/8, \pi/2\}$. If the maximum footprint is 15 then selection is performed from a candidate set of 946 distinct functions. At each iteration a function, that for a fixed number of atoms (6000 atoms are taken), reduces the distortion the most (i. e. maximises PSNR), is selected.

footprint	1	3	5	7	9	11	13	15
number of generators	1	16	49	96	136	176	216	256
candidate set size	1	17	66	162	298	474	690	946

Table 4.3: Number of generators in the candidate set by their footprints.

Since the sizes of generators have a large effect on the complexity of the encoder (see Section 4.3 for more details) we are interested in short-support discrete filters matched in the transform domain as already described in Section 4.1. The next section analyses the effects of different parameters on training and evaluates the resulting dictionaries.

Trained on:	Gabor				Walsh			Mixed
max. length:	5	7	9	15	7	8	10	9(8)
Averaging over grayscale decompositions into 1000 atoms.								
Grayscale Lenna	24.43	24.56	24.63	24.67	24.71	24.69	24.69	24.74
Grayscale Goldhill	24.47	24.63	24.67	24.69	24.71	24.70	24.74	24.76
Colour Lenna	24.44	24.61	24.65	24.63	24.64	24.69	24.69	24.71
Colour Goldhill	24.45	24.64	24.62	24.71	24.71	24.70	24.74	**24.77**
Averaging over colour decompositions into 1000 atoms.								
Grayscale Lenna	23.67	23.81	23.86	23.89	23.91	23.90	23.90	23.95
Grayscale Goldhill	23.71	23.86	23.90	23.91	23.91	23.88	23.93	23.97
Colour Lenna	23.69	23.85	23.88	23.84	23.86	23.91	23.90	23.94
Colour Goldhill	23.71	23.87	23.85	23.92	23.90	23.92	23.93	**23.98**
Averaging over grayscale decompositions into 6000 atoms.								
Grayscale Lenna	30.35	30.56	30.61	30.63	30.66	30.66	30.65	30.75
Grayscale Goldhill	30.38	30.61	30.66	30.67	30.67	30.66	30.62	30.72
Colour Lenna	30.34	30.62	30.64	30.63	30.57	30.64	30.64	30.70
Colour Goldhill	30.38	30.63	30.58	30.66	30.66	30.66	30.66	**30.75**
Averaging over colour decompositions into 6000 atoms.								
Grayscale Lenna	29.13	29.30	29.34	29.35	29.37	29.35	29.36	29.45
Grayscale Goldhill	29.15	29.33	29.38	29.37	29.37	29.34	29.31	29.42
Colour Lenna	29.13	29.35	29.37	29.35	29.31	29.35	29.36	29.43
Colour Goldhill	29.16	29.35	29.32	29.37	29.35	29.36	29.36	**29.45**

Table 4.4: PSNR (RGB-PSNR for colour images) averaged over 10 test images for dictionaries composed of 16 bases.

4.2.3 Comparing Dictionaries

Properties of the training method of choice are visualised using the colour and grayscale (i. e. Y channel after YCC colour transform) *Goldhill* and *Lenna* as training images. The remaining 10 images are used as a test set. We trained the dictionaries for colour and grayscale representation separately. In all the experiments we initialised a dictionary: $\mathcal{D}_1 = \{g_1\}$, where $g_1 = [1]$ is a single pixel (Dirac delta), as g_1 is always picked in one of the first iterations (if not in the first) anyway [79]. Table 4.4 collects average performances of different dictionaries trained using Basis Picking. Considering a Gabor candidate set, our results confirm those in [79] that searching for filters of size greater than 7 increases computational complexity, as we are selecting from 162 candidates rather than from 946, while the improvement in overall performance is negligible. Results of the t-tests performed to compare the averages shown in Table 4.4 indicate that there is either no statistical difference between the average performance of the dictionaries or the selection of the longer filters can cause over-fitting. For example, comparing a dictionary trained from 162 candidates on the colour *Lenna* image on decompositions into 6000 atoms

against those picked from 946 and 298 (see Table 4.3) gives p-values for the t-test 0.3496 and 0.1228 respectively. For colour *Goldhill* as a training image the corresponding p-values are $p = 0.1617$ (for comparing footprints up to 7 against 15) and $p = 0.0442$ (for comparing footprints up to 7 against 9) suggesting that with a confidence level 5% the shorter support dictionary can even have a slightly higher average performance. However, limiting search to footprints smaller than 7 degrades performance significantly. Comparing candidate sets with footprints up to 7 against up to 5 gives p-values less than 10^{-4}, for example for colour *Goldhill* for 6000 atoms $p = 1.5961 \times 10^{-5}$, thus indicating significantly poorer average performance of the smaller-support dictionaries. Furthermore, there is no significant difference between dictionaries trained on different images, even though *Goldhill* and *Lenna* are very different images with *Goldhill* being highly detailed. Similarly the same quality dictionaries are obtained regardless of whether the training is performed on grayscale or on colour. This is due to the use of short-support filters which are more likely to be efficient for a wide range of images. Later in this section we use those findings to design improved general-purpose dictionaries for MP in the wavelet domain.

The shortcoming of the Basis Picking algorithm, on top of the inevitable sub-optimality is restricting the resulting dictionary to be a subset of a predefined candidate set. Gabor atoms, proposed in [79], are useful to represent smooth structures in the image data. We argue here that in order to benefit from the MP algorithm after preprocessing the image with DWT we need to represent as general signal features as possible using short-support filters. Therefore we tried replacing a Gabor candidate set with the functions defined by Equation (4.8) for different footprints n:

$$g_{i,n}(t) = (-1)^{i_t} \frac{1}{\sqrt{n}}, \tag{4.8}$$

where $i = 0, 1, \ldots, 2^n - 1$ and $i_{n-1} \ldots i_1 i_0$ is a binary representation of i such that:

$$i = \sum_{t=0}^{n-1} i_t 2^t. \tag{4.9}$$

Equation (4.8) generates 2^n functions of footprint n. Half of them can be discarded as the negatives of the other half and we need to consider only 2^{n-1} functions of footprint n. This candidate set includes Walsh functions as a subset, therefore we refer to it as the Walsh-like candidate set. The size of the whole candidate set with bases of a maximal footprint n, i. e. with the functions of a support from 1 to n, is $2^n - 1$. If we train on the Walsh-like set as an alternative to the Gabor then comparable or slightly better performance can be achieved (see Table 4.4) for comparable sizes of the candidate sets when picking from a candidate set of 127 (footprints up to 7) or 255 (footprints up to 8) bases. Interestingly, performance does not degrade significantly if we limit the candidate set to short-support bases. Taking atoms of the footprints as low as 5 will still give a reasonable performance. An increase in the footprint, similarly as for Gabors, does not improve performance and causes exponential expansion of the candidate set.

One of the next possible steps towards an improved dictionary is to merge together Walsh and Gabor candidate sets. Small but consistent improvements in PSNR can be

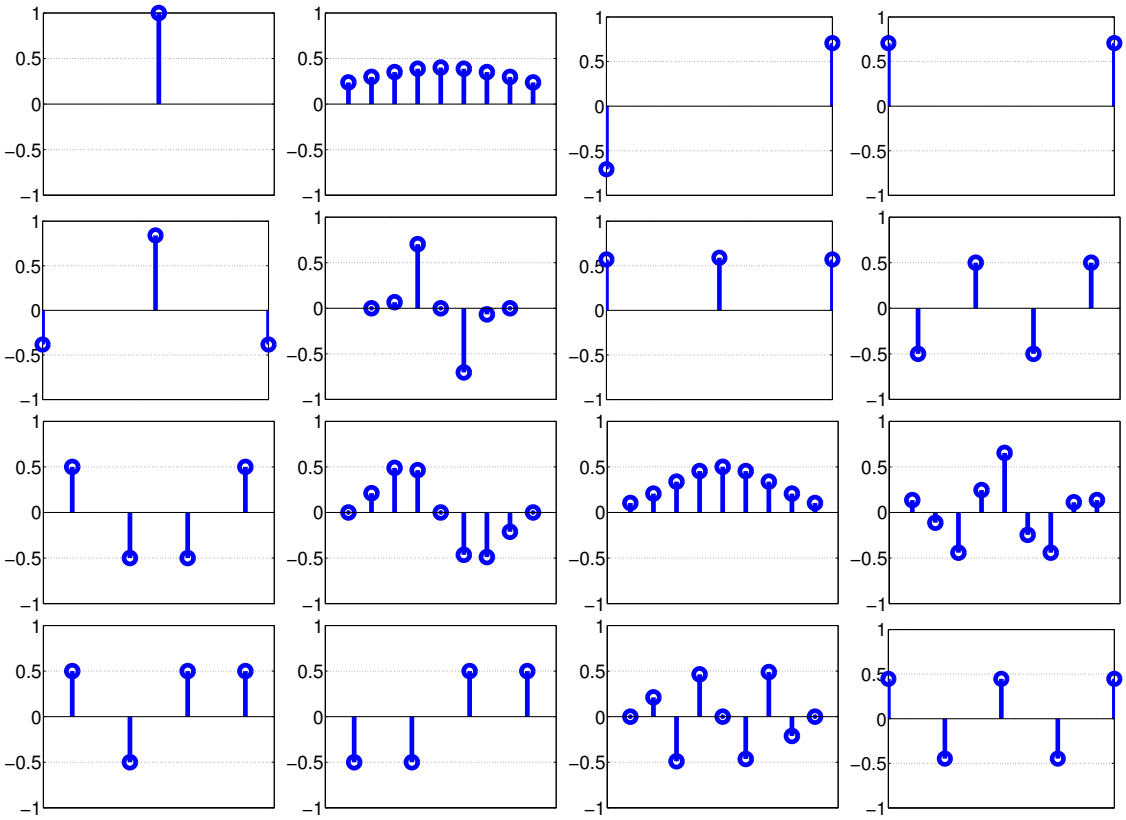


Figure 4.6: Colour dictionary trained on *Goldhill* with bases in the order they are picked during training.

achieved as shown in Table 4.4. Picking from the combination of the set of Gabors of footprint up to 9 and the set of Walsh-like functions up to 8 gave the dictionary with the best average performance when trained on the *Goldhill* colour image. 1D filters trained in this way for colour and grayscale *Goldhill* are shown in Figure 4.6 for colour and Figure 4.7 for grayscale. From now on, the colour dictionary visualised in Figure 4.6 is denoted by $\mathcal{D}_{16}^{(t)}$.

The results analysed so far confirm that a dictionary for our MP codec should be built from short-support filters. Moreover all the dictionaries studied so far exhibit very similar performance. Then a natural question is: if only the number of generators and their footprints matter then why not just use any (random) dictionary of a particular structure. The following experiment shows the significance of the difference between the best trained dictionaries from Table 4.4 and the best dictionary picked from randomly generated population of dictionaries. If we know the number of filters and their sizes we can treat a generator of footprint n as a point on an n-dimensional unit sphere. To form a dictionary we generate points uniformly on an appropriate number of such unit spheres. For fairer comparison we generated in this way 4351 random dictionaries of the same structure as the one trained on colour *Goldhill* (Figure 4.7) which corresponds to the total number of MP decompositions done when performing 16 iterations of Basis Picking from a candidate set of 298 bases. We simple take the dictionary with the best performance for the *Goldhill* image and denote it as $\mathcal{D}_{16}^{(rand)}$. It has to be noted that with this idea we are

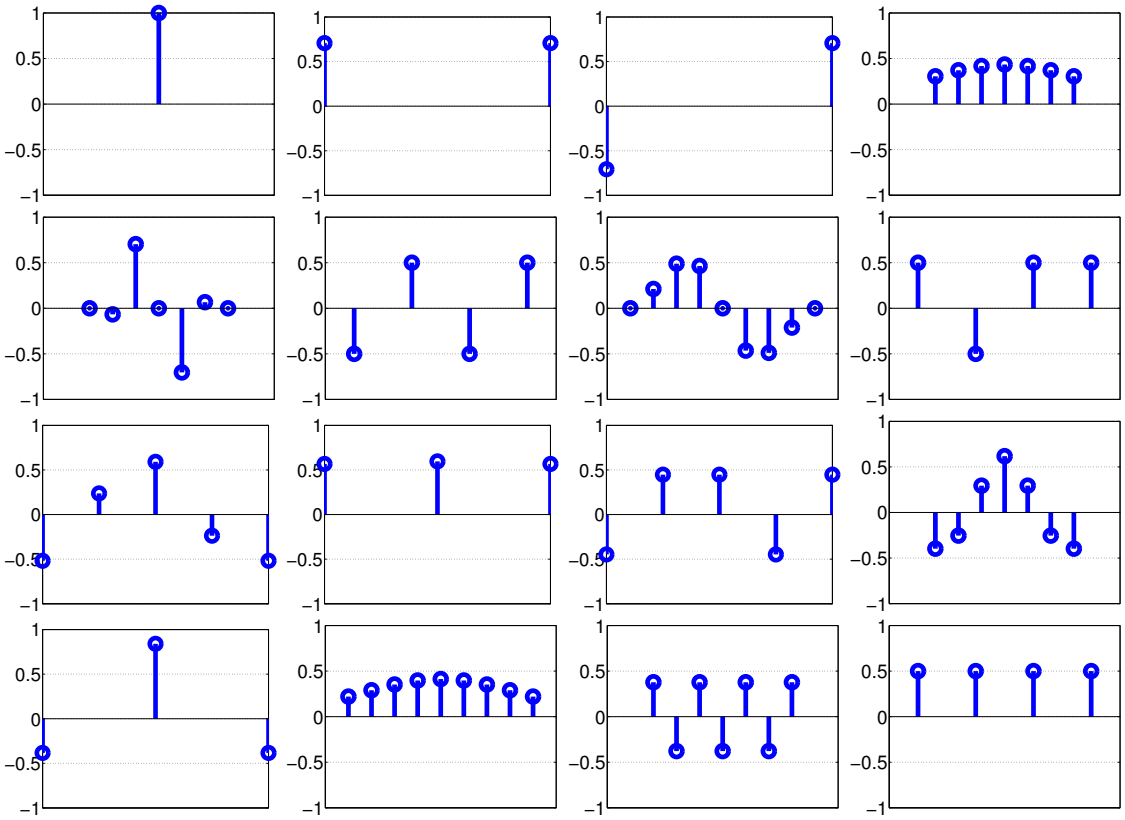


Figure 4.7: Grayscale dictionary trained on Y-channel of *Goldhill* with bases in the order they are picked during training.

not limited to bases from a particular candidate set. Nevertheless the experiments showed (see Table 4.5) that in this way it is not possible to obtain a better dictionary than the one picked by Basis Picking. The results for a random dictionary will serve as a useful reference point when evaluating our further experiments.

Following the fact that picking from a relatively small candidate set of limited-support generators can give a reasonable performance, we can, instead of training, build a separable dictionary analytically based on some of the observations made during the training. Firstly, it can be noted that short-support Walsh-like filters are picked in every training run. Moreover, typically short-support atoms in the dictionary were picked from the Walsh-like subset while the longer ones from the Gabor subset. Therefore, the first dictionary that we build ($\mathcal{D}_{16}^{(b1)}$) consists of the Walsh-like functions of lengths up to 4 and the following Gabor functions: Gaussians of lengths 3, 5, 7, 9 and the modulated Gaussians of lengths 5, 7 and 9. The functions are defined as follows (with normalisation factors

$K_{10}, K_{11}, \ldots, K_{16}$):

$$\begin{array}{rclrcl}
g_1 &=& [1], & g_9 &=& \frac{1}{2}[1,-1,1,-1], \\
g_2 &=& \frac{\sqrt{2}}{2}[1,1], & g_{10}(t) &=& K_{10}\exp(-\frac{\pi}{4}t^2),\ \text{for } t=-1,0,1, \\
g_3 &=& \frac{\sqrt{2}}{2}[1,-1], & g_{11}(t) &=& K_{11}\exp(-\frac{\pi}{8}t^2),\ \text{for } t=-2,\ldots,2, \\
g_4 &=& \frac{\sqrt{3}}{3}[1,1,1], & g_{12}(t) &=& K_{12}\exp(-\frac{\pi}{12}t^2),\ \text{for } t=-3,\ldots,3, \\
g_5 &=& \frac{\sqrt{3}}{3}[1,-1,1], & g_{13}(t) &=& K_{13}\exp(-\frac{\pi}{16}t^2),\ \text{for } t=-4,\ldots,4, \\
g_6 &=& \frac{1}{2}[1,1,1,1], & g_{14}(t) &=& K_{14}\exp(-\frac{\pi}{8}t^2)\sin(\frac{\pi}{8}t),\ \text{for } t=-2,\ldots,2, \\
g_7 &=& \frac{1}{2}[1,1,-1,-1], & g_{15}(t) &=& K_{15}\exp(-\frac{\pi}{12}t^2)\sin(\frac{\pi}{12}t),\ \text{for } t=-3,\ldots,3, \\
g_8 &=& \frac{1}{2}[1,-1,-1,1], & g_{16}(t) &=& K_{16}\exp(-\frac{\pi}{16}t^2)\sin(\frac{\pi}{16}t),\ \text{for } t=-4,\ldots,4.
\end{array} \tag{4.10}$$

This set of generators, from the results we have analysed in this chapter so far, seems to be a reasonable design for a general-purpose dictionary. However, as shown by the results in Table 4.5, it does not perform better than the random dictionary $\mathcal{D}_{16}^{(rand)}$. In fact it is on average indistinguishable from the random one with p-value from the paired t-test $p = 0.2539$.

We introduce here an additional criterion that the functions in the general-purpose dictionary should be as distinct as possible in order to be able to represent a wide range of the signal features. One of the metrics to measure how much the functions in the dictionary vary is *coherence* defined as the maximal absolute value of the inner products between distinct atoms [119]:

$$c(\mathcal{D}) = \max_{i \neq j} |\langle g_{\lambda_i}, g_{\lambda_j} \rangle|. \tag{4.11}$$

The coherence $c(\mathcal{D})$ is always less than 1 and equals 0 only for an orthonormal basis. A smaller value of $c(\mathcal{D})$ indicates less correlation between dictionary entries which can be seen as greater variety in a dictionary. It is worth noting here that in the case of separable dictionaries defined by Equation (4.5) if in the dictionary there are at least two generators g_{λ_1} and g_{λ_2} of sizes 1 and 2 respectively then already $c(\mathcal{D}) \geq \sqrt{2}/2$, (since for any t: $g_{\lambda_1} = [1], g_{\lambda_2} = [\cos(t), \sin(t)]$ implies that $\max |\langle g_1, g_2 \rangle| = \max\{|\cos(t)|, |\sin(t)|\} \geq \sqrt{2}/2$ with equality only for $t = \pi/4$). It can be calculated that for the studied dictionaries: $c(\mathcal{D}_{16}^{(rand)}) = 0.9619$, $c(\mathcal{D}_{16}^{(t)}) = 0.9557$ and $c(\mathcal{D}_{16}^{(b1)}) = 0.9945$.

We now build a dictionary from short-support Walsh-like filters trying to keep the coherence small. The resulting dictionary $\mathcal{D}_{16}^{(b2)}$ is defined as:

$$\begin{array}{rclrcl}
g_1 &=& [1], & g_9 &=& \frac{1}{2}[1,1,-1,-1], \\
g_2 &=& \frac{\sqrt{2}}{2}[1,1], & g_{10} &=& \frac{1}{2}[1,-1,-1,1], \\
g_3 &=& \frac{\sqrt{2}}{2}[1,-1], & g_{11} &=& \frac{1}{2}[1,-1,1,-1], \\
g_4 &=& \frac{\sqrt{3}}{3}[1,1,1], & g_{12} &=& \frac{\sqrt{2}}{2}[1,0,0,1], \\
g_5 &=& \frac{\sqrt{3}}{3}[1,-1,1], & g_{13} &=& \frac{\sqrt{2}}{2}[1,0,0,-1], \\
g_6 &=& \frac{\sqrt{2}}{2}[1,0,-1], & g_{14} &=& \frac{\sqrt{5}}{5}[1,1,1,1,1], \\
g_7 &=& \frac{\sqrt{2}}{2}[1,0,1], & g_{15} &=& \frac{\sqrt{5}}{5}[-1,1,-1,1,-1], \\
g_8 &=& \frac{1}{2}[1,1,1,1], & g_{16} &=& \frac{\sqrt{3}}{3}[1,0,-1,0,1].
\end{array} \tag{4.12}$$

For this dictionary $c(\mathcal{D}_{16}^{(b2)}) = 0.8944$ which is the lowest so far.

Image	Trained $\mathcal{D}_{16}^{(t)}$	Random $\mathcal{D}_{16}^{(rand)}$	Built $\mathcal{D}_{16}^{(b1)}$	Built $\mathcal{D}_{16}^{(b2)}$
airplane512x512	33.86	33.47	33.48	33.85
baboon512x512	23.22	23.14	23.06	23.20
barbara720x576	28.46	27.56	27.92	28.08
house768x512	26.51	26.30	26.29	26.49
lighthouse768x512	28.60	28.43	28.37	28.60
motocross768x512	24.19	23.97	23.91	24.13
parrots768x512	36.29	35.87	35.97	36.29
peppers512x512	31.55	31.27	31.38	31.53
sailboat512x512	27.92	27.76	27.74	27.97
sailboats512x768	33.92	33.34	33.52	33.75
average:	**29.45**	29.11	29.16	29.39

Table 4.5: Comparing the dictionaries built from 16 filters by the RGB-PSNR on 10 decompositions of colour images into 6000 atoms.

In Table 4.5 the dictionaries: $\mathcal{D}_{16}^{(t)}$, $\mathcal{D}_{16}^{(rand)}$, $\mathcal{D}_{16}^{(b1)}$ and $\mathcal{D}_{16}^{(b2)}$ are compared on 10 images. On average in terms of PSNR the trained dictionary $\mathcal{D}_{16}^{(t)}$ remains the best and the random one the worst. However, there is no significant difference between $\mathcal{D}_{16}^{(t)}$ and $\mathcal{D}_{16}^{(b2)}$. In addition the dictionary $\mathcal{D}_{16}^{(b2)}$ includes much shorter generators which significantly reduces the computational complexity of MP (more details will be given in Section 4.3.1). Highly detailed images such as *Baboon* and *Motocross* are known to achieve low PSNR values when compressed with wavelet-based methods. Therefore, also for MP performed with wavelets, the PSNR values obtained for those images are equally low regardless of the dictionary. It has to be remembered that for the system considered here there are two stages that define a dictionary. The first is a choice of the spatio-frequency domain. If the DWT is chosen its disadvantages have to be accepted. The ideas introduced here for a selection of dictionary generators are not limited to wavelets. In the next section, details of the MP implementation for this work are given and then the dictionaries are analysed from the perspective of computational complexity.

4.3 Implementation and Complexity

4.3.1 Complexity of Subband Implementation

The MP algorithm is implemented in this work in a similar way to the *full 2D separable inner product search* from [129]. The maximal inner products and the corresponding atom indexes are stored for each location and for each wavelet subband. At each iteration, inner products have to be recomputed only on a sub-area of one subband. For colour coding it has to be done for all channels and requires approximately three times more multiply-accumulate operations than for grayscale. Recalculating inner products is where the majority of computations is done within this approach. Here, we analyse the complexity of our implementation showing how it depends on the structure of the dictionary.

Algorithm 4.1 Inner products calculation from [83].

```
for i = 1 to b do
  Calculate vertical inner products V_i = ⟨Rf_n, g_i⟩
  for j = 1 to b do
    Calculate horizontal inner products ⟨Rf_n, g_i ⊗ g_j⟩ = ⟨V_i, g_j⟩
  end for
end for
```

Denote the length of the filter g_i by w_i for $i = 1, \ldots, b-1$. Following [83], if separability is ignored then calculation of all inner products for an image of size N by M would require $T_{inner}^{(non-sep)}$ multiply-accumulate operations:

$$T_{inner}^{(non-sep)} = NM \left(\sum_{i=1}^{b} w_i \right)^2 . \tag{4.13}$$

By exploiting separability in the same way as in [83] (see Algorithm 4.1), the number of operations can be reduced to:

$$T_{inner}^{(sep)} = NM\,(b+1) \sum_{i=1}^{b} w_i. \tag{4.14}$$

The overall complexity of the proposed implementation of MP can be estimated as follows:

$$T^{(sep)} = T_{in}^{(sep)} + \sum_{n=1}^{K} \left(T_{update_n}^{(sep)} + T_{search_n}^{(sep)} \right). \tag{4.15}$$

Initialisation and update steps involve mainly multiply-accumulate operations performed according to Algorithm 4.1. The search for the maximum value is performed over the whole image at initialisation and only over the modified subbands at each subsequent iteration.

$$T_{search_n}^{(sep)} = O(NM). \tag{4.16}$$

Update of the maximum inner product is performed only on the area where it could change. This area has size $W_n \times H_n$ with $W_n = H_n = 2(W-1)$, where $W = \max_i (w_i)$ is the length of the longest generator. By combining Equation (4.14)-(4.16) and writing $S = \sum_{i=1}^{b} w_i$ we get the following complexity estimate for the update step:

$$T_{update_n}^{(sep)} = 4(W-1)^2(b+1)S. \tag{4.17}$$

Complexity from the Equation (4.17) can be expressed, using the fact that $S \leq bW$, in terms of O-notation as:

$$T^{(sep)} = O(b^2 W^3 K) + O(KNM), \tag{4.18}$$

where $N \times M$ is size of the image (subband in our case). From both Equation (4.17) and Equation (4.18) it is clear that size of the dictionary and lengths of generators are the critical factors for the complexity of the MP algorithm. The maximal footprint has a more significant impact than the number of generators.

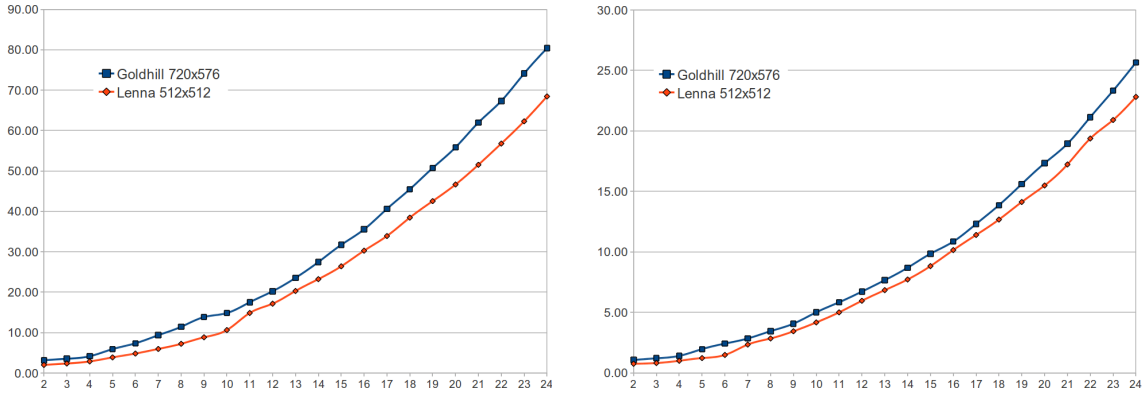


Figure 4.8: Increase in complexity during the process of adding functions picked by Basis Picking [76] to a dictionary. Colour decomposition (left), grayscale (right), time in seconds on Linux PC with Intel Core 2 Duo (y-axis) and number of dictionary generators (x-axis).

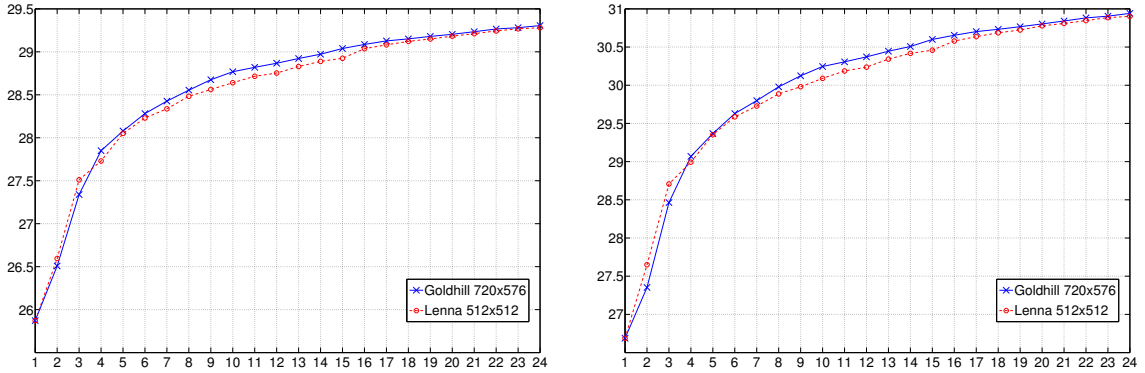


Figure 4.9: Changes in average PSNR over 10 test images during Basis Picking for 6000 atoms. Colour decomposition (left), grayscale (right), PSNR and RGB-PSNR (y-axis) and number of dictionary generators (x-axis).

4.3.2 Dictionary Structure and Computational Complexity

For MP performed in the transformed domain the details of complexity are much more complicated than Equation (4.17) since they also depend in which subband the atoms are found and other factors such as the boundary treatments. For the low-frequency subbands the inequality $b^2W^3 \gg NM$ typically holds. For example for 5-scale DWT decomposition of 512×512 image the approximation subband is only of size 16×16. For a 512×512 image we already need to search up to 65536 (256×256) values in the highest-frequency subbands. If atoms are found in the higher frequencies the computational cost of the maximum inner product search becomes more significant. Therefore the initial iterations, when more low-frequency atoms are picked, are computed slightly faster. In practice the inner-product recalculation is always the core part of computations.

Experimental results of complexity analysis of MP performed during the process of training dictionaries on different images are shown in Figure 4.8. The growth in MP decomposition time with increasing dictionary size during Basis Picking process is shown for grayscale and colour *Goldhill* and *Lenna* images. For a fixed number of atoms the size of an input image is less critical for complexity than the structure of a dictionary. On

the other hand, it has to be remembered that more atoms are usually needed for higher-resolution images to achieve the same distortion. The current C++ implementation that uses uBlas on a Linux PC with Intel Core 2 (3 GHz) finds 6000 colour atoms in the wavelet domain in around 30 seconds for the dictionary $\mathcal{D}_{16}^{(t)}$ and colour image *Goldhill* of dimension 720×576 (see Figure 4.8). For the same dictionary 6000 grayscale atoms are found in around 10 seconds. For this dictionary: $b = 16$, $S = 95$ and $W = 9$ resulting in not more than $T_{update_n}^{(sep)} = 413440$ multiply-accumulate operations per iteration (see Equation (4.17)). For $\mathcal{D}_{16}^{(b2)}$ we have: $b = 16$, $S = 56$ and $W = 5$ with $T_{update_n}^{(sep)} = 60928$, which is more than six times less. Since the time complexity of the search for the maximum value, $T_{search_n}^{(sep)}$, depends in our implementation on the subband size and not on the dictionary, in practice the change of the dictionary from $\mathcal{D}_{16}^{(t)}$ to $\mathcal{D}_{16}^{(b2)}$ gives around 3-fold speed-up. For example, finding 6000 grayscale atoms for *Goldhill* image takes around 3 seconds with $\mathcal{D}_{16}^{(b2)}$ comparing to 10 seconds for $\mathcal{D}_{16}^{(t)}$.

At this point it is also worth to ask the question of the optimal number of bases in the dictionary as there is a trade-off between time complexity and distortion. Figure 4.9 shows the improvements in PSNR for the training images after picking each basis in turn. It is clear that during the first few iterations adding a new function significantly improves coding efficiency while later in the process only a minor improvement in PSNR was achieved. Our choice here is a fixed number of 16 generators. Adding more bases would improve PSNR but it has to be remembered that for image compression encoding has to be taken into account. The more generators are used the more bits are needed to encode them. Moreover for more than 16 bases the average gain in PSNR is lower than 0.1 dB. We propose and describe a quantisation and encoding method in Chapter 5 and leave its analysis in terms of dictionary size for Chapter 6.

4.4 Summary

In this chapter we analysed in depth the MP performed in the spatio-frequency. A special focus was on the choice of the transform and designing the dictionaries. We have demonstrated clearly the advantage of performing MP in the spatio-frequency domain over MP directly in the spatial domain. We have also shown that choosing DWT with smooth regular filters such as CDF 9/7 from JPEG 2000 gives much better performance than DCT or Haar transforms. The process of decomposition can be significantly speeded-up by block-based processing with 16×16 blocks however at the cost of higher distortions.

The choice of spatio-frequency transform can be viewed as the first stage of a dictionary design process. The second stage is the choice of generators (filters) to perform MP. The number of generators and their lengths are critical for the complexity of the decomposition algorithm. We trained the dictionaries using a method proposed in the literature [76] for a similar hybrid Wavelet and MP-based image codec. We proposed a dictionary with shorter support generators that can give statistically the same performance with significantly reduced complexity. A key to the success is to minimise coherence and not to limit the search for generators to Gabor atoms. We extended the Gabor candidate set with Walsh-

like bases achieving better performance in terms of PSNR. The topic of dictionaries will be reconsidered in Chapter 6 in terms of quantising and encoding MP decomposition into a bit-stream proposed in, the next, Chapter 5.

5 Quantisation and Atom Encoding

For data compression applications an MP decomposition has to be encoded into a bit-stream. Typically, in transform based lossy image coding the floating-point data obtained after the transformation are quantised and then encoded using some lossless scheme. In the case of an MP-based system quantisation can be performed inside (*in-loop*) [83] or outside (*a-posteriori*) [38] the MP loop. In-loop quantisation has the aim of correcting the introduced quantisation error during later iterations, thus achieving better R-D performance than a-posteriori schemes [23]. With this approach, unlike in the case of wavelet or DCT-based coders, quantisation becomes an integral part of the transformation step.

Quantisation is a subject of Section 5.1. Sufficient conditions for convergence are given and their practical implications are analysed. Results are extended into the multichannel case. Then, Section 5.2 introduces and presents the details of the encoding stage. The proposed algorithm, inspired by advances in representing file and database indexes, is a novel method for encoding MP decomposition into a bit-stream. The idea to adaptively mix run length with entropy coding for the case of sorted and grouped symbols is the main contribution of this chapter. The coding of MP decomposition has been extensively studied in the literature focusing on a single-channel data [1, 36, 78, 83]. In colour image coding with simultaneous decomposition of all the channels, new problems arise with the new types of data. A detailed description of the proposed solutions is the main subject of Section 5.2.

Most of the information included in this chapter is available in the report available from [69]. The ideas for encoding have been presented in [68].

5.1 Quantisation

5.1.1 Quantised Matching Pursuit

The values $a_n = \langle R^n f, g_{\gamma_n} \rangle$ obtained by MP are quantised (e. g. rounded) to values A_n that can be mapped into the symbols from a finite alphabet and encoded. This step, when performed in-loop, modifies the update steps from the Algorithms 3.1 and 3.3 introduced in Section 3.1. MP with in-loop quantisation is referred to as Quantised Matching Pursuit (QMP) [46]. Below, we give the necessary conditions which have to be met for the single-channel QMP to converge. The conclusions are then generalised for multichannel QMP.

For MP without quantisation and decomposition given by Equation (3.9) the Parseval-like equality (see Equation (3.11)) is satisfied. If we replace a_n by A_n in the update step to reflect in-loop quantisation then Equation (3.11) will change into [84]:

$$||f||^2 = \sum_{n=1}^{N} \left(|a_n|^2 - |A_n - a_n|^2 \right) + ||R^{N+1} f||^2. \tag{5.1}$$

To preserve convergence of the algorithm the energy of the residual $R^n f$ has to keep decreasing [72, 84]. Therefore we only can use the quantisation methods for which:

$$|a_n|^2 - |A_n - a_n|^2 > 0. \tag{5.2}$$

Applying polarisation inequality, it follows that: $|A_n|^2 < 2\Re\langle A_n, a_n \rangle$. When A_n and a_n are real numbers Equation (5.2) implies $A_n(2a_n - A_n) > 0$ which is equivalent to a_n, A_n having the same sign and their absolute values to satisfy [84]:

$$0 < |A_n| < 2|a_n|. \tag{5.3}$$

An issue with a too coarse quantisation is the possibility of a *dead-lock* occurring which, for example, would be present if we allowed quantisation to 0 [82, 84]. Therefore it is important to analyse the effect of quantisation on the decomposition algorithm. The condition from Equation (5.2) does not allow quantisation to 0. It is a necessary condition for MP with quantisation to converge. Nevertheless, only the decreasing norm of the residuals $R^n f$ is guaranteed and with a too coarse quantisation the algorithm still could converge to a non-zero residual. Proposition 5.1.1 gives a stronger but sufficient condition under which it can be proved analogously to [117], where MP is referred to as the Weak Greedy Algorithm, that the convergence is guaranteed.

Proposition 5.1.1. *For MP with in-loop quantisation to converge it is enough that the quantisation error introduced at the nth iteration is lower than than some fixed fraction* $\theta \in (0, 1)$ *of the actual amplitude* $|a_n|$*:*

$$|\epsilon_n| = |A_n - a_n| \leq \theta |a_n|. \tag{5.4}$$

Proof. To prove Proposition 5.1.1 we follow the proof of Theorem 1 from [117]. We shall present the proof in full details in Appendix B. Equation (5.1) guarantees that $||R^N f||$ is strictly decreasing, hence $||R^N f||$ converges and it can be proven (see Lemma 2.2

from [117] and Appendix B) that also $R^N f \to R^\infty$ as $N \to \infty$. It is enough to show that $||R^N f|| \to 0$ which is equivalent to $R^N f \to \mathbf{0}$. Suppose this is not true and $R^\infty \neq \mathbf{0}$. Hence, there exist $\delta > 0$ such that $\sup_{g_\gamma \in \mathcal{D}} |\langle R^\infty, g_\gamma \rangle| \geq 2\delta$. This implies that there exist M such that for all $N > M$ we have $\sup_{g_\gamma \in \mathcal{D}} |\langle R^N f, g_\gamma \rangle| \geq \delta$. For all N we have: $|a_N| = |\langle R^N f, g_{\gamma_N} \rangle| \geq \alpha \sup_{\gamma \in \mathcal{D}} |\langle R^N f, g_\gamma \rangle| \geq \alpha\delta$, and hence:

$$||R^{N+1} f||^2 = ||f||^2 - \sum_{n=1}^{N} \left(|a_n|^2 - |\epsilon|^2\right) \leq$$
$$||f||^2 - (1-\theta^2) \sum_{n=1}^{N} |a_n|^2 \leq ||f||^2 - N(1-\theta^2)\alpha^2\delta^2,$$

which implies that:

$$||f||^2 - ||R^{N+1} f||^2 \geq N(1-\theta^2)\alpha^2\delta^2 \geq 0,$$

which is impossible as the term $N(1-\theta^2)\alpha^2\delta^2 \to \infty$ as $N \to \infty$ while $||f||^2$ and $||R^{N+1} f||^2$ are bounded. □

The inequality in Equation (5.2) can be derived as follows from Equation (5.4):

$$0 < |A_n| \leq |a_n| + |A_n - a_n| \leq (1+\theta)|a_n| < 2|a_n|. \tag{5.5}$$

Moreover, we have:

$$|A_n| \geq |a_n| - |A_n - a_n| \geq (1-\theta)|a_n| > 0. \tag{5.6}$$

For real numbers the condition from Equation (5.4) becomes:

$$0 < (1-\theta)|a_n| \leq |A_n| \leq (1+\theta)|a_n| < 2|a_n|. \tag{5.7}$$

Proposition 5.1.1 provides certainty that every new atom will sufficiently refine a decomposition to lead to convergence. Parameter θ can be viewed as an analogy to the sub-optimality parameter α. Letting $\alpha < 1$ we allow selecting a non-optimal atom while $\theta > 0$ means non-optimal choice of the amplitude.

To satisfy Proposition 5.1.1 the quantisation scheme must involve adaptivity as the bounds in Equation (5.4) depend on the actual value a_n of the inner product at the nth iteration. The methods used for MP-based scalable image coding including those used in grayscale video [84,130] and image [36,38,107,131] coding all apply adaptive schemes in this sense. Our grayscale implementation utilises Precision Limit Quantisation (PLQ) [77,131] while the colour codec uses PLQ and Uniform Quantisation (see Section 5.1.3). These quantisation schemes conform to requirements of Proposition 5.1.1. In the next sections we evaluate their performance in grayscale and colour image coding applications. Encouragingly, the experiments show that for a single-channel codec and PLQ quantisation, the distortion is only slightly higher than without quantisation for the same number of atoms. This confirms the advantage of in-loop quantisation and shows that even very coarse quantisation can give satisfactory results in imaging applications. It serves as an important guidance for designing the atom encoder.

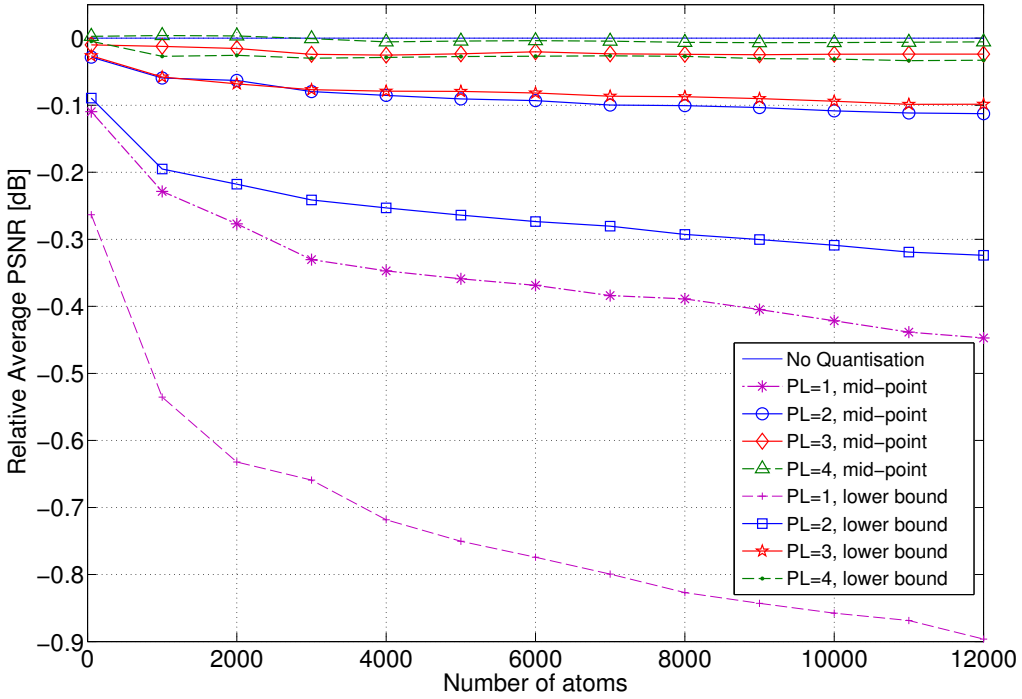


Figure 5.1: Differences in PSNR for varying quantisation parameters at a given number of atoms averaged over 12 grayscale test images relative to the MP without quantisation (5 scales, Dictionary $\mathcal{D}_{16}$).

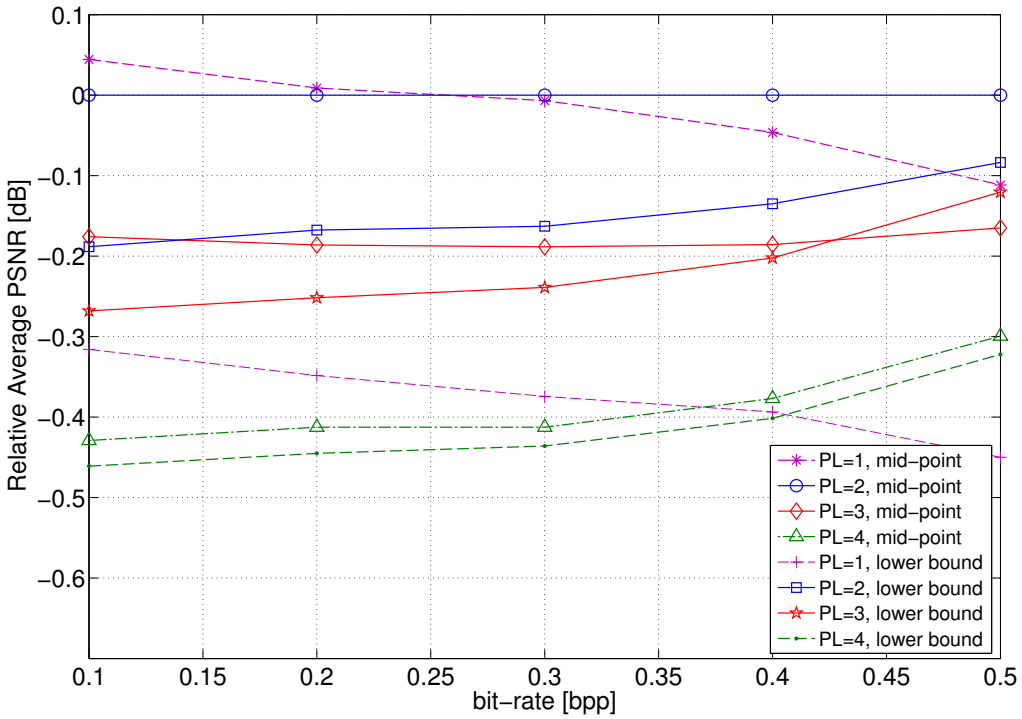


Figure 5.2: Differences in PSNR for varying quantisation parameter at a given bit-rates averaged over 12 grayscale test images relative to: $PL = 2$ with PLQ to the mid-point (5 scales, Dictionary $\mathcal{D}_{16}$).

5.1.2 PLQ Quantisation

The original idea of PLQ comes from bit-plane coding where only the most significant bits are encoded for each coefficient (here each atom amplitude a_n) [77]. Quantisation bins in PLQ with parameter PL ($PL > 0$) are of the form:

$$\begin{aligned} \text{bit-planes:} \quad & k = M, M-1, \ldots, 1, 0, -1, \ldots, \\ \text{refinements:} \quad & r = 1, 1 + \tfrac{1}{2^{PL-1}}, \ldots, 1 + \tfrac{2^{PL-1}-1}{2^{PL-1}}, \\ \text{quantisation bins:} \quad & B_{kr} = \left[r2^k, (r + \tfrac{1}{2^{PL-1}})2^k \right). \end{aligned} \tag{5.8}$$

The original approach was to keep only the most significant bit and some refinement bits governed by PL which means that the a_n was quantised to $|A_n| = r2^k$ [77]. The integer value k indicates the bit-plane and the positive value r is called a refinement. However, it is known from quantisation theory [41] that the quantisation to the middle point of the bin is a better choice (in fact, optimal in the mean-squared error sense when data are uniformly distributed over the bins). The experiments with PLQ (see Figure 5.1) clearly confirm superiority of a mid-point quantisation only when $PL \leq 2$. The advantage known in the theory is non-obvious due to the non-linear character of both MP algorithm and in-loop quantisation.

If we select mid-point quantisation then the value $|a_n| \in B_{kr}$ is quantised to: $|A_n| = (r + 2^{-PL})2^k$. Such a scheme satisfies the conditions from Equation (5.4). We have $|a_n| \in B_{kr}$ and further:

$$\begin{aligned} \tfrac{|a_n|}{2} \leq r2^k \leq |A_n|, \\ |A_n - a_n| \leq \tfrac{1}{2^{PL}}2^k \leq \tfrac{1}{2^{PL}}|a_n|. \end{aligned} \tag{5.9}$$

Figure 5.1 shows that for a single-channel decomposition the level of distortion introduced by quantisation to the lower bound $r2^k$ with parameter $PL = p$ gives the similar results to quantisation to the mid-point with $PL = p-1$, i. e. one level coarser. The results from Figure 5.1 are in agreement with the bounds described in Equation (5.9). For quantisation to the lower bound these bounds could be violated when $PL = 1$. We can also see in Figure 5.1 that for $PL = 1$ quantisation to the lower bound gives by far the highest distortion. The benefit of mid-point quantisation becomes less obvious for $PL = 3$ and gives even slightly worse coding performance for $PL = 4$ (Figure 5.2).

The use of PLQ allows us to group the atoms with the same value $|A_n|$ and only signalling the group counts or end of groups. Moreover the atoms inside one group can be rearranged providing a potential for additional compression.

Experiments show (see Figure 5.2) that for our coding scheme the optimal value of the quantisation parameter is $PL = 2$ as advised in [131]. This is not surprising as our coding scheme for grayscale utilises the same concept of grouping atoms as the MERGE coder [78] used in [131]. Figure 5.3 presents the PLQ quantisation for $PL = 2$. In our case, $PL = 2$ and quantisation to the mid-point implies that the refinement $r \in \{1.25, 1.75\}$. For example the value $|a_n| = 13.5$, which belongs to the bit-plane with $k = 3$, will be quantised to 14 which is a mid-point between 12 and 16 ($12 = 1.50 \cdot 2^3$, $14 = 1.75 \cdot 2^3$ and $16 = 2^4$ respectively).

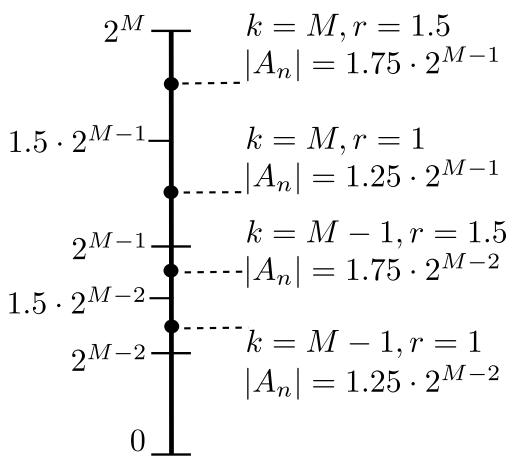


Figure 5.3: Precision Limit Quantisation with parameter $PL = 2$ (bit-planes M and $M-1$ shown).

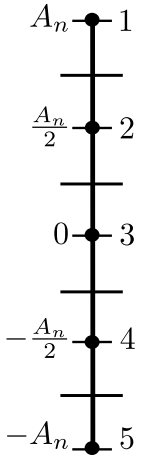


Figure 5.4: Scalar Uniform Quantisation with parameter $L = 2$ (5 quantisation bins indexed from 1 to 5).

Interestingly, as visible in Figure 5.1, for a wide range of parameters there is only a minor difference in PSNR between MP with PLQ and without any quantisation. For quantisation to mid-point and $PL = 3$ the average PSNR over 12 test images is lower than for decomposition without quantisation by less than 0.05 dB with up to 12000 atoms. The case of 12000 atoms corresponds to bit-rates higher that 0.5 bpp for all tested images. The gap in PSNR between MP without and with quantization increases with the number of atoms. Nevertheless, for $PL = 4$ the effect of quantisation is practically negligible at all tested rates and numbers of atoms. This confirms usefulness of PLQ for in-loop quantisation of the MP decomposition.

Since, in the end the coding performance has to be taken into account, the preferable value for PL is $PL = 2$ rather than $PL = 4$ (see Figure 5.2). $PL = 1$ appears to be effective only for very low bit rates. Smaller values of PL mean grouping atoms into fewer groups with a larger number of atoms. Larger groups allow more benefit to be gained from Run Length Encoding at the encoding stage, which is described in detail in Section 5.2. The size of groups increases with decreasing amplitudes which corresponds to the increasing number of atoms. This explains the significant improvement in relative performance with increasing bit-rate for the higher values of parameter PL ($PL \geq 3$). Nonetheless, for the bit-rates up to 0.5 bpp the choice of $PL = 2$ is on average at least 0.1 dB superior over $PL = 3$ and even 0.3 dB over $PL = 4$.

5.1.3 Colour Amplitude Quantisation

The nature of multi-channel MP applied to decomposition of RGB images is far more complicated than single-channel MP for grayscale. This section proposes an extension of the quantisation scheme based on PLQ and grouping similar atoms into colour coding. Similarly as for grayscale we start with analysis of the convergence bounds. Proposition 5.1.1 can be easily adapted for multichannel signals by the following:

Proposition 5.1.2. *For multichannel MP with in-loop quantisation to converge it is enough that the maximal amplitude over all channels is quantised so that the condition* (5.4) *from Proposition 5.1.1 is satisfied and the norms of channel residuals are not increasing.*

Proof. We adapt the proof of Proposition 5.1.1 to M-channel signals from the Hilbert space $\mathcal{H}^M$ i. e. M-dimensional vectors of the form $f = [f^{(1)}, \ldots, f^{(M)}]$. The convergence means here that for any dictionary that spans a space $\mathcal{H}$ the residual $R^N f^{(i)}$ converges to 0 as $N \to \infty$ for each component $f^{(i)}$, where $i \in \{1, \ldots, M\}$. Non-increasing channel norm means that for each channel only a relaxed condition (i. e. with weak inequalities) from Equation (5.2) needs to be satisfied, i. e.:

$$\forall_{i \in \{1,\ldots,M\}} |a_N^{(i)}|^2 - |A_N^{(i)} - a_N^{(i)}|^2 \geq 0, \tag{5.10}$$

requiring a stricter condition $|a_N^{max}|^2 - |A_N^{max} - a_N^{max}|^2 > 0$ only for a maximal amplitude:

$$|a_N^{max}| = \max_{i \in \{1,\ldots,M\}} |\langle R^N f^{(i)}, g_{\gamma_N} \rangle|, \tag{5.11}$$

which guarantees that for at least one component the norm $||R^N f^{(max)}||$ is strictly decreasing. Similarly as for Proposition 5.1.1, it is enough to prove, this time for each component $i \in \{1, \ldots, M\}$ that $||R^N f^{(i)}|| \to 0$ as $N \to \infty$. Suppose for a contradiction, like in the proof of Proposition 5.1.1, that there exists $\delta > 0$ such as:

$$\max_{i \in \{1,\ldots,M\}} \sup_{g_\gamma \in \mathcal{D}} |\langle R^N f^{(i)}, g_\gamma \rangle| \geq \delta. \tag{5.12}$$

The Parseval equality (Equation (3.11)) is satisfied for each channel $i \in \{1, \ldots, M\}$. We simply sum it up over the all M channels:

$$\sum_{i=1}^{M} ||R^{N+1} f^{(i)}||^2 = \sum_{i=1}^{M} ||f^{(i)}||^2 - \sum_{i=1}^{M} \sum_{n=1}^{N} \left(|a_n^{(i)}|^2 - |\epsilon_n^{(i)}|^2 \right). \tag{5.13}$$

By extracting the maximal amplitude components from the sum on the right-hand side and skipping the rest of the terms as non-negative numbers (due to the weak inequality from Equation (5.10)), we have:

$$\sum_{i=1}^{M} ||R^{N+1} f^{(i)}||^2 \leq \sum_{i=1}^{M} ||f^{(i)}||^2 - \sum_{n=1}^{N} \left(|a_n^{(max)}|^2 - |\epsilon_n^{(max)}|^2 \right). \tag{5.14}$$

Which leads to the same type of impossible inequality as for Proposition 5.1.1:

$$\sum_{i=1}^{M} ||R^{N+1} f^{(i)}||^2 \leq \sum_{i=1}^{M} ||f^{(i)}||^2 + N(1 - \theta^2)\alpha^2 \delta^2, \tag{5.15}$$

and finishes the proof. □

This proof is adapted from [117] and [118] where different criterion for atom selection from Algorithm 3.3 has been used. It can be easily shown (see (5.16) below) that the criterion of

picking the atom that maximises L_2-norm used in our codec satisfies inequality from [118] by taking a sub-optimality factor $\frac{\alpha}{3}$.

$$\begin{gathered}\max_{i\in\{r,g,b\}} |\langle R^n f^{(i)}, g_{\gamma_n}\rangle| \geq \\ \tfrac{1}{3}\sqrt{|\langle R^n f^{(r)}, g_{\gamma_n}\rangle|^2 + |\langle R^n f^{(g)}, g_{\gamma_n}\rangle|^2 + |\langle R^n f^{(b)}, g_{\gamma_n}\rangle|^2} \geq \\ \tfrac{\alpha}{3}\sqrt{|\langle R^n f^{(r)}, g_{\gamma}\rangle|^2 + |\langle R^n f^{(g)}, g_{\gamma}\rangle|^2 + |\langle R^n f^{(b)}, g_{\gamma}\rangle|^2} \geq \\ \tfrac{\alpha}{3}|\langle R^n f^{(max)}, g_{\gamma}\rangle|.\end{gathered} \tag{5.16}$$

In the quantisation scheme proposed in this work, the amplitude with the maximal value over the three colour channels (a_n^{max}) is quantised using PLQ and serves as a base for grouping atoms. The atoms with the same quantised absolute value of maximal amplitude $|A_n^{max}|$ compose one group. We record the channel c_n for which the maximal value occurred. The remaining two amplitudes for the other two colours are quantised using uniform scalar quantisation with dead-zone subtraction [41]. The values sent to the encoder are $d_n^{(i)}$ (given for $i = 1, 2$ by Equation (5.17)).

$$\begin{aligned} Q(A_n^{(i)}) &= \min\left(\text{round}\left(L\tfrac{|a_n^{(i)}|}{|A_n^{max}|}\right), L\right), \\ d_n^{(i)} &= L + 1 - \text{sgn}(a_n^{max})\,\text{sgn}(a_n^{(i)})Q(A_n^{(i)}). \end{aligned} \tag{5.17}$$

The values $a_n^{(i)}$ are quantised in a range determined by $|A_n^{max}|$ which is available for both the encoder and decoder. This means introducing the *overload error* [41] when $|a_n^{(i)}| > |A_n^{max}|$. Parameter L determines a number of quantisation bins and the *granularity* of quantisation. The values $d^{(i)}$ given by Equation 5.17 are sent to the encoder as additional atom parameters. The signs of the amplitudes are included into these values which is visualised in Figure 5.4. $Q(A_n^{(i)})$ would be used instead if signs were considered separately. Below we give an example to clarify the quantisation scheme defined by Equation (5.17).

Example of quantisation

Suppose we want to encode the nth atom with the amplitudes for R, G, B channels: $a_n^{(r)} = 11$, $a_n^{(g)} = 13.5$ and $a_n^{(b)} = -9$ respectively. If the parameters are $PL = 2$ and $L = 2$, then $a_n^{max} = a_n^{(g)}$ is quantised to $A^{max} = 14$ using the PLQ quantiser. Channel index, $c_n = 2$ and the sign of the maximum amplitude, $\text{sgn}(a_n^{max}) = +1$ are recorded as a side information. The assumed range of quantised values is $[-14, 14]$. If there is a value outside the range then it is simply truncated as given by Equation (5.17). There are 5 quantisation bins: $[-14, -10.5)$, $[-10.5, -3.5)$, $[-3.5, 3.5)$, $[3.5, 10.5)$ and $[10.5, 14]$ (see Figure 5.3). The value $a_n^{(1)} = a_n^{(r)}$, since it is greater than 10.5, is quantised to $A_n^{(1)} = 14$ and mapped into the first bin ($d_n^{(1)} = 1$) while $a_n^{(2)} = a_n^{(b)} \in (-10.5, -7]$ is quantised into $A_n^{(2)} = -9$ and mapped into $d_n^{(2)} = 4$.

5.1.4 Choice of Quantisation Parameters

In this Section the effects of colour quantisation granularity are studied. It has been observed that the three channel amplitudes after MMP performed in RGB are highly

image	$\mathrm{corr}(a^{(r)}, a^{(g)})$	$\mathrm{corr}(a^{(r)}, a^{(b)})$	$\mathrm{corr}(a^{(g)}, a^{(b)})$
airplane512x512	0.8428	0.7513	0.9049
baboon512x512	0.3933	0.1004	0.7805
barbara720x576	0.8261	0.6959	0.8875
goldhill720x576	0.9181	0.8780	0.9406
house768x512	0.9632	0.9341	0.9519
lena512x512	0.8505	0.5454	0.7734
lighthouse768x512	0.8200	0.5450	0.8654
motorcross768x512	0.8692	0.7690	0.8422
parrots768x512	0.5835	0.4666	0.6409
peppers512x512	0.2286	0.3936	0.8047
sailboat512x512	0.8774	0.7826	0.9299
sailboats512x768	0.9306	0.6475	0.7253

Table 5.1: Correlations between channel amplitudes for the decompositions of 12000 atoms obtained with the parameters: $PL = 2$, $L = 2$, $S = 5$.

correlated. Table 5.1 contains the cross-correlations between pairs of channels for quantised amplitudes for a set of test images, which are usually high and always positive. The presence of such high correlations can be expected but due to non-linearity of both quantisation and MP, it is not obvious. Moreover, for more than 72 percent of the atoms, the signs of all three amplitudes are the same. In [37], in order to exploit these high correlations between atom amplitudes, the amplitudes were quantised after transforming into HSV colour space.

In our coding scheme, based on in-loop PLQ quantisation and encoding the groups with the same maximal amplitude, we try to exploit these correlations at the coding stage. A low value for the optimal uniform quantisation parameter L means longer runs of the same amplitude differences which can be exploited during encoding stage using Run Length Encoding. The value of L has been experimentally chosen to be as low as $L = 2$ in order to maximally reduce the number of bits required (see Figure 5.5). It can be seen in Figure 5.5 that for $L = 1$ the quantisation error increases significantly comparing to $L = 2$. The value of the parameter $PL = 2$ for PLQ quantisation appeared to be optimal as it is for grayscale. However, contrarily to grayscale where mid-point quantisation is preferred, there is a significant gap between a mid-point and lower bound quantisation in favour of the lower bound. The difference grows from 0.1 dB for 50 atoms to almost 0.2 dB for 12000 atoms. PLQ Quantisation to a lower bound of the bin results in a lower absolute value of A_n^{max} and finer uniform quantisation of the other two channels, leading to the possibility of lower overall error. When using finer uniform quantisation the choice of a mid-point becomes more attractive. For fixed PLQ parameter at $PL = 2$, as in Figure 5.5, mid-point is clearly preferable for $L > 2$. A general tendency is that for lower PL the advantage of choosing mid-point starts from lower L. Figure 5.6 shows the performance loss to MP without quantisation for different values of PL for fixed $L = 2$. For $PL = 1$

quantisation to a mid-point gives clearly a lower distortion while for $PL = 3$ it is clearly higher.

PLQ affects directly only one out of the three amplitudes but also determines the range and granularity of uniform quantisation. In [130], where the idea of Multichannel MP (called there Replicated MP) was used for coding Group of Frames of grayscale video, PLQ quantisation was used for all the channels. It can be seen in Figure 5.9, though, that the error introduced by PLQ is higher than for the proposed uniform scheme. The presented comparison is done for the parameters selected as optimal for video coding in [130], i. e. $PL = 2$ for a maximum amplitude and $PL = 1$ for the other two channels. Uniform quantisation gives better results even if no threshold is used for PLQ but the difference is then within 0.05 dB. However, in order to be able to encode the amplitudes as differences between bit-planes as in [130], some threshold (i. e. dead-zone) for PLQ must be introduced. This results in the clear inferiority of the PLQ from [130] to the proposed uniform scheme. A comparison between the two methods for the same number of quantisation bins is shown Figure 5.9. It is worth mentioning here that in the PLQ scheme [130] the quantisation to the mid-point of the quantisation bin is clearly superior.

Figures 5.7 and 5.8 show the effect of quantisation on the actual coding performance. Finer quantisation requires more bits. Therefore the values of PL and L have to be kept as low as possible. In Figures 5.7 and 5.8 the optimal configuration of parameters serves as a reference for comparisons. The optimal choice is $PL = 2$ and $L = 2$ with a quantisation to the middle point rather than lower bound (with the difference less than 0.05 dB though).

In Figure 5.6 the gaps between quantisation with optimal in R-D sense parameters ($PL = 2$ and $L = 2$) and MP without quantisation can be observed. The difference grows from 0.1 dB to 0.4 dB. This suggests that there is still a potential for improvement over the proposed colour quantisation scheme. On the contrary for the single-channel MP, PLQ quantisation with parameter $PL = 2$ or higher can achieve a distortion close to the case when no quantisation is performed.

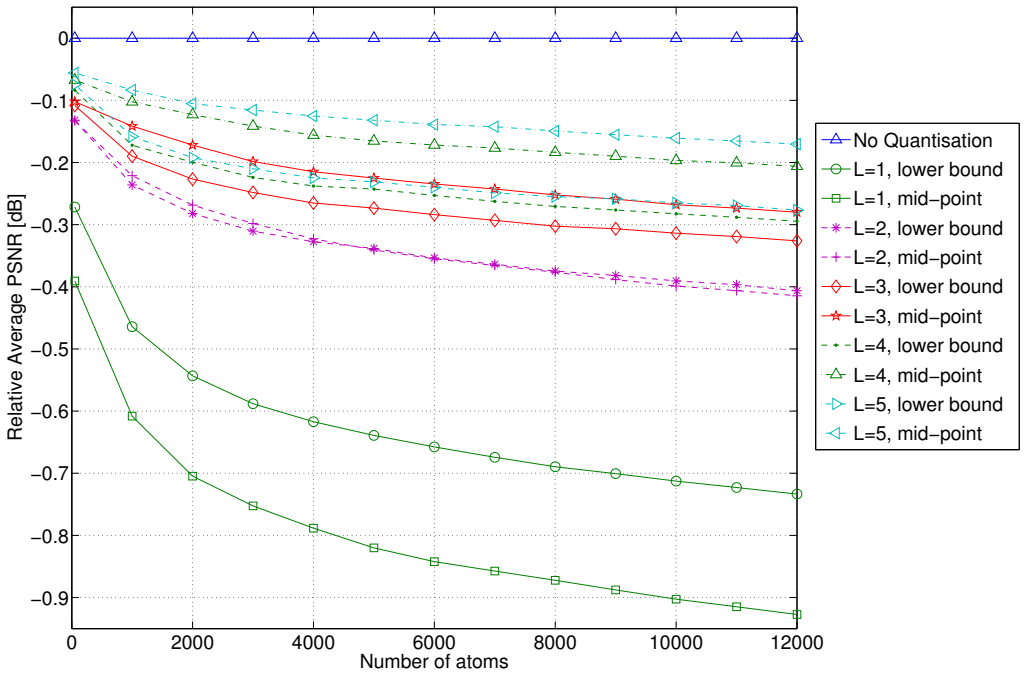


Figure 5.5: Differences in PSNR relative to Multichannel MP without quantisation averaged over 12 test images for different granularity of Uniform Quantiser ($PL = 2$, 5 scales, Dictionary $\mathcal{D}_{16}$).

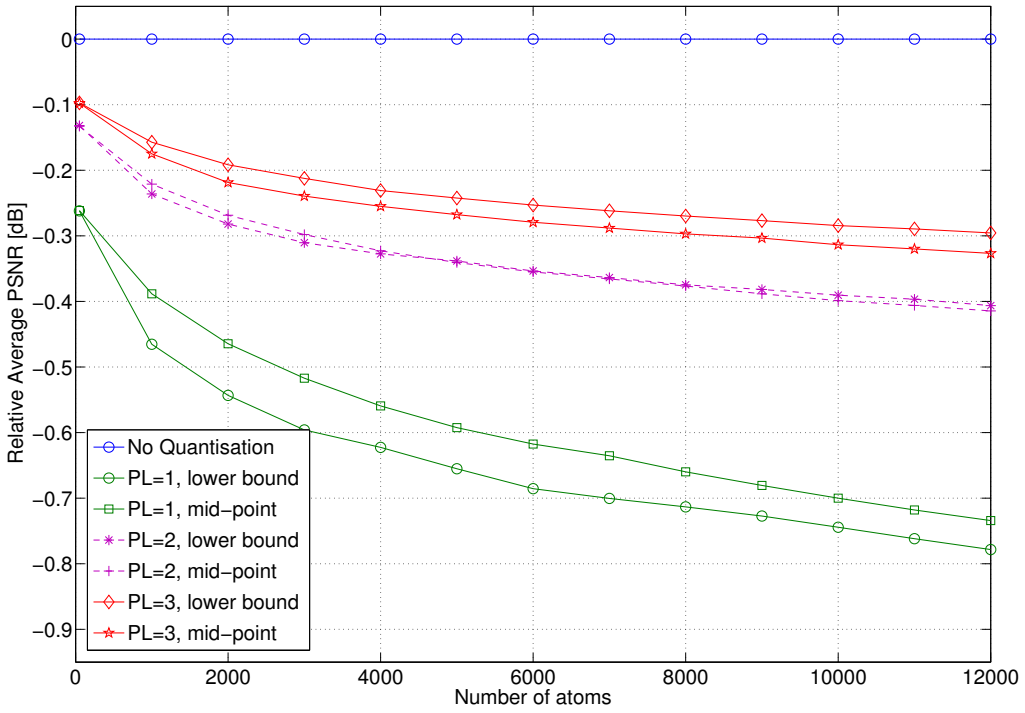


Figure 5.6: Differences in PSNR relative to Multichannel MP without quantisation averaged over 12 test images for different values of PL parameter ($L = 2$, 5 scales, Dictionary $\mathcal{D}_{16}$).

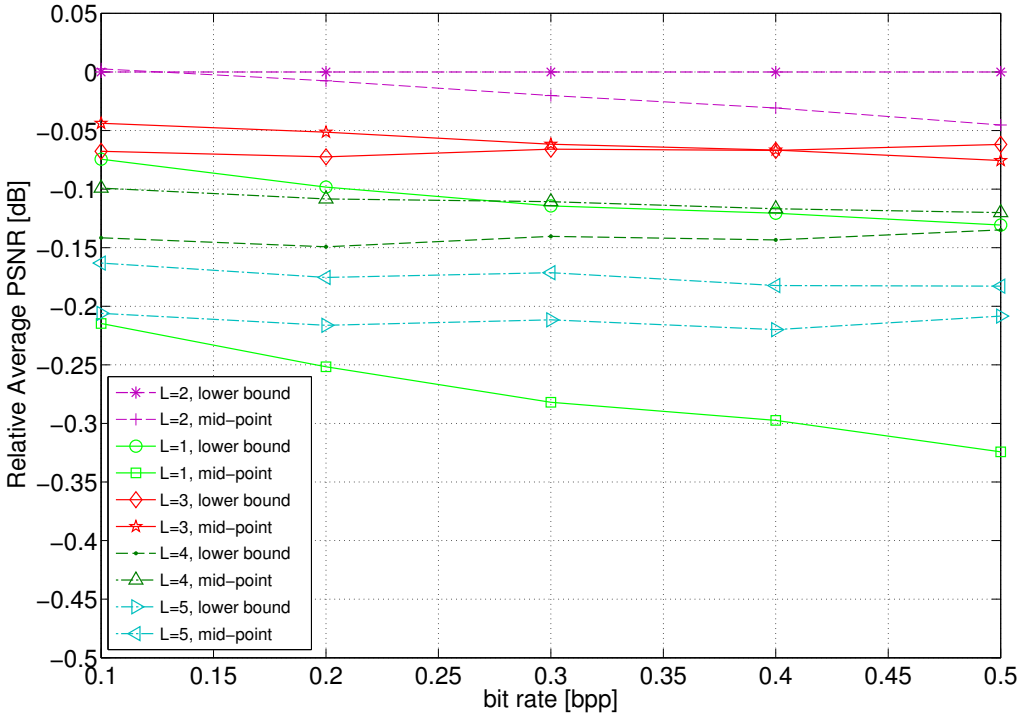


Figure 5.7: Differences in PSNR averaged over 12 test images for different granularity of Uniform Quantiser relative to: $L = 2$ and $PL = 2$ with PLQ to the lower-bound ($PL = 2$, 5 scales, Dictionary $\mathcal{D}_{16}$).

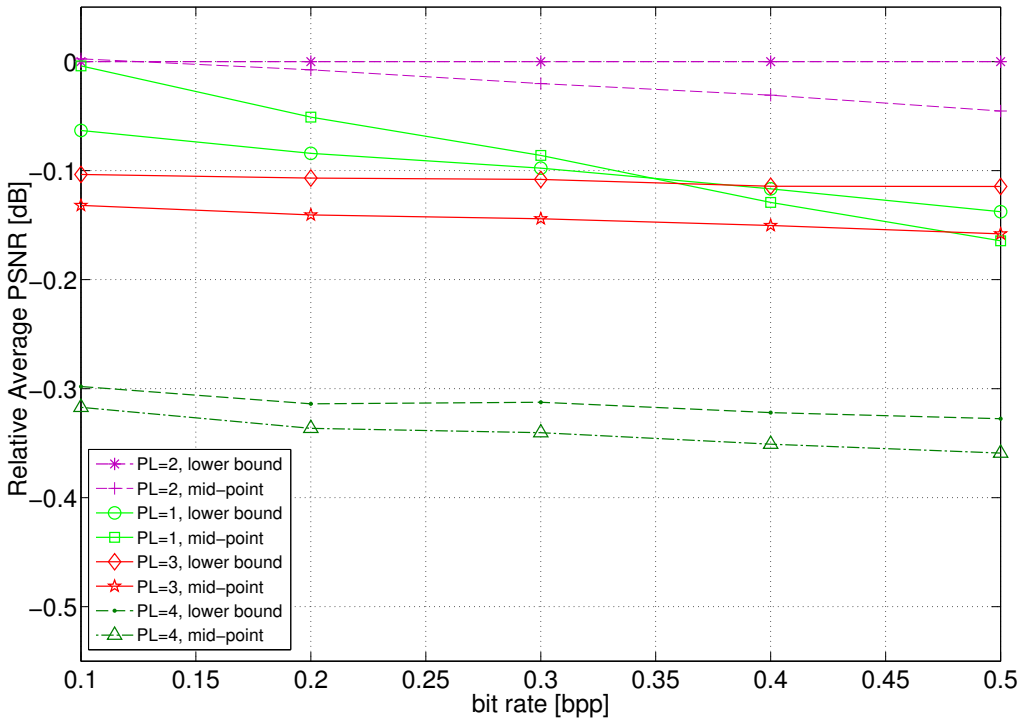


Figure 5.8: Differences in PSNR averaged over 12 test images for different values of PL parameter relative to: $L = 2$ and $PL = 2$ with PLQ to the lower-bound ($L = 2$, 5 scales, Dictionary $\mathcal{D}_{16}$).

5.2 Atom Encoding

5.2.1 Details of the Coding Algorithm

After MP decomposition and quantisation described in Section 5.1 there are groups of atoms with the same maximal amplitude to be encoded. Each group can be seen represented as a matrix M with the rows indicating atoms and columns their attributes. It is assumed that the size of each group of atoms is known: in practice group counts are encoded as a side information. The matrix of atoms within one group can be viewed as the analogy of a table from relational database.

For a table with c columns each row is often referred to as a c-tuple. The data come from finite domains (alphabets) and can always be mapped onto integer values from 1 to N, where N is the size of alphabet. We denote the alphabet size for the c-th column by N_c and call it the *column cardinality* [66]. For our colour coding, 8 columns can be distinguished, describing the following attributes of the atoms:

$1:$	s_n	sign of the maximal amplitude,	$s_n \in \{-1, 1\}$,
$2-3:$	d_n^1, d_n^2	quantised amplitude differences,	$d_n^* \in \{1, 2, \ldots, 2L+1\}$,
$4:$	c_n	maximum amplitude colour channel,	$c_n \in \{1, 2, 3\}$,
$5:$	w_n	sub-band index,	$w_n \in \{1, 2, \ldots, 3S+1\}$,
$6:$	λ_n,	2D dictionary entry,	$\lambda_n \in \{1, 2, \ldots, B^2\}$,
$7:$	x_n	atom location inside the sub-band w_n,	$x_n \in \{1, \ldots, W_{x_n}\}$,
$8:$	y_n	atom location inside the sub-band w_n,	$y_n \in \{1, \ldots, W_{y_n}\}$.

In the grayscale case, there are only 5 columns: s_n,w_n,λ_n,x_n and y_n. Our goal is to encode such a table into a bit-stream so that the maximal compression ratio is achieved. The atoms from one group have the same amplitude, hence can be considered equivalent without significantly affecting the scalability of the decomposition. The possibility of reordering rows gives a lot of flexibility in designing a coding algorithm to exploit redundancies among atoms within one group.

A choice of the optimal row ordering for maximal compression is an NP-complex problem, infeasible to solve during encoding. In practice sorting rows can be a good heuristic [66]. Moreover, there are efficient encoding techniques for sorted data known from index compression [125, ch.3]. It is natural to consider here column-oriented indexes and sort the rows in a *lexicographical order* recommended for such indexes in databases [66].

Algorithm 5.1 Encoding one group of atoms in the MP decomposition.

function encode(M,d,i_{start},i_{end})
inputs:
Matrix M of data rows.
depth parameter d.
the first i_{start} and the last row i_{end}.
body:
encode d-th column using Algorithm 5.2.
if $d < MAX$ **then**
Group the same symbols into groups g_i.
for all Groups g_i **do**
encode(M,$d+1$,i^g_{start},i^g_{end})
end for
else
encode atom positions.
end if

Algorithm 5.2 Encoding inside one column.

inputs: K length of data, N size of the alphabet.
input sequence: $\{v_s\}_{s=1,2,\ldots,K}$ with $s < s' \Rightarrow v_s \leq v_{s'}$
$s_l = K$ *symbols remaining*
$a_l = N$ *alphabet entries remaining*
while $s_l > 0$ and $a_l > 1$ **do**
if $s_l > 2a_l$ **then**
encode z_l (if any) zero lengths assuming range $0 \ldots s_l$
encode run of length R in range $0 \ldots s_l$
$s_l = s_l - R$
$a_l = a_l - 1 - z_l$
else
encode symbol v_{K-s_l+1} in range $1 \ldots a_l$
$s_l = s_l - 1$
$a_l = N - v_{K-s_l+1} + 1$
end if
end while

After grouping and sorting, the data are ready to be encoded. Algorithm 5.1 summarises one of the possible ways to scan matrix M for encoding. It calls the proposed encoding procedure outlined in Algorithm 5.2. Algorithm 5.2 is used to encode data from columns 1-6 for colours (1-3 for grayscale). Figure 5.10 presents an example of scan order while encoding the data consisting of 10 rows which represent 10 grayscale atoms.

In the lexicographical order the two rows are compared based on the first value reading left to right on which they differ. Lexicographical order is an example of *recursive order* [66]. The definition guarantees that the projection from c-column data onto $c-1$ columns generates a *recursive order*. By projection we mean taking the rows with the same values for the c-th column. This guarantees that the data recursively passed to Algorithm 5.1 are always sorted lexicographically.

Algorithm 5.2 defines the procedure to encode a sorted sequence $\{v_s\}_{s=1,2,\ldots,K}$ from an alphabet of size N_c. Data in the first few columns tend to contain a lot of consecutively

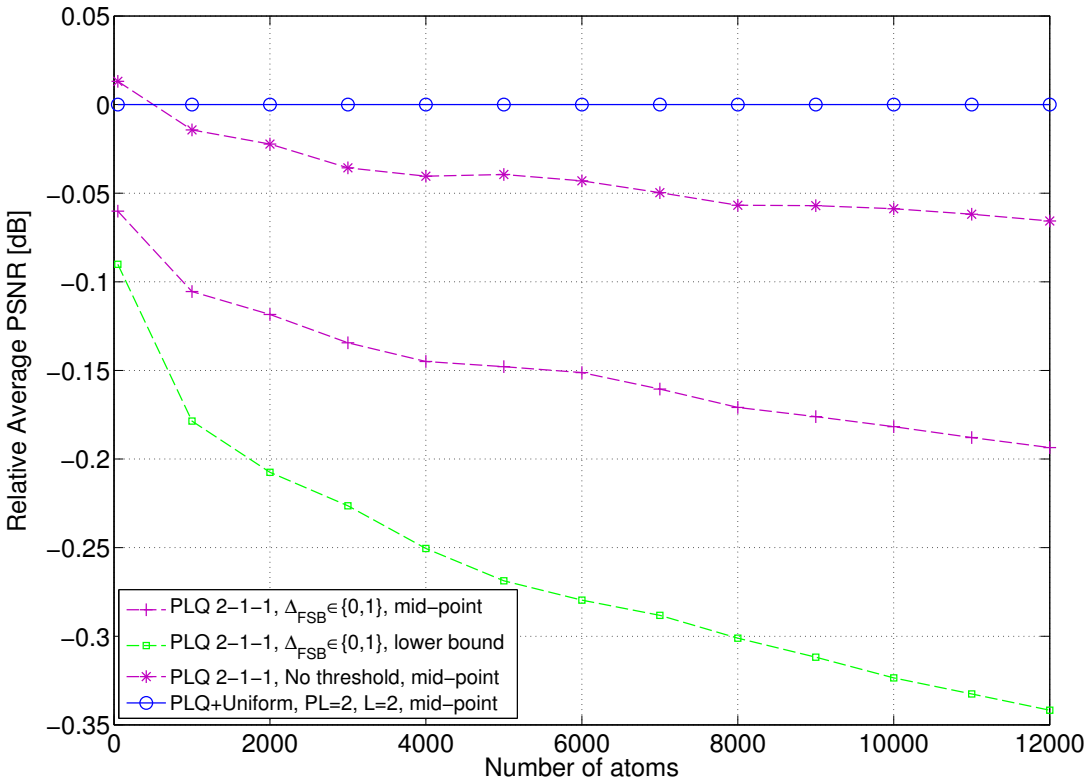


Figure 5.9: Comparison of PLQ+Uniform Quantisation against PLQ for the same numbers of quantisation bins.

$$M = \begin{array}{c} \begin{array}{ccccc} s_n & w_n & \lambda_n & x_n & y_n \end{array} \\ \left(\begin{array}{ccccc} 1 & 1 & 10 & 18 & 19 \\ 1 & 1 & 25 & 9 & 5 \\ 1 & 1 & 26 & 7 & 19 \\ 1 & 1 & 114 & 8 & 4 \\ 1 & 1 & 165 & 10 & 15 \\ 1 & 1 & 209 & 2 & 3 \\ 2 & 1 & 9 & 9 & 11 \\ 2 & 1 & 139 & 1 & 2 \\ 2 & 1 & 148 & 1 & 15 \\ 2 & 5 & 16 & 1 & 6 \end{array} \right) \end{array}$$

Figure 5.10: Example of encoding of one sorted group of coefficients.

repeating values therefore utilising Run Length Encoding (RLE) should be considered. On the other hand, deeper in the recursion of Algorithm 5.1 the consecutively repeating values are less frequent. Therefore, the atom positions (the last two columns) are considered separately and encoded as two raw values x_n and y_n that come from the ranges $1 \dots W_{x_n}$ and $1 \dots W_{y_n}$, where $W_{x_n} \times W_{y_n}$ is dimension of the sub-band w_n. For a dictionary built from 16 1D bases it rarely happens that there are more atoms with all the attributes the same. In such a conditions the most efficient way to encode position is to send a raw value.

It has to be noted that, although our coding is based on the idea of grouping and sorting atoms, it is different from MERGE coding [78] where the significance map is coded analogously to standard methods like EBCOT, EZW, SPIHT or SPECK described in Chapter 2. We use the wavelet scale index as an additional atom parameter and allow change of column order. This makes our method more flexible and, what is more important, more suitable for colour data, where there are more attributes to be encoded. The choice of column permutation that can be applied prior to sorting and encoding is analysed in Section 5.2.2.

Algorithm 5.2 summarises the proposed universal and adaptive procedure for column encoding. At each iteration of the main loop a decision is made whether to encode the s-th symbol directly or to signal its run length. The number of symbols remaining to encode after already encoding $s-1$ symbols is stored in a variable s_l. The relation of s_l to the current s-th position in the stream is: $s_l = K - s + 1$. The next symbols can come from an alphabet of size $a_l = N - v_{s-1} + 1$ ($v_s \in \{v_{s-1}, v_{s-1}+1, \dots, N\}$, i. e. v_s can take any value greater or equal to the previous symbol). The values s_l and a_l are available for both encoder and decoder at each iteration. We assume as a rule of thumb that run length coding is only efficient when there is an expected run of at least two symbols. We estimate the average expected run length as s_l/a_l, i. e. as a ratio of the number of remaining symbols to the size of alphabet. When $s_l/a_l > 2$ we encode a count of the next expected symbols which can be 0, otherwise a raw symbol v_s is encoded. The next section shows an example of an described algorithm which generates symbols that are input to a final step of arithmetic coding [126].

Example of the basic version of encoding

Consider a sorted sequence: $v = \{1, 1, 1, 1, 2, 2, 3, 4, 4\}$ from the alphabet $A = \{1, 2, 3, 4\}$ to be encoded using Algorithm 5.2. The length of the sequence is $K = 9$ which needs to be known prior to encoding/decoding. The first symbol is $v_1 = 1$. Control variables s_l and a_l are initialised to be $s_l = K = 9$ and $a_l = 4$. The ratio $s_l/a_l = 9/4 > 2$ so we start by encoding a run length of four 1s. Therefore we encode 4 and move forward by 4 symbols which implies $s_l = s_l - 4$. Now, as we encoded all 1s, we expect a symbol from the reduced alphabet $\{2, 3, 4\}$ which means that $a_l = 3$. There are also only 5 symbols left, since $s_l = 9 - 4 = 5$. The ratio $s_l/a_l = 5/3 \leq 2$ so we encode the raw symbol: 2 and move forward by one symbol ($s_l = s_l - 1$). Now there are $s_l = 4$ symbols left that still can come from the set $\{2, 3, 4\}$, i. e. $a_l = 3$. $s_l/a_l = 4/3 \leq 2$ so again we encode

the raw symbol: 2. Now there are 3 symbols left again from $\{2,3,4\}$, $a_l = 3$, $s_l = 3$, $s_l/a_l = 3/3 \leq 2$, $v_6 = 3$ comes and we encode it as a raw 3. The next symbol can only be either 3 or 4. So $s_l = 2$ symbols remaining from alphabet of size $a_l = 2$. Now, $v_7 = 4$ and $s_l/a_l < 2$ so we encode raw 4. The last symbol does not need to be encoded as we know how many symbols remained (the value of variable s_l). The input sequence v is going to be sent to the arithmetic coder as $4,2,2,3,4$. The decoder, basing on the number of remaining symbols (s_l) and the size of the alphabet (a_l), will have the same information during decoding as encoder while encoding.

5.2.2 Column Order

image	the best order	the worst order	(π_g) $(2,3,1)$	$2 \cdot \frac{worst-best}{worst+best}$
airplane512x512	101776(2,3,1)	108487(3,1,2)	101776	6.36%
baboon512x512	103017(2,1,3)	113788(1,3,2)	104073	9.94%
barbara720x576	105699(2,1,3)	115314(1,3,2)	106577	8.70%
goldhill720x576	102978(2,3,1)	111453(3,1,2)	102978	7.90%
house768x512	105457(2,3,1)	114743(1,3,2)	105457	8.43%
lena512x512	102321(2,3,1)	110012(3,1,2)	102321	7.24%
lighthouse768x512	104441(2,1,3)	112076(1,3,2)	104905	7.05%
motorcross768x512	101065(2,3,1)	108161(1,3,2)	101065	6.78%
parrots768x512	107218(2,3,1)	113520(3,1,2)	107218	5.71%
peppers512x512	102222(2,3,1)	108971(3,1,2)	102222	6.39%
sailboat512x512	99303(2,3,1)	107284(3,1,2)	99303	7.73%
sailboats512x768	107438(2,3,1)	116146(3,1,2)	107438	7.79%

Table 5.2: Number of bits required for 6000 grayscale atoms for different column orders.

In the previous section the encoding procedure has been explained. It was mentioned that lexicographical ordering of rows is a heuristic recommended for database indexes. Before sorting rows it is always possible to apply a fixed permutation of columns. The problem of finding the optimal column order for lexicographical order of rows is in general, similarly to the row ordering problem, NP-complex as pointed out in [66].

To investigate the effect of column sorting on overall performance each permutation was tried for 12 grayscale (see Table 5.2) and colour images (see Table 5.3). The differences in the size of a bit-stream for different column orders are significant. For grayscale, where there are only 6 possible column permutations, the differences between the maximum and minimum bit-stream sizes are from $6-10\%$. For colour, where we have 720 orders, the differences can be up to 20%. In the proposed coding scheme the best, or close to the best, performance is achieved when atoms are sorted by wavelet scale first. Atom indexes and signs of amplitudes are the last sorting criteria for both grayscale and colour. There are column permutations that perform close to optimal for all tested images: $\pi_g = (2,3,1)$ for grayscale and $\pi_c = (5,2,3,4,6,1)$ for colours. The performance difference between sub-

image	the best order	the worst order	(π_c) (5,2,3,4,6,1)	$2 \cdot \frac{worst-best}{worst+best}$
airplane512x512	123815 (3,4,2,5,6,1)	142816 (1,6,5,2,4,3)	124130	14.25%
baboon512x512	124316 (5,3,4,2,6,1)	144858 (1,6,4,5,3,2)	124455	15.26%
barbara720x576	126937 (5,2,3,4,6,1)	148111 (6,1,4,5,3,2)	126937	15.40%
goldhill720x576	124938 (5,3,2,4,6,1)	142009 (1,6,4,5,3,2)	124971	12.79%
house768x512	125048 (2,3,5,4,6,1)	147576 (1,6,4,5,2,3)	125081	16.53%
lena512x512	127077 (5,3,2,4,6,1)	142753 (6,1,4,5,2,3)	127113	11.62%
lighthouse768x512	121129 (5,2,3,4,6,1)	147995 (1,6,4,5,3,2)	121129	19.97%
motorcross768x512	119399 (5,2,3,4,6,1)	141864 (6,1,5,4,3,2)	119399	17.20%
parrots768x512	130512 (5,3,2,6,1,4)	145187 (6,4,1,5,3,2)	130739	10.65%
peppers512x512	128686 (5,3,2,4,6,1)	138515 (6,1,4,5,2,3)	128803	7.36%
sailboat512x512	123035 (5,2,3,4,6,1)	140519 (1,6,4,5,3,2)	123035	13.27%
sailboats512x768	125867 (5,2,3,4,6,1)	147651 (6,1,4,5,3,2)	125867	15.93%

Table 5.3: Number of bits required for 6000 colour atoms for different column orders.

optimal order and the best one is always marginal. The optimal order may be dependent on the method of encoding and its parameters. Experimental results for our method consistently suggest that wavelet scales should be the first column while atom indexes and signs the last two.

5.2.3 Data Modelling

With the use of Algorithm 5.2 the data that constitute the core part of a stream can be classified as: run lengths, raw symbols and atom positions. These data are sent to the arithmetic coder from [126]. Bits needed for header information, group counts and synchronisation form a side part which occupies no more than 5% of the output file size if we consider decomposition into more than 6000 atoms. We counted the number of bits needed for each type of core data. Detailed statistics, for the case without any data modelling (assuming uniform) can be seen in Table 5.4. Most of the bit-stream is occupied by explicit coding of the positions, which is opposite to coding schemes based on Significance Maps where positions are never coded explicitly. Without modelling data $63 - 68\%$ of a grayscale bit-stream is formed by atoms positions ($55 - 61\%$ for colour). However, it has been mentioned in Section 5.2.1 that for dictionaries constructed from 16 1D bases, which corresponds to 256 2D bases, there is usually no more than one atom with all the attributes the same. This fact does depend on dictionary size. In the case of size 256 the most efficient way to encode positions is always just to send a raw number. Therefore, to reduce the size of the stream better methods should be sought for raw symbols and run lengths rather than for positions.

Arithmetic coding allows, knowing the probability distribution of data, us to achieve a compression ratio close to a theoretical bound given by the Shannon's entropy. The key element is a probability model of the symbols' source. Run Length Encoding represents compactly most of the columns in matrix M. Run lengths (R and z_l in Algorithm 5.2) are coded as uniformly distributed in a range $0 \dots s_l$, contributing less than 5% to the output stream size.

On the other hand, modelling the raw symbols v_s as uniformly distributed, which we further refer to as *No-model*, is wasteful as every next symbol in one group is the minimum of the remaining symbols since they are sorted. In our case, if there are s_l symbols $v_s, v_{s+1}, \dots, v_K$ to be encoded, then:

$$v_s = \min\{v_s, v_{s+1}, \dots, v_K\}. \tag{5.18}$$

It is well known in the theory of probability [108] that if $v_s, v_{s+1}, \dots, v_K$ are drawn independently from the same distribution with cumulative distribution function (CDF) F_v then v_s from Equation (5.18) has, if there are s_l symbols, the CDF F_{v_s} given as:

$$F_{v_s}(x) = 1 - (1 - F_v(x))^{s_l}. \tag{5.19}$$

For the discrete case of s_l symbols from the alphabet of size a_l the probabilities can be written as:

$$p(v_s = k) = (1 - F_v(k-1))^{s_l} - (1 - F_v(k))^{s_l}, \text{ for } k = 1 \dots a_l. \tag{5.20}$$

If $v_s, v_{s+1}, \ldots, v_K$ are drawn independently from a distribution $\{p_i\}_{i=1,2,\ldots,a_l}$ then Equation (5.20) follows that:

$$p(v_s = k) = \left(\sum_{i=k}^{n} p_i\right)^{s_l} - \left(\sum_{i=k+1}^{n} p_i\right)^{s_l}, \text{ for } k = 1, \ldots, a_l. \tag{5.21}$$

If the symbols are independently drawn from a uniform distribution then:

$$p(v_s = k) = \frac{1}{a_l} \Rightarrow F_v(k) = \frac{k}{a_l}, \text{ for } k = 1, \ldots, a_l,$$

and the probability distribution for v_s is given by:

$$p(v_s = k) = \frac{(a_l - k + 1)^{s_l} - (a_l - k)^{s_l}}{{a_l}^{s_l}}, \text{ for } k = 1, \ldots, a_l. \tag{5.22}$$

It has to be remembered that the uniform distribution has the highest entropy among discrete distributions and hence the results achieved for a described coding could be still improved. The probabilities in Equation (5.22) are calculated every time a symbol v_s is to be sent directly to the arithmetic coder and are used to update the model during encoding and decoding. We refer to this improvement as *Min-value* model. It can be viewed as a local method of index coding related to interpolative coding from [73]. The performance is significantly improved compared to *No-model* as shown in Table 5.5 and makes the proposed coding system comparable to JPEG 2000 by R-D performance (see Chapter 6). The size of the data targeted by the described refinement i. e. raw symbols (see Table 5.5) is often reduced by a factor of two. The overall coding gains are as high as $15 - 20\%$ for grayscale and from $6 - 13\%$ for colours with the most of the improvement achieved for coding of the atom index column.

5.3 Summary

In this Chapter the full quantisation and coding scheme of MP decomposition has been described and studied. The sufficient conditions for convergence of MP and Multichannel MP with in-loop quantisation have been proven. Experiments with quantisation show that even very coarse in-loop quantisation like PLQ, which introduces high errors especially in initial iterations, can be efficient for MP-based image coding. Moreover, a simple method of colour atom amplitude quantisation based on scalar uniform quantisation has been proposed showing, for RGB image data, superiority over the method based on PLQ used for application to video coding in [130]. The optimal quantisation parameters have been selected considering the coding efficiency.

The proposed encoding algorithm (Algorithm 5.1), inspired by methods for representing database and file indexes [66, 73] and related to MERGE coding of single-channel MP decomposition [78], appears to be a promising idea for colour coding. The idea of MERGE is based on using PLQ quantisation and grouping by bit-planes and atom indexes. The steps forward comparing to MERGE for grayscale are that we also allow grouping by wavelet scale and sign and introduce simple adaptive coding (Algorithm 5.2). The atom

Grayscale Image	Run Lengths	Raw Symbols	Positions
airplane512x512	5806 2.32%	84030 33.63%	159440 63.80%
baboon512x512	9644 3.89%	71184 28.69%	166773 67.22%
barbara720x576	9089 3.61%	74846 29.71%	167469 66.47%
goldhill720x576	9915 3.99%	74506 29.97%	163576 65.79%
house768x512	8553 3.36%	77792 30.53%	167948 65.91%
lena512x512	4376 1.73%	87613 34.69%	159914 63.33%
lighthouse768x512	8171 3.19%	76959 30.04%	170512 66.55%
motorcross768x512	11197 4.59%	69220 28.39%	162922 66.82%
parrots768x512	4178 1.60%	90271 34.59%	165861 63.56%
peppers512x512	9018 3.70%	72738 29.88%	161035 66.16%
sailboat512x512	3800 1.51%	87987 34.96%	159326 63.30%
sailboats512x768	3691 1.41%	88655 33.84%	169052 64.53%
Colour Image	Run Lengths	Raw Symbols	Positions
airplane512x512	9151 3.37%	108045 39.74%	154090 56.67%
baboon512x512	7219 2.64%	103509 37.92%	161688 59.23%
barbara720x576	8036 2.93%	105137 38.28%	160933 58.59%
goldhill720x576	7702 2.84%	104961 38.66%	158213 58.27%
house768x512	6701 2.43%	103009 37.42%	164945 59.93%
lena512x512	10139 3.72%	108696 39.92%	152841 56.13%
lighthouse768x512	6066 2.21%	101875 37.07%	166335 60.52%
motorcross768x512	6284 2.35%	102438 38.24%	158697 59.23%
parrots768x512	9481 3.43%	110472 39.97%	155805 56.37%
peppers512x512	10920 4.02%	108779 40.08%	151032 55.65%
sailboat512x512	9202 3.41%	105893 39.20%	154456 57.17%
sailboats512x768	7316 2.64%	105205 38.03%	163500 59.10%

Table 5.4: Contributions into size of a bit-stream by data type for grayscale and colour coding of 12000 atoms using *No-model*.

parameters can be grouped into columns. We generalised this idea for the case of colour atoms and proposed colour quantisation scheme (see Section 5.1.3). The optimal column orders have been suggested for both grayscale and colour data. The statistics of bit-streams generated by Algorithm 5.1 have been analysed and the potential of the proposed method has been recognised. Studying the effect of dictionary size on the distribution of atom positions is a natural next step to improve coding. It is important to mention here that, the MERGE coding from [78] achieves performance comparable to JPEG 2000 thanks to the use of smaller dictionaries and optimising position coding. Our scheme with optimal ordering of columns and *Min-value* model achieves the same with sending positions in a raw format. The ideas for improvement of position coding in our method for smaller dictionaries are subject of Section 6.4.

grayscale image	*Min-value*: total bits (reduction)	*No-model*: total bits
airplane512x512	203999 (18.36%)	249889
baboon512x512	207072 (16.54%)	248103
barbara720x576	212505 (15.66%)	251962
goldhill720x576	208659 (16.08%)	248626
house768x512	210615 (17.35%)	254815
lena512x512	205437 (18.65%)	252524
lighthouse768x512	213048 (16.84%)	256198
motorcross768x512	205786 (15.61%)	243838
parrots768x512	212949 (18.39%)	260944
peppers512x512	206277 (15.25%)	243402
sailboat512x512	201630 (19.89%)	251700
sailboats512x768	213616 (18.46%)	261983
colour image	*Min-value*: total bits (reduction)	*No-model*: total bits
airplane512x512	248245 (8.70%)	271886
baboon512x512	246239 (9.79%)	272964
barbara720x576	251630 (8.39%)	274685
goldhill720x576	247995 (8.66%)	271502
house768x512	244581 (11.14%)	275244
lena512x512	254419 (6.57%)	272306
lighthouse768x512	240434 (12.52%)	274840
motorcross768x512	234486 (12.48%)	267914
parrots768x512	257170 (6.96%)	276399
peppers512x512	254062 (6.38%)	271378
sailboat512x512	247395 (8.43%)	270166
sailboats512x768	248604 (10.13%)	276637

Table 5.5: Reductions of a bit-stream size for grayscale and colour coding of 12000 grayscale and colour atoms with *Min-value* model.

6 Evaluation of Coding Results

We studied the transform part of MP-based codecs in Chapter 4 and, then, proposed a novel quantisation and coding scheme in Chapter 5. In this chapter, the performance of the whole coding system is analysed and compared with the state-of-the-art scalable image codecs: JPEG 2000 and SPIHT. We show, for a range of different RGB images, the potential of MP with wavelets for scalable colour image coding.

Section 6.2 and Section 6.1 discuss the flexibility of the MP for colour image coding. In Section 6.1 the possibility of optimising different criteria than MSE across all channels is explored. Then, in Section 6.2 we explore the alternative idea to MMP based on a single-channel MP in decorrelated colour space such as YC_bC_r. In Section 6.3 we analyse the coding performance of our codec in comparison to the standards. We explore potential ways to improve coding performance in Section 6.4. We tackle the question of the dictionary size for a fixed coding algorithm (in this case the method proposed in Chapter 5). Section 6.5 summarises the main results and concludes the chapter.

6.1 Atom Search Criteria

The multichannel MP introduced in Section 3.5 is truly flexible in terms of atom search criteria. At first, we show that changing atom selection criterion to best match the Y-channel can optimise Y-PSNR. However, we point out that a care must be taken to keep a decomposition algorithm convergent in the signal space. We could minimise Y-PSNR at each step of the MMP by minimising the weighted MSE which is equivalent

to maximisation of the *W-PSNR*. This can be easily done by changing atom selection criterion. It can be shown, by extending the result about MSE minimisation (RGB-PSNR maximisation) from Section 3.5, that the atom selection criterion becomes:

$$\max_{g\in\mathcal{D}} \left|\alpha\langle Rf^{(r)}, g\rangle + \beta\langle Rf^{(g)}, g\rangle + \gamma\langle Rf^{(b)}, g\rangle\right|. \tag{6.1}$$

The parameters are: $\alpha = 0.299, \beta = 0.587, \gamma = 0.114$ as in the RGB to $\mathrm{YC_bC_r}$ transform (see Section 2.6).

Comparing Y-PSNR averaged over 12 test images against JPEG 2000 (Figure 6.1b) we achieved a significant improvement for the MMP with Y-channel atom selection. On the other hand it is clearly visible in Figure 6.1a that now the RGB-PSNR values are very low. Figure 6.4 presents *Lighthouse* decoded at 0.5 bpp with the MMP minimising only Y-channel distortion. The issue is that visual artefacts originated in colour distortion are clearly present. It is visible that some of the areas of the sky are represented with false colour. For comparison, Figure 6.3 shows the same image decomposed from similar number of atoms using MMP while Figure 6.2 presents results of decoding with JPEG 2000 for the same rate of 0.5 bpp as used in Figure 6.4.

A non-linear nature of MP makes it difficult to control such effects. MMP with the criterion expressed by (6.1) can converge to any point $Rf = [Rf^{(r)}, Rf^{(g)}, Rf^{(b)}]$ such as $\alpha Rf^{(r)} + \beta Rf^{(g)} + \gamma Rf^{(b)} = 0$. This case is a good example where an IQM which does not account for colour information, such as Y-PSNR dramatically fails as a measure of distortion. It also shows that selection criteria that do not affect convergence of decomposition algorithm has to be used.

Secondly, we can compare L_2-maximisation used so far against the further two criteria.

L_1-maximisation that selects atom g that at n-th iteration maximises:

$$L_1 = |\langle Rf_n^{(r)}, g\rangle| + |\langle Rf^{(g)}, g\rangle| + |\langle Rf^{(b)}, g\rangle|, \tag{6.2}$$

and L_∞-maximisation, also refereed to as replicated MP [130], where we select an atom that maximises:

$$L_\infty = \max\{|\langle Rf^{(r)}, g\rangle|, |\langle Rf^{(g)}, g\rangle|, |\langle Rf^{(b)}, g\rangle|\}. \tag{6.3}$$

The proof of convergence was presented in [118] for L_∞-maximisation. We extended the proof from [118] in Chapter 5 to take into account quantisation and MSE minimisation. We also showed in Section 5.1 that atom selection according to the maximum L_2-norm minimises joint mean squared error. Therefore, it is not surprising that in terms of the MSE, the L_2-norm is the best criterion. The gain comparing L_2-norm against Replicated MP or L_1-norm is small but statistically significant (t-tests) and consistent. Figure 6.1 shows the comparison of performance averaged over 12 test images at different bit-rates. In terms of Y-PSNR and SHSIM, the L_1 and L_2 norms perform similarly. L_2 achieves slightly higher averages but not statistically significant for a given test set and significance level 0.05. L_∞-maximisation is significantly inferior in terms of any metrics tried.

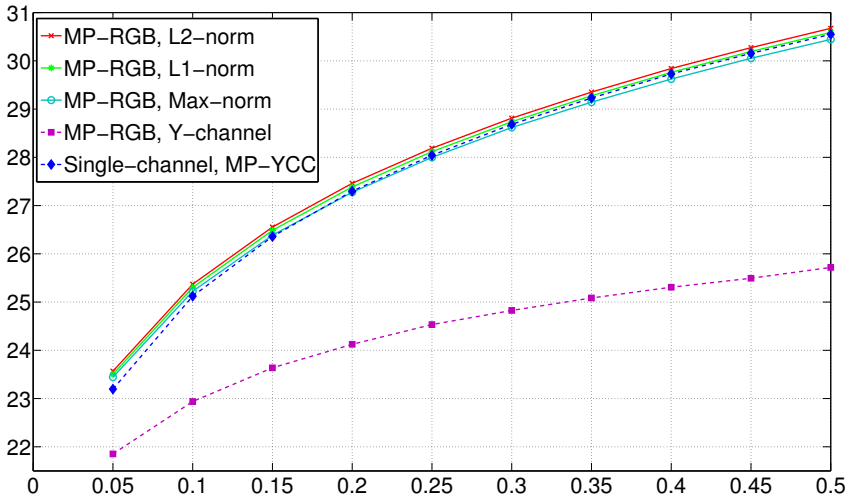
6.2 Colour Space Choice

So far we focused our attention only on encoding multi-channel decomposition obtained by the MMP. An alternative way to use the MP for colour images is to apply a single-channel algorithm to each channel in a decorrelated colour space. A similar idea of encoding performed after decorrelating transform is implemented in JPEG 2000 and colour version of SPIHT [111]. In Section 4.1.2, we observed that in this way many more atoms are needed to achieve the same distortion. However, if we take quantisation and encoding into consideration, a fewer parameters are needed to define one atom in case of the single-channel algorithm. We can easily adapt our method of coding (Section 5.2) to encode MP decomposition obtained in the luma-chroma colour space. The attributes of each atom now include:

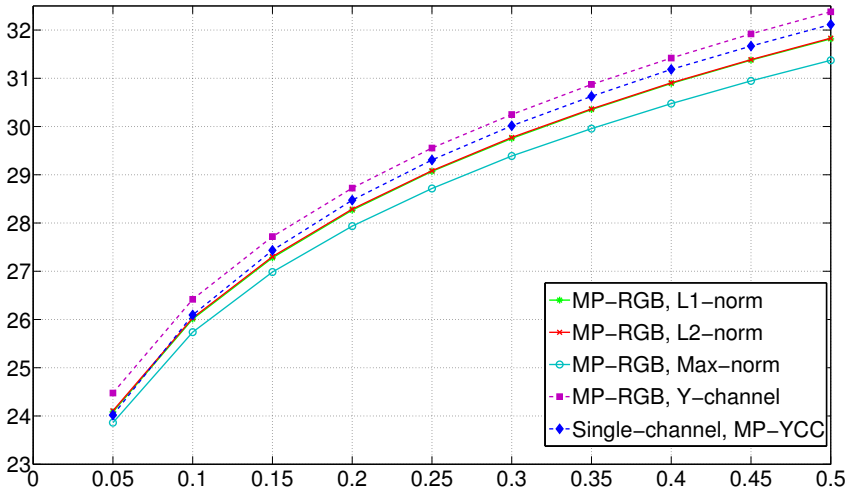
1 :	s_n	sign of the amplitude,	$s_n \in \{-1, 1\}$,
2 :	c_n	colour channel,	$c_n \in \{1, 2, 3\}$,
3 :	w_n	sub-band index,	$w_n \in \{1, 2, \ldots, 3S+1\}$,
4 :	λ_n	2D dictionary entry,	$\lambda_n \in \{1, 2, \ldots, B^2\}$,
5 :	x_n	atom location inside the sub-band w_n,	$x_n \in \{1, \ldots, W_{x_n}\}$,
6 :	y_n	atom location inside the sub-band w_n,	$y_n \in \{1, \ldots, W_{y_n}\}$.

The difference comparing to grayscale coding is that an additional attribute c_n was added to indicate the colour channel. Note, that in the case of the MMP, there were two additional attributes related to quantised amplitudes. We apply the same coding algorithm as described in Section 5.2 (Algorithm 5.2) to columns 1-4. Grouping is done in order to form an input to the encoder starting from subband index w_n followed by the colour channel c_n, dictionary entry λ_n and the amplitude sign s_n.

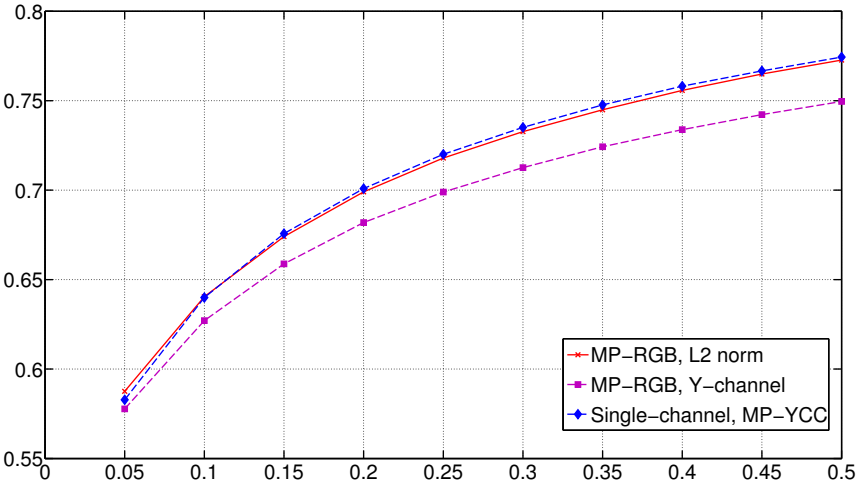
We can now evaluate the coding performance of the single-channel MP in $\mathrm{YC_bC_r}$ colour space (MP-YCC) comparing to MP-RGB. Performing MP-YCC gives much lower distortion for the Y-channel than MP-RGB. Results in Figure 6.1b visualise a gain in distortion measured by the Y-PSNR. The Y-PSNR gain is not surprising considering the numbers of atoms found for each channel. For example, for 12156 atoms used to encode *Goldhill* image at 0.50 bpp 10169 are found in the Y-channel and only 835 in $\mathrm{C_b}$ and 1152 in $\mathrm{C_r}$. There are no annoying artefacts being introduced since, unlike when using selection criterion from Equation (6.1), the method is guaranteed to converge. However, it is interesting to note that Y-PSNR performance is significantly worse than if optimising Equation (6.1). This suggests some space to improve decorrelating properties of colour transform to be used with MP. On the other side, if we measure distortion by the RGB-PSNR a multichannel decomposition is preferred as shown in Figure 6.1a. The two ideas are comparable in terms of SHSIM metric (Figure 6.1c).



(a) RGB-PSNR



(b) Y-PSNR



(c) SHSIM

Figure 6.1: Average R-D comparison of different atom selection criteria (x-axis: bit-rate [bpp], y-axis: IQM values).

Figure 6.2: *Lighthouse* decompressed with default mode of JPEG 2000 at 0.50 bpp.
Y-PSNR=29.96, RGB-PSNR=29.57, Y-M-SSIM=0.8811, SHSIM=0.8072 (SSIM=0.8695, HSIM=0.4962)

Figure 6.3: *Lighthouse* decomposed by RGB-MP with L_2-norm minimisation into 9840 atoms.
Y-PSNR=30.25, RGB-PSNR=29.95, Y-M-SSIM=0.8732, SHSIM=0.8017 (SSIM=0.8635, HSIM=0.4926)

Figure 6.4: *Lighthouse* decomposed from 9867 atoms (0.50 bpp) selected optimising the Y-channel.
Y-PSNR=30.79, RGB-PSNR=26.71, Y-M-SSIM=0.8818, SHSIM=0.7893 (SSIM=0.8596, HSIM=0.4383)

6.3 Comparisons with Standards

In this section we compare the proposed coding of MP and MMP decompositions against SPIHT and EBCOT (JPEG 2000). For all the experiments we apply 5-scale wavelet decomposition using CDF 9/7 filters. Five wavelet scales are, for considered images, typically the best choice. For MP and MMP, the same dictionary $\mathcal{D}_{16}^{(t)}$ is used for colour and grayscale, following conclusions of Chapter 4.

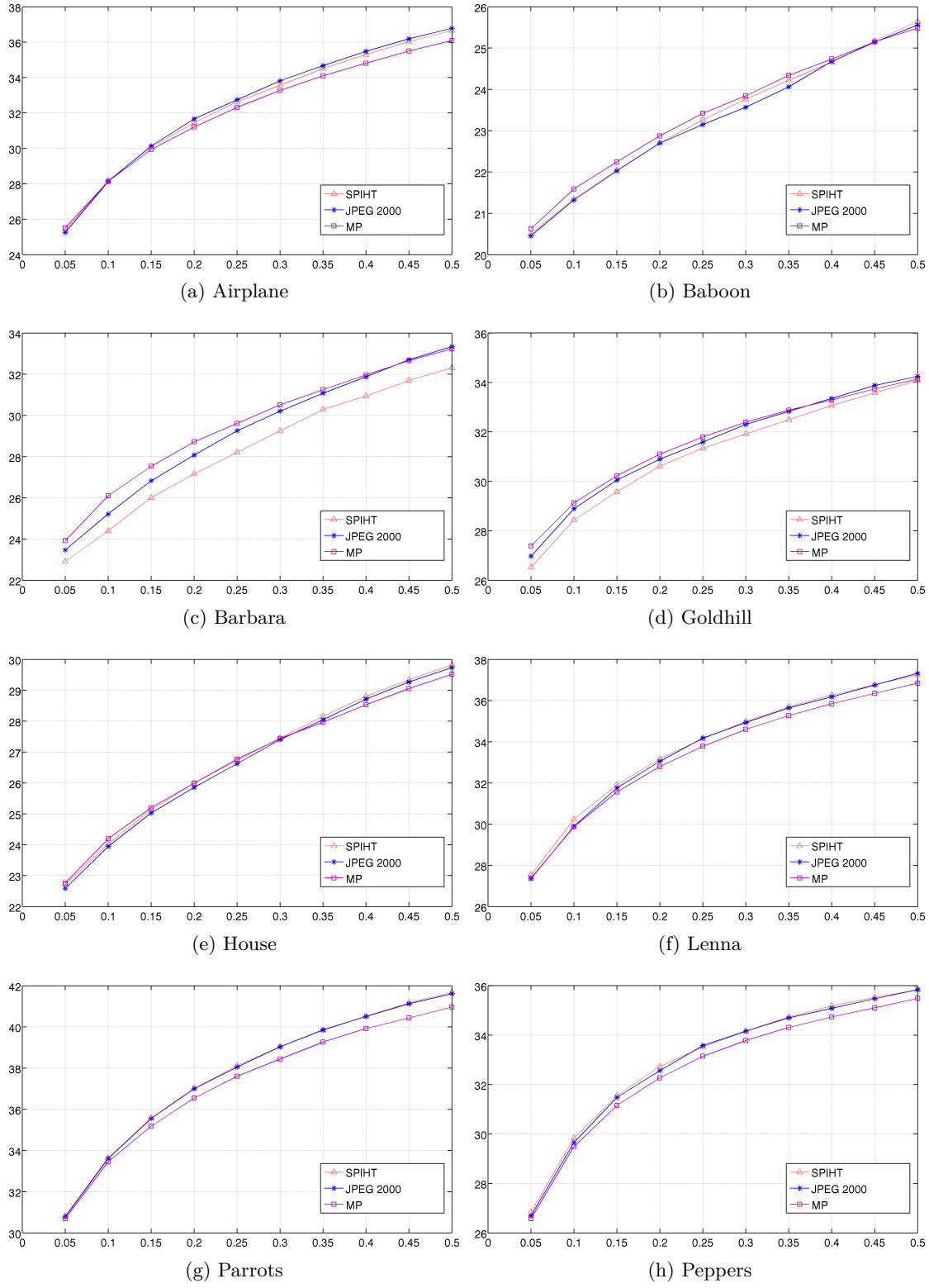
Our encoder and decoder are implemented in separate programs to ensure correct R-D results by measuring the actual output file sizes. The final distortion at each bit-rate was calculated after the pixel values were rounded to the nearest integer values. This process is performed to keep consistency with how JPEG 2000 and SPIHT are tested. For the SPIHT we use freely available executable programs from [111] while for the JPEG 2000 we used the Kakadu implementation [113] from the author of the EBCOT encoding algorithm [114].

In practical image compression each stream is preceded by the header that contains information such as image size and compression parameters. For the SPIHT, 6-7 bytes are used in a message header. For MP-RGB, we add 100 bits of header information which is 12.5 bytes. JPEG 2000 adds header information for each packet in addition to the main header which for the default setting occupies a few bytes. In all cases, header bytes are added to the coding rates. For image sizes bigger than 512×512 and rates higher than 0.05 bpp these few bytes correspond to insignificant contributions to the bit-rate.

6.3.1 R-D Performance

We compare JPEG 2000, SPIHT and MP algorithms on single and multi-channel images. We present individual results using PSNR in Figures 6.5 and 6.6 and averages in Figure 6.7. Despite the shortcomings of the MSE as the image quality metric, we believe that it is the fairest way to compare compression algorithms designed with MSE-minimisation in mind. For colour images we measure here RGB-PSNR (i. e the joint MSE introduced in Equation (2.12)). Colour SPIHT is known to be particularly efficient for colour images in terms of RGB-PSNR [111]. This is partially thanks to the use of the KLT as a colour transform for each image in opposition to the fixed RGB to $\mathrm{YC_bC_r}$ transform used by JPEG standards. For fair comparison, we use the *no_weights* option of Kakadu software which forces a joint optimisation of MSE analogously to the Colour SPIHT and ours. The default mode of JPEG 2000 applies visual weights to different colour channels and subbands in order to potentially improve the visual appearance of images. The effect is that many more bits are assigned to encode data from lower-frequencies subbands and the RGB-PSNR values are much lower especially at higher rates. This will be discussed using visual examples in Section 6.3.2.

Table 6.1 collects comparative results of the average PSNR values. At rates lower than 0.25 bpp, MP achieves higher average PSNR than JPEG 2000. The situation turns around from 0.25 bpp upwards. SPIHT is inferior to MP and JPEG 2000 average performance at low bit-rates but its colour version achieves the highest values. In Figure 6.6 and Figure 6.5 full R-D curves are presented for selected images for bit-rates up to 0.50 bpp.



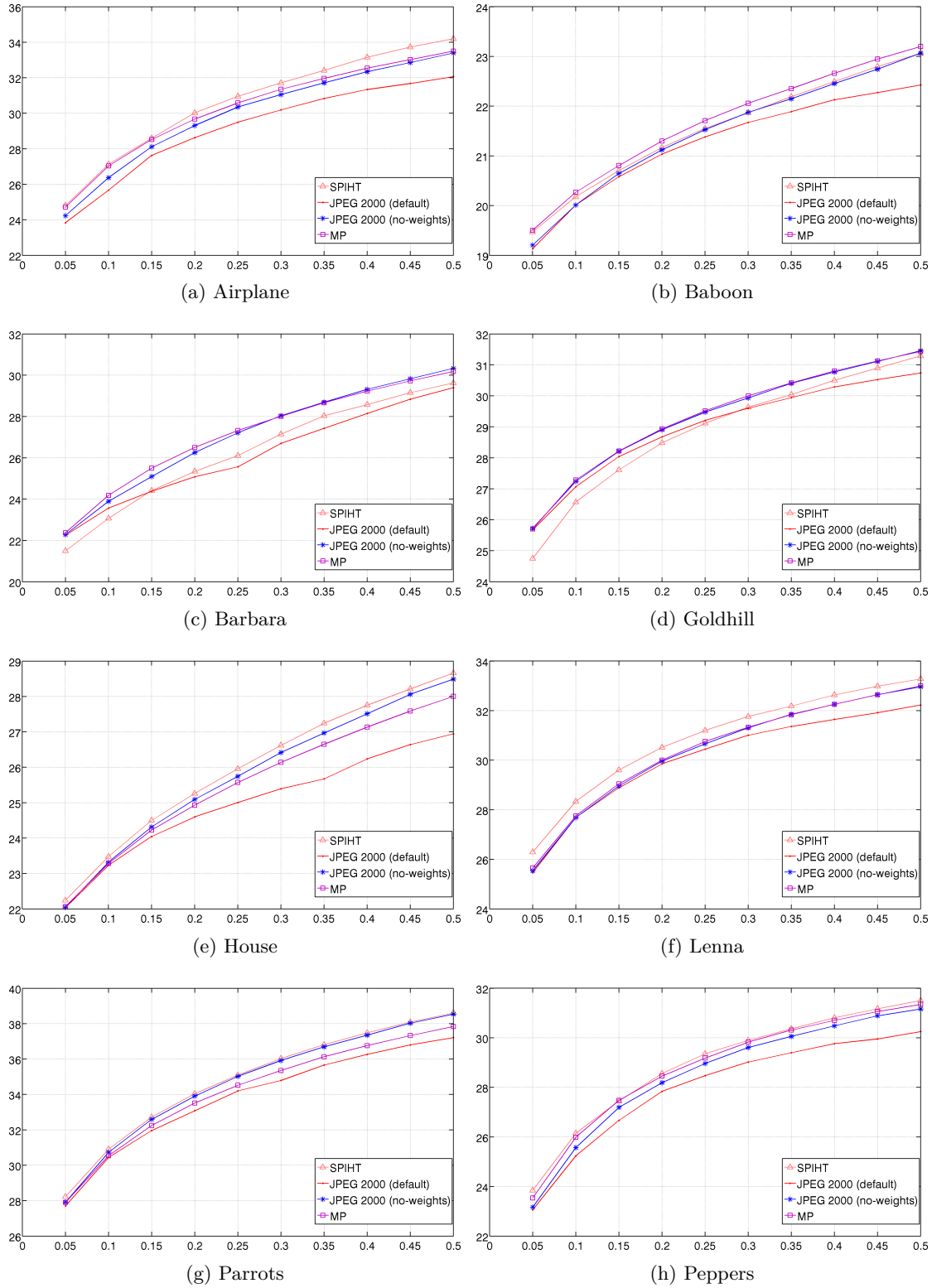
(a) Airplane (b) Baboon

(c) Barbara (d) Goldhill

(e) House (f) Lenna

(g) Parrots (h) Peppers

Figure 6.5: R-D comparisons between JPEG 2000, SPIHT and MP for different grayscale images (x-axis: bit-rate [bpp], y-axis: PSNR [dB]).



(a) Airplane (b) Baboon

(c) Barbara (d) Goldhill

(e) House (f) Lenna

(g) Parrots (h) Peppers

Figure 6.6: R-D comparisons between JPEG 2000, Colour SPIHT and MP-RGB for different colour (RGB) images (x-axis: bit-rate [bpp], y-axis: RGB-PSNR [dB]).

Codec	0.10 bpp	0.25 bpp	0.50 bpp
Grayscale JPEG 2000	26.95	**30.45**	**33.66**
Grayscale SPIHT	26.90	30.33	33.54
Grayscale MP	**27.09**	30.35	33.30
Colour JPEG 2000	25.24	28.23	30.89
Colour SPIHT	**25.42**	**28.33**	**31.01**
Colour MP	25.37	28.19	30.67

Table 6.1: Average PSNR performance of codecs over 12 test images.

It is clear that different relative results are obtained for different images. For example, for grayscale *Barbara* MP is clearly superior. The grayscale SPIHT is the worst for *Barbara* and *Goldhill* but for the other images performs similarly to JPEG 2000. For grayscale *Parrots* and *Peppers* MP performs the worst. The variation of the results is even higher for colour images.

As the differences between PSNR values approximately follow a normal distribution (which was checked using the Kolmogorov-Smirnoff test) we could use paired t-tests to compare the average performance in the same way as when studying different configurations of the MP system in Chapter 4. For the low bit-rates up to 0.5 bpp at 0.50 bpp SPIHT is by average PSNR statistically significantly better (at significance level 0.05) than both MP-RGB and JPEG 2000 for grayscale. For lower rates there is no evidence for superiority of any of the codecs. In fact from the more detailed analysis of the results of t-tests we can say that SPIHT outperforms our MP-RGB at rates higher than 0.45 bpp and JPEG 2000 at higher than 0.40 bpp. We can draw the conclusion here that all the codecs perform, in terms of PSNR, similarly at low bit-rates. At higher rates MP is slightly worse but it has to be remembered here that the target of our method was mainly low bit-rate image/video coding. Nevertheless, there is a potential for using our code also at medium rates, for example, the position encoding could be improved which is the topic of Section 6.4. In the next section we look closer at compression artefacts introduced by the studied methods at low bit rates.

6.3.2 Visual Evaluation

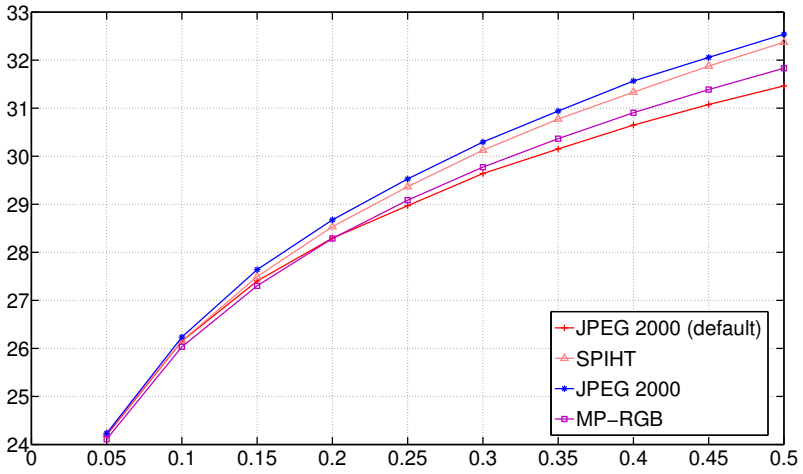
In Section 6.3.1 we compared JPEG 2000, SPIHT and MP-RGB as MSE minimizers for RGB-images. However, when designing still-image compression system, the method that gives the best looking image is desired. Although, as outlined in Section 2.3.3, there are a lot of advances in image quality assessment to judge image and video codecs, visual comparisons need to be performed. Impressions about quality of visual data are purely subjective. Not only the distortion of the actual data is important but also viewing conditions and the context within which the image is viewed. For example, the same image will be perceived differently on the screen than in print. Practically, when designing a general purpose method, we are restricted to analyse only the distortion of the actual data. Better correlation with human perception of visual data is usually achieved by

adapting optimisation criterion that attempts to model the HVS. A notable example is the default mode of JPEG 2000 where visual weights are added for the different image channels and for different wavelet subbands in YC_bC_r colour space.

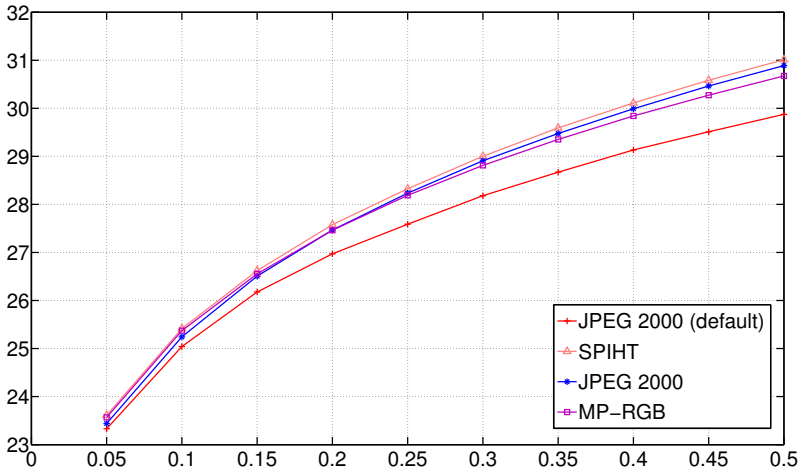
The Y-PSNR metric could be used to reflect the fact that the HVS is most sensitive to luminance information. Figure 6.7a shows a comparison of average Y-PSNR performance. The JPEG 2000 default mode still achieves low values compared to colour SPIHT and JPEG 2000. Our method achieves significantly lower Y-PSNR values compared to the other algorithms. This is due to the fact that JPEG 2000 perform MSE optimisation after YC_bC_r colour transformation and colour SPIHT uses KLT, which still favours Y-channel information.

Further, Figure 6.7c shows results averaged over 12 test images using a colour version of SSIM metric outlined in Section 2.3.3 (SHSIM). It shows that the performance of all the methods is close, favouring both modes of the JPEG 2000 at higher rates, while MP-RGB performs similarly to SPIHT.

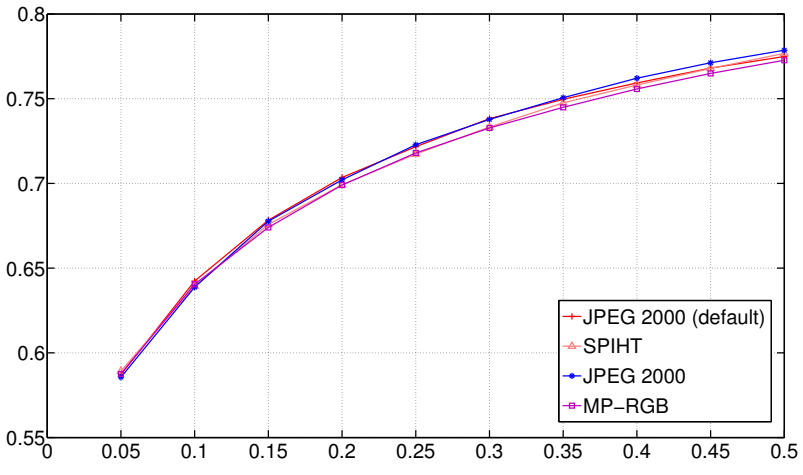
Examples of the visual comparison of the studied codecs are shown in Figure 6.9 for the *Goldhill* and Figure 6.8 for the *Barbara*. Both images are decompressed at 0.30 bpp which corresponds to $CR = 80 : 1$. We compare JPEG 2000 in default and MSE-minimising mode, SPIHT and MP-RGB. In general the images decomposed using the default setting of Kakadu JPEG 2000 appear to be the sharpest even though the PSNR values are by far the lowest in that case. It seems there is more blurring and ringing introduced when just minimising MSE than by the other methods. Visually images represented by a default mode of JPEG 2000 seem to be the least blurred. However it seems that MP is capable of decomposing some patterns better than the other methods: for example a texture checker-board-like patterns which can be seen on the roof of the house in Figure 6.11 and the pattern on the chair in Figure 6.10. This is thanks to enriched set of filters in redundant dictionary applied in the wavelet domain. Possible directions to improve performance, related mainly to dictionary design are outlined in the next section.



(a) Y-PSNR



(b) RGB-PSNR



(c) SHSIM

Figure 6.7: Average R-D performance comparison using different metrics (x-axis: bit-rate [bpp], y-axis: IQM values).

(a) Original image: *Barbara*

(b) JPEG 2000 default.
RGB-PSNR=26.70,
Y-PSNR=27.67, Y-M-SSIM=0.8295,
SHSIM=0.7290

(c) SPIHT.
RGB-PSNR=27.14,
Y-PSNR=28.08, Y-M-SSIM=0.8181,
SHSIM=0.7233

(d) JPEG 2000 no_weights.
RGB-PSNR=28.03,
Y-PSNR=29.37, Y-M-SSIM=0.8552,
SHSIM=0.7524

(e) MP-RGB. 5902 colour atoms.
RGB-PSNR=28.01,
Y-PSNR=29.09, Y-M-SSIM=0.8537,
SHSIM=0.7545

Figure 6.8: Visual comparisons for colour *Barbara* at 0.30 bpp.

(a) Original image: *Goldhill*

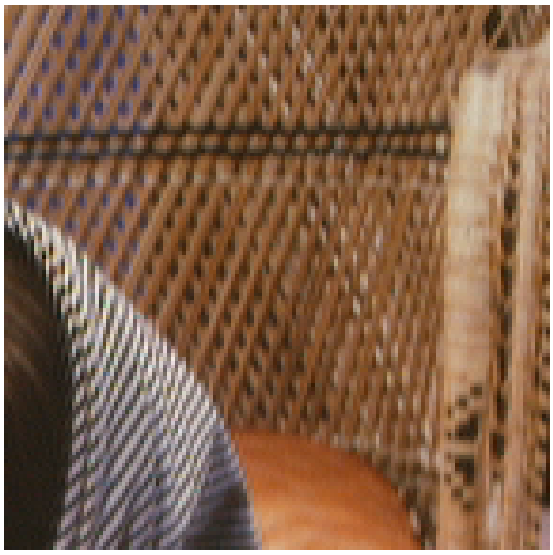
(b) JPEG 2000 default.
RGB-PSNR=29.60,
Y-PSNR=31.34, Y-M-SSIM=0.8233,
SHSIM=0.7345

(c) SPIHT.
RGB-PSNR=29.64,
Y-PSNR=30.97, Y-M-SSIM=0.7912,
SHSIM=0.7114

(d) JPEG 2000 no_weights.
RGB-PSNR=29.93,
Y-PSNR=31.58, Y-M-SSIM=0.8121,
SHSIM=0.7280

(e) MP-RGB. 5902 colour atoms.
RGB-PSNR=30.00,
Y-PSNR=31.30, Y-M-SSIM=0.8089,
SHSIM=0.7300

Figure 6.9: Visual comparisons for colour *Goldhill* at 0.30 bpp.

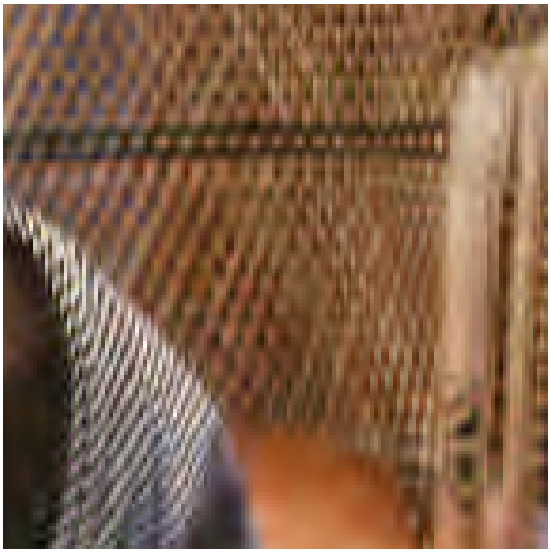


(a) Original image fragment.

(b) JPEG 2000 default.

(c) SPIHT.

(d) JPEG 2000 no_weights.



(e) MP-RGB. 5902 colour atoms.

Figure 6.10: Visual comparisons for fragment of *Barbara* of size 144×144 at 0.30 bpp.

(a) Original image fragment.

(b) JPEG 2000 default.

(c) SPIHT.

(d) JPEG 2000 no_weights.

(e) MP-RGB. 6065 colour atoms.

Figure 6.11: Visual comparisons for fragment of *Goldhill* of size 144×144 at 0.30 bpp.

6.4 Coding and Dictionaries

We have described in Chapter 4 how the size of a dictionary affects the computational complexity of our MP implementation. It was also shown that with increasing number of bases in the dictionary a sparser approximation can be obtained for a given distortion. Using a simple iterative method to build dictionaries (Basis Picking) allowed us to analyse the rate of increase in both encoder complexity and approximation quality. The complexity increases quadratically with a dictionary size and cubically with the maximal filter length (see Equation (4.18)). Adding more bases to a dictionary gives performance gain for a fixed number of atoms. However with increasing size of the dictionary the gain becomes lower and lower. For example in our case on average for 10 images (see Figure 4.9) we can gain almost 0.5 dB for 6000 atoms by increasing dictionary size from 16 to 24 for both grayscale and colour. For comparison, starting from one generator there is more than 2.5 dB difference when comparing against a dictionary of size 8.

The situation changes when the actual coding is considered. We expect that with the bigger dictionary fewer atoms are needed for the same distortion but more bits are needed to encode one atom. It can be seen in Table 6.2 that the gains in PSNR at fixed bit-rate are significantly lower than at a fixed number of atoms. On average there is no gain after adding more than 16 bases to the dictionary for colour coding while for grayscale we can improve only by 0.07 dB. The results in Table 6.2 are obtained for the

Dictionary:	$\mathcal{D}_1^{(t)}$	$\mathcal{D}_4^{(t)}$	$\mathcal{D}_8^{(t)}$	$\mathcal{D}_{11}^{(t)}$	$\mathcal{D}_{\mathbf{16}}^{(\mathbf{t})}$	$\mathcal{D}_{24}^{(t)}$
Colour	28.03	29.58	29.95	30.06	**30.14**	30.14
Grayscale	29.26	31.66	32.35	32.55	32.72	**32.79**

Table 6.2: Average PSNR over 10 test images compressed at fixed rate of 0.5 bpp using dictionaries of different size.

same learning process (on colour images) as the results presented in Figure 4.9. However, the same conclusions are valid for other experiments that were considered in Chapter 4. Note that the absolute values of PSNR from Table 6.2 and Figure 4.9 cannot be compared directly as the results presented here are for a fixed bit-rate and the number of encoded atoms can vary from image to image. However it is appropriate to compare the relative differences for varying number of bases. A discrepancy in conclusions that can be drawn from both cases highlight the importance of taking coding into account when considering sparse approximations in compression applications rather than relying on sparsity on its own as the performance indicator.

The problem with encoding is that due to the complex nature of the output of MP decomposition different approaches to coding exploit redundancy in different way ignoring some dependencies present in the data. For example, in [84] atoms were grouped by blocks and position differences from the centre of each block were encoded rather than the raw position coordinates thus reducing the number of bits spent on position coding. On the other hand, more bits were needed to explicitly encode the quantised amplitude. Alternatively atoms can be encoded, as in this work, by decreasing amplitude. In this

way, the bits spent on the encoding of the amplitudes are saved at the cost of more bits spent on atom locations. Experiments reported in the literature suggest that the two approaches are approximately equivalent in terms of coding performance [107]. However, for application in scalable (SNR-scalable) and embedded image coding it is natural to encode the more significant atoms first. Encoding atoms by decreasing amplitude from the lower to higher wavelet scale supports, in a similar fashion to EBCOT and SPIHT, both resolution and SNR scalability.

We can assume that the final size of the stream (R) is proportional to the number of atoms (N) and the average number of bits used per one atom ($\overline{A}$):

$$R = N\overline{A}\,(B), \tag{6.4}$$

where B is the size of the dictionary. From the information theory point of view, if dictionary entries are equally likely to appear in the decomposition then we expect A to be proportional to the logarithm of number of bases B:

$$A\,(B) \sim \log_2(B). \tag{6.5}$$

Figure 6.12 presents the functions of the form: $\alpha + \beta \log_2(B)$ fitted to the actual results for average number of bits per atom computed for the test image *Lenna*. Parameters α and β depend on the contribution of the other atom parameters such as positions (see Section 5.2) and can differ across different images.

It can be assumed that, for our method of coding, the number of bits needed to encode one atom is proportional to $\log_2(B)$. Although the number of bits per atom can be estimated, owing to a non-linear nature of MP, it is difficult to predict the value of distortion. The choice of 16-bases-dictionaries used throughout this thesis is a trade-off between coding performance and computational complexity since above certain number of bases (16) there is no significant improvement in PSNR for fixed rates when adding more generators to the dictionary.

Moreover, it has to be understood that the idea for atom encoding proposed in Chapter 5 is very general. Statistical modelling of index distributions for particular dictionaries and optimising position coding could potentially reduce the size of the streams. For example, as mentioned in Chapter 5, atom positions are sent in a raw form. For smaller dictionaries, employing Golomb codes [78] or performing deeper recurrence in Algorithm 5.1 (i. e. also for atom positions) could further reduce the size of the stream.

Encoding atom positions in Algorithm 5.1 is equivalent to indicating locations of the particular 2D basis within a subband. It can happen that there are many atoms with common parameters.

The significance map M can be defined for each set of common parameters as a binary map of length $n = W \times H$ if the subband size is $W \times H$:

$$M = (c_1 c_2 \ldots c_n)\,,\ c_i \in \{0, 1\}, \tag{6.6}$$

with $c_i = 1$ indicating the atom at location i. If we have just one atom then without any prior knowledge about how the atoms are distributed the best encoding uses $\log_2 n$

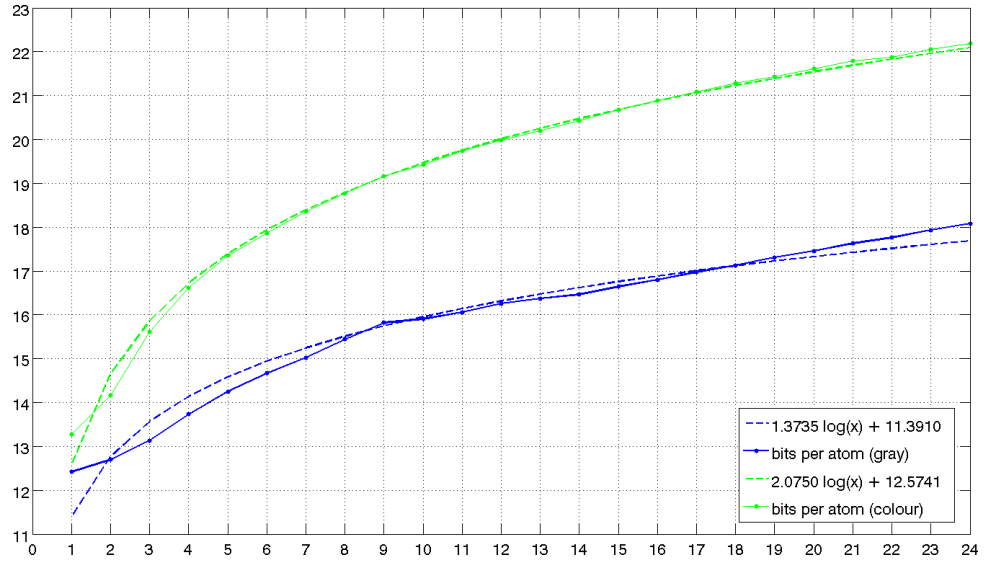


Figure 6.12: Average number of bits per one atom (y-axis) for decomposition to 0.5 bpp for different dictionary sizes (x-axis) for colour and grayscale *Lenna*.

bits, with n being the number of ways you can place one atom into n locations (bins). This is exactly the number of bits needed to encode an integer in the range $[0...n-1]$. In practice, vertical and horizontal locations are sent separately to the arithmetic coder. Using arithmetic coding allows us to assign a number of bits per symbol that is close to optimal even when n is not a power of 2. We separate x and y locations in order to keep the values sent to arithmetic coder small so that we are able to maintain feasible size of the model for arithmetic coding.

Theoretically, the amount of information when placing k atoms into n bins is $\log_2 \binom{n}{k}$ bits. Let us estimate what could maximally be gained in this way for real images in comparison to raw position coding. In the raw case we need $k \log_2 n$ bits. For images of size 512×512 the lowest frequency subband, which is the smallest one, is of size 16×16, hence $n = 256$. In that case if $k = 2$ then the difference:

$$k \log_2 n - \log_2 \binom{n}{k} = |_{k=2}\ 2 \log_2 n - \log_2 \frac{n(n-1)}{2} =$$
$$= 2 \log_2 n - \log_2 n - \log_2 (n-1) + \log_2 2 \approx |_{n=256}\ 1.0056 \text{ bits}$$

In the more general case of $k \ll n$, which holds in our case, we can put $\log_2(n) \approx \log_2(n-1) \approx \ldots \approx \log_2(n-k)$ and then we have:

$$k \log_2 n - \log_2 \binom{n}{k} \approx \log_2 k!. \tag{6.7}$$

For the *Lenna* image encoded using a dictionary of size 8 ($D_8^{(t)}$) at 0.50 bpp we could theoretically make a 10% improvement by optimising position coding. If a dictionary is of size 16 then we cannot expect more than 4% at the same rate. As the number of repeating atoms increases with the number of iterations for the same image, at lower rate of 0.10 bpp the improvement is only about 1.5% for $D_{16}^{(t)}$ and about 5.6% for $D_8^{(t)}$. Optimising the position coding is an obvious step to improve the encoder for grayscale images. The

situation is different for colour data where the potential gains are up to 10 times smaller than for grayscale. The numerical results for *Lenna* are collected in Table 6.3. We can see that only the fraction of atoms (fourth column comparing to the third one) can serve as a basis for position encoding improvement. The benefit could be significant only for grayscale images and especially at higher bit-rates. It has to be remembered that the cost of coding improvement is increased complexity of the encoder as well as the decoder and that the values shown in Table 6.3 are the theoretical bound.

We can see here that changing the size of a dictionary affects the distribution of atom parameters. Data modelling for smaller dictionaries and also for alternative way of image partitioning introduced in Section 4.1.1 still remains an open problem.

Grayscale					
Dictionary	Rate	No. Atoms	Repetitions	Saved bits	Improvement
$\mathcal{D}_{16}^{(t)}$	0.5 bpp	7800	1988	4818	3.68 %
$\mathcal{D}_{16}^{(t)}$	0.1 bpp	1535	276	403	1.53 %
$\mathcal{D}_{8}^{(t)}$	0.5 bpp	8489	2047	14094	10.75 %
$\mathcal{D}_{8}^{(t)}$	0.1 bpp	1774	526	1479	5.64 %
Colour					
$\mathcal{D}_{16}^{(t)}$	0.5 bpp	6278	348	405	1.55 %
$\mathcal{D}_{16}^{(t)}$	0.1 bpp	1251	33	34	0.13 %
$\mathcal{D}_{8}^{(t)}$	0.5 bpp	6988	886	1479	1.13 %
$\mathcal{D}_{8}^{(t)}$	0.1 bpp	1394	75	106	0.40 %

Table 6.3: Maximal theoretical coding gain from optimising position encoding.

6.5 Summary

We have shown that MP-based methods can compete with the industrial image compression standards in terms of compression ratio. We compared average results on a set of test images with the aid of statistical tests. The idea for coding MP-decomposition introduced in Chapter 5 was used. Some ideas for improvements at coding stage were also presented. The two methods of applying MP to colour images, the first based on the Multichannel MP performed directly in RGB colour space (MP-RGB) and the second based on the single-channel MP performed after a decorrelating transform (MP-YCC) were analysed and found out to perform similarly. Different atom selection criteria for multichannel algorithm were compared. The importance of preserving convergence in the signal space has been highlighted. Finally, we compared a proposed MP-RGB codec against well-established wavelet-based methods. A similar performance for both MP-RGB and MP-YCC in terms of selected IQMs to SPIHT and JPEG 2000 have been observed at bit-rates up to 0.50 bpp. A potential to better represent some image features using MP-based methods has been also recognised. In addition, some of the visual results obtained in this chapter highlighted the limitations of existing objective quality metrics.

7 Conclusions and Directions for Future Work

In this chapter we summarise our findings and propose directions for future work. The main focus of this thesis was on the problem of sparse signal approximation of multichannel signals such as RGB colour images. The application in mind is scalable image coding capable of achieving high compression. The problem of image coding was analysed in two parts: signal representation and encoding of that representation into a bit-stream. We introduced general concepts from signal processing, theory of information and coding together with presentation of the state-of-the art methods in image coding in Chapter 2. Then we concentrated on signal representation in Chapters 3-4 and focused on quantisation and encoding in Chapter 5. Finally, in Chapter 6 the proposed codec was benchmarked against EBCOT and SPIHT. For the test images, the performance in terms of Rate-Distortion at low and medium rates came out to be on average statistically indistinguishable from the standards. This means a better compression for some of the images and worse for the others. However, further ideas to improve coding were suggested, and it was shown that the proposed encoding method can serve as a staring point to develop a more efficient method of significance map encoding for a redundant representation.

7.1 Conclusions

Image compression is only one of the possible applications of sparse approximations. Image de-noising, restoration, in-painting and analysis can be named among others. Moreover, different types of signals can be modelled, such as sound, video, EEG, or even 3D scenes

and models. A suboptimal solution of the NP-complete combinatorial problem is of interest when searching for sparse approximation using a redundant dictionary. A wide range of heuristics, including greedy optimisation methods and linear programming can be applied. We compared a few distinct methods such as Orthogonal Matching Pursuit and Basis Pursuit and evaluated their usefulness for image compression. Some clear advantages of MP over other known methods were highlighted in Chapter 3. However its limitation to lossy compression only at low and medium bit-rates due to slow convergence was pointed out.

The idea of combining MP with other image transforms was studied in Chapter 4. A formulation of the main idea to perform MP in the spatio-frequency domain was presented. DWT and DCT were considered, as the examples of spatio-frequency transforms. In terms of sparse representation, i. e. minimising distortion for a fixed number of atoms, smooth regular wavelets such as CDF 9/7 filters were favoured over non-continuous Haar filters and block-based DCT. The benefits of performing decomposition of the zero-mean signal and symmetrical periodical extension to treat image borders were also recognised. Although performing MP on the full wavelet subbands gives the lowest distortion, it was found out that decomposition of fixed-size blocks can be preferred in practice due to reduced time complexity.

In Chapter 4 we looked at the problem of dictionary design for a hybrid MP and wavelet compression system. The problem was separated into two stages: choosing wavelets for the spatio-frequency representation and finding filters for the MP. To keep the complexity of the encoder tractable, the search of dictionaries was restricted to separable sets of short-support bases. The dictionaries were trained using a simple method of Basis Picking proposed for a similar framework in [77]. We compared trained dictionaries with randomly generated and analytically constructed with minimising a metric called coherence in mind. Coherence measures maximal similarity between atoms inside dictionary. We found that, for a hybrid MP and DWT codecs, a dictionary that minimises coherence can achieve a similar distortion to the dictionaries trained with the Basis Picking while maintaining significantly lower computational complexity due to the inclusion of shorter-support filters.

Multichannel MP was introduced in Chapter 3 as the promising method to represent multi-channel signals when the channels are highly correlated. In Chapter 4 we developed, for the first time, a representation of RGB images using multichannel MP performed in the spatio-frequency domain. All the findings about mean-shifting, choice of wavelets, border treatment, image partitioning and dictionaries from Chapters 3-4 transfer across from a single to multi-channel signals. Moreover, it has been verified that multi-channel MP with L_2-norm as optimisation criterion outperforms, in terms of PSNR, other norms, namely L_1 and L_∞, for image compression (Chapter 6). L_∞-norm minimisation was the criterion already tried in grayscale video coding [130]. We have shown that for decomposition of RGB images using the L_1 or L_2-norm gives significantly lower distortion.

Encoding the atomic decomposition into a bit-stream is the critical issue with the application of MP for image and video compression. A new idea for encoding coefficients obtained after a redundant transform was presented in Chapter 5. We generalised MERGE

from [78] by introducing a simple adaptive Run Length Encoding algorithm. A proposed method is particularly suited for encoding of the multi-channel decomposition and thus it was applied for colour image coding.

The transformed data are floating point numbers hence the quantisation method has to be designed together with the encoding. While analysing quantisation in Chapter 5 the importance of convergence was pointed out. Proofs of convergence of Quantised MP were adapted from from [118] for both single and multi-channel. Analysis of quantisation error for different parameters has been performed. Our findings were that when quantisation is performed in-loop then for single-channel images even a very coarse quantisation has little effect on distortion. For the quantisation parameter $PL = 2$, chosen for grayscale encoding, the difference in distortion is less than 0.1 dB when comparing to MP without quantisation for up to 8000 atoms. However, in the multi-channel case a difference of up to 0.4 dB is observed. This suggests that further work is required to find more efficient quantisation of colour amplitudes.

Two main ideas to represent RGB images using MP in the transformed domain were considered. First, multichannel MP performed directly on RGB images (MP-RGB) and second, single-channel MP in a decorrelated colour space such as $\mathrm{YC_bC_r}$ (MP-YCC). After comparing the two in terms of sparsity of the decomposition it has been noted in Chapter 4 that many fewer atoms are needed for the same distortion using MP-RGB with L_2-norm optimisation thus confirming the decorrelating potential of multi-channel MP. However, much more information is needed to be encoded for each colour atom in the case of MP-RGB than for each atom obtained by MP-YCC. In Chapter 6 it was noted that if the proposed encoding algorithm is applied then the MP-YCC can compete with the MP-RGB in terms of the coding performance.

In terms of the comparison to related coding standards including SPIHT and JPEG 2000, statistical t-tests performed on a set of standard images show no significant difference in R-D performance for up to 0.45 bpp in terms of the average PSNR. Visual comparisons suggest that MP represents some image patterns better. Investigation in the image features better represented by MP-like methods opens a list of possible future research directions in image coding suggested in Section 7.2. Moreover a proposed system is more general and flexible than standard wavelet methods on both transform and encoding stages, therefore its practical usefulness to other images types of signals should be also investigated.

7.2 Future Directions

Sparse approximations were recognised to be a potential step forward in the field of scalable lossy compression. Nevertheless, questions of the optimal dictionary, atom selection algorithm and efficient atom coding although extensively studied still remain open.

The method of encoding, proposed in Chapter 5, achieves compression performance comparable to wavelet-based standards. However, the standards use additional data modelling in the form of Spatial Orientation Trees (SOTs) in SPIHT and context modelling for arithmetic coding (BAC) in JPEG 2000. A proposed idea is a general algorithm that can

be used for any data that can be mapped into table of rows and, in our implementation, a uniform distribution for each column is assumed. A range of ideas to better model the data that form the atomic decomposition of an image and as a consequence improve coding performance include: improved position coding by exploiting their distribution with the use of ideas presented in Section 6.4 and modelling dictionary indexes distributions. Moreover, predicting amplitude values to improve colour quantisation creates an additional option for colour coding. Also, atom selection according to a criterion closer than MSE to human visual perception is possible. However the example shown in Section 6.1 demonstrates that care must be taken to preserve convergence of decomposition. For example, this is already a challenging task if we were to to optimise SSIM or HSSIM metrics introduced in Chapter 1, which could be an obvious step to improve the visual appearance of decompressed images.

It is always possible to search for more efficient dictionaries and wavelets, but it seems that a more sensible direction is to narrow the target domain to medical, astronomical, fingerprints, faces or some other specific class of images. For example, the single-channel MP with dictionaries trained using the K-SVD from [30] and its variations [132] was successfully applied for facial image compression especially at low bit-rates [8, 132]. The decorrelating properties of multi-channel MP profoundly widen the application of sparse approximations to decomposition of many images at the same time, hyper-spectral data and to explore both temporal and spatial directions in video coding. MP-like methods are suitable to be used for both inter (already successfully applied [83, 130]) and intra frames in modern video codecs such as H.264. Multichannel MP is an idea that can be applied in video coding in both spatial and temporal directions.

The drawback of MP, that stops it from being a widely accepted in image compression, is the computationally expensive decomposition process. Exploiting a dictionary structure and partitioning into blocks was suggested as a way to speed up the algorithm in Section 4.1. Finally, due to possibility of processing sub-bands (or blocks) separately and performing a lot of mathematical operations on the same data, the sub-band implementation developed in this work can be easily adapted to run on parallel architectures or Graphical Processors (GPU).

A Test Images

(a) Airplane, 512 × 512

(b) Baboon, 512 × 512

(c) Barbara, 720 × 576

(d) Goldhill, 720 × 576

(e) House, 768 × 512

(f) Lighthouse, 768 × 512

(g) Motocross, 768 × 512

(h) Parrots, 768 × 512

(i) Sailboats, 512 × 768

(j) Lenna, 512 × 512

(k) Peppers, 512 × 512

(l) Sailboat, 512 × 512

B Mathematical background

B.1 Terminology.

A *Linear space* is any set $\mathcal{H}$ with operations of addition $+$ and multiplication $\cdot$ by a scalar from the field $\mathcal{C}$ (we consider here the fields of real $\mathcal{C} = \mathbb{R}$ or complex $\mathcal{C} = \mathbb{C}$ numbers) that satisfies the following axioms for $f, g, h \in \mathcal{H}$ and $\alpha, \beta \in \mathcal{C}$:

1. $f + g = g + f$,
2. $(f + g) + h = f + (g + h)$,
3. $\exists_{\mathbf{0}\in\mathcal{H}} f + \mathbf{0} = f$,
4. $\forall_{f\in\mathcal{H}} \exists_{(-f)\in\mathcal{H}} f + (-f) = \mathbf{0}$, we shall use notation: $f - f = \mathbf{0}$,
5. $\exists_{\mathbf{1}\in\mathcal{C}} 1 \cdot f = f$,
6. $\alpha(\beta f) = (\alpha\beta)f$,
7. $(\alpha + \beta)f = \alpha f + \beta f$,
8. $\alpha(f + g) = \alpha f + \alpha g$.

An *inner product* can be defined in the linear space $\mathcal{H}$ as a function $\mathcal{H} \times \mathcal{H} \to \mathcal{C}$ such that for any $f, g, h \in \mathcal{H}$ and any $\alpha, \beta \in \mathcal{C}$ the following conditions are satisfied:

1. $\langle f, g\rangle = \overline{\langle g, f\rangle}$,

2. $\langle \alpha f + \beta g, h\rangle = \alpha\langle f, h\rangle + \beta\langle g, h\rangle$,

3. $\langle f, f\rangle \in \mathcal{R}, \langle f, f\rangle \geq 0$,

4. $\langle f, f\rangle = 0 \Rightarrow f = \mathbf{0}$.

We denote the complex conjugate of $z \in \mathbb{C}$ by $\overline{z}$, the real part of z by $\Re z$ and the absolute value by $|z|$. We recall the following properties of complex conjugation:

$$z + \overline{z} = 2\Re z, \text{ and } z\overline{z} = |z|^2. \tag{B.1}$$

In the linear space with inner product $\langle .,. \rangle$ we can define the *norm* $\|.\| : \mathcal{H} \to \mathcal{R}$ as:

$$\|f\| = \sqrt{\langle f, f\rangle}. \tag{B.2}$$

The function $\rho(f, g) = \|f - g\|$ induces a *metric* in the linear space which we refer to as the distance between f and g. The following well-known inequalities are satisfied:

$$|\langle f, g\rangle| \leq \|f\|\,\|g\| \text{ (Cauchy inequality)} \tag{B.3}$$

$$\|f + g\| \leq \|f\| + \|g\| \text{ (triangle inequality)} \tag{B.4}$$

$$\|f - g\| \geq |\|f\| - \|g\|| . \tag{B.5}$$

For each bounded subset of real numbers $D \subset \mathbb{R}$ there exists a real number $a = \sup D$ called the *supremum* that is the smallest number such that $\forall_{x\in D} a \geq x$. Analogously the *infimum* is the largest number $b = \inf D$ such that $\forall_{x\in D} b \leq x$. We shall use the notation $\sup_{x\in D} x$, which is the same as $\sup D$, and for indexed sets $\sup_{i\in I} x_i$,which is the same as $\sup\{x_i\}_{i\in I}$. The following properties of $\sup D$ will be used:

$$\forall_{i\in I} a_i < b_i \Rightarrow \sup_{i\in I} a_i \leq \sup_{i\in I} b_i \tag{B.6}$$

$$\sup_{i\in I}(a_i + b_i) \leq \sup_{i\in I} a_i + \sup_{i\in I} b_i. \tag{B.7}$$

A sequence $\{f_n\}$ converges to f_∞, called a *limit* in a metric space with metric $\|.\|$, if and only if

$$\forall_{\epsilon>0}\exists_N \forall_{n>N} \|f_n - f_\infty\| < \epsilon.$$

We shall write $\lim_{n\to\infty} f_n = f_\infty$, or just $f_n \to f_\infty$ as $n \to \infty$. For every bounded sequence we can choose a subsequence that converges. If we form a set F of all limits for all convergent sub-sequences then we can define the *lower limit* by $\liminf_{n\to\infty} f_n = \inf F$, and the *upper limit* by $\limsup_{n\to\infty} f_n = \sup F$.
A *Cauchy sequence* is a sequence f_n such that

$$\forall_{\epsilon>0}\exists_N \forall_{n,m>N} \|f_n - f_m\| < \epsilon.$$

A metric space in which every Cauchy sequence converges to an element from this space is called *complete*. A complete linear space with norm induced by the inner product is called a *Hilbert space*.
The following well known facts hold in the Hilbert space:

Lemma B.1.1. *If f_n is non-decreasing i.e. $\forall_{n>N} f_{n+1} \geq f_n$ and bounded from above then f_n converges. Similarly f_n converges if it is bounded from below and non-increasing.*

Lemma B.1.2. $f_n \to f \Rightarrow \|f_n\| \to \|f\|$.

Lemma B.1.3. $\|f_n\| \to 0 \Leftrightarrow f_n \to \mathbf{0}$.

Lemma B.1.4. $f_n \to f \Rightarrow \langle f_n, g\rangle \to \langle f, g\rangle$.

Lemma B.1.5. *Generalisation of Pythagoras theorem:*

$$\|f+g\|^2 = \|f\|^2 + \|g\|^2 + 2\Re\langle f, g\rangle. \tag{B.8}$$

Lemma B.1.6. *Polarisation inequality, for any $f, g \in \mathcal{H}$:*

$$\|f+g\|^2 + \|f-g\|^2 = 2\left(\|f\|^2 + \|g\|^2\right). \tag{B.9}$$

B.2 Matching Pursuit in Hilbert Space

Note that a_n is the inner product with residual at nth iteration while A_n the quantised values, we denote a quantisation error as: $\epsilon_n = A_n - a_n$ (see Chapter 5).

Lemma B.2.1. *Parseval-like equality for Matching Pursuit:*

$$\|f\|^2 = \sum_{n=1}^{N} |a_n|^2 + \left\|R^{N+1}f\right\|^2. \tag{B.10}$$

Proof. Equation (B.10) is a direct consequence of the MP update step (Algorithm 3.1):

$$\left\|R^{N+1}f\right\|^2 = \left\|R^N f - a_N g_{\gamma_N}\right\|^2 = \left\|R^N f\right\|^2 + |a_N|^2 - 2\Re\langle R^N f, a_N g_{\gamma_N}\rangle =$$

$$\left\|R^N f\right\|^2 + |a_N|^2 - 2\Re\left(\overline{a_N} a_N\right) = \left\|R^N f\right\|^2 - |a_N|^2.$$

Hence, for all N: $\left\|R^N f\right\|^2 = \left\|R^{N+1}f\right\|^2 + |a_N|^2$, which with $R^1 f = f$ inductively implies Equation (B.10). □

Lemma B.2.2. *Parseval-like equality for Quantised Matching Pursuit:*

$$\|f\|^2 = \sum_{n=1}^{N} \left(|a_n|^2 - |A_n - a_n|^2\right) + \left\|R^{N+1}f\right\|^2. \tag{B.11}$$

Proof. Here, the starting point is an update step with quantisation of the amplitude (see Section 5.1). We apply (B.1) and (B.8).

$$\left\|R^{N+1}f\right\|^2 = \left\|R^N f - A_N g_{\gamma_N}\right\|^2 = \left\|R^N f\right\|^2 + |A_N|^2 - 2\Re\langle R^N f, A_N g_{\gamma_N}\rangle =$$

$$\left\|R^N f\right\|^2 + |A_N|^2 - 2\Re\left(\overline{A_N} a_N\right) = \left\|R^N f\right\|^2 + |A_N - a_N|^2 - |a_N|^2.$$

From which we have, for all N: $\left\|R^N f\right\|^2 = \left\|R^{N+1}f\right\|^2 + |a_N|^2 - |A_N - a_N|^2$, which inductively implies Equation (B.11). □

B.3 Full proofs of convergence.

All proofs below are combination or direct translation of proofs that can be found in [72], [117], [118] and [67]. It can be shown that ideas from those proofs can be directly applied to Quantised MP.

Lemma B.3.1. *If* $\{s_n\}_{n=1,2,\dots}$ *is a positive sequence such that* $\sum_{n=1}^{+\infty} {s_n}^2 < \infty$ *then:*

$$\lim_{n\to\infty} \inf s_n \sum_{k=1}^{n} s_k = 0. \tag{B.12}$$

Proof. (see [72], Lemma 3 and [117], Lemma 2.3)
We choose n and k for any $\epsilon > 0$. In particular for a chosen sequence $\{\epsilon_N\}$ such as $\epsilon_N \to 0$ we have:

$$\sum_{i=1}^{\infty} s_i^2 < \infty \Rightarrow \sum_{i=n}^{\infty} s_n^2 \leq \frac{\epsilon_N}{2}.$$

Also

$$\sum_{i=1}^{\infty} s_i^2 < \infty \Rightarrow \lim_{i\to\infty} s_i^2 = 0 \Rightarrow \lim_{i\to\infty} s_i = 0 \Rightarrow s_k \sum_{i=0}^{k} s_i \leq \frac{\epsilon_N}{2}.$$

We can choose a subsequence $\{j_N\}$ such $s_{j_N} = \min_{\{i\in n+1,\dots,k\}} s_i$ and then:

$$s_{j_N} \sum_{k=0}^{j_N} s_k = s_{j_N} \sum_{k=0}^{n} s_k + s_{j_N} \sum_{k=n+1}^{j_N} s_k \leq \frac{\epsilon_N}{2} + \sum_{k=n+1}^{j_N} {s_k}^2 \leq \epsilon_N.$$

Hence, by definition of lower limit and the fact that s_n is a positive sequence (B.12) holds. □

Lemma B.3.2. *If* $R^N f$ *converges then it converges to* $\mathbf{0}$ *for both MP and QMP.*

Proof. This proof is an adaptation of Lemma 2.1 from [117] to the quantised version of the MP. Assume $R^N f \to R^\infty \neq \mathbf{0}$. $R^\infty \neq \mathbf{0}$ implies that there exists $\delta > 0$ and a dictionary element g_λ (completeness of the dictionary is assumed) such that:

$$|\langle R^\infty, g_\lambda \rangle| \geq 2\delta.$$

Therefore also (B.6):

$$\sup_{g_\lambda \in \mathcal{D}} |\langle R^\infty, g_\lambda \rangle| \geq 2\delta. \tag{B.13}$$

By Lemma B.1.4:

$$\forall_{g_\lambda \in \mathcal{D}} \langle R^N f, g_\lambda \rangle \to \langle R^\infty, g_\lambda \rangle.$$

By definition of the limit and properties (B.6) and (B.7) of supremum there exists M such that for all $N > M$ and any dictionary element g_λ:

$$\left| \sup_{g_\lambda \in \mathcal{D}} |\langle R^N f, g_\lambda \rangle| - \sup_{g_\lambda \in \mathcal{D}} |\langle R^\infty, g_\lambda \rangle| \right| \leq \sup_{g_\lambda \in \mathcal{D}} \left| |\langle R^N f, g_\lambda \rangle| - |\langle R^\infty, g_\lambda \rangle| \right| \leq \delta.$$

For the above inequality to be satisfied it must be by (B.13):

$$\sup_{g_\lambda \in \mathcal{D}} |\langle R^N f, g_\lambda \rangle| \geq \delta > 0,$$

Finally, we have for all $N > M$: $|a_N| = |\langle R^N f, g_{\gamma_N}\rangle| \geq \alpha \sup_{\gamma\in\mathcal{D}} |\langle R^N f, g_\gamma\rangle| \geq \alpha\delta$, and hence:

$$\left\|R^{N+1}f\right\|^2 = \|f\|^2 - \sum_{n=1}^{N}\left(|a_n|^2 - |\epsilon_n|^2\right) \leq$$
$$\|f\|^2 - (1-\theta^2)\sum_{n=1}^{N}|a_n|^2 \leq \|f\|^2 - N(1-\theta^2)\alpha^2\delta^2,$$

which implies that:

$$\|f\|^2 - \left\|R^{N+1}f\right\|^2 \geq N(1-\theta^2)\alpha^2\delta^2 \geq 0,$$

which is impossible as the terms $N(1-\theta^2)\alpha^2\delta^2 \to \infty$ as $N \to \infty$ while $\|f\|^2$ and $\left\|R^{N+1}f\right\|^2$ are bounded. Hence, if $R^N f$ converges it must converge to $\mathbf{0}$. □

Lemma B.3.3. *$R^N f$ converges for both MP and QMP.*

Proof. The proof uses the following lemma (Lemma 2.4 in [117]):

Lemma B.3.4. *If for all n and m there is: $\|x_n - x_m\|^2 = y_n - y_m + h_{n,m}$, and a sequence $\{y_n\}$ converges and*

$$\liminf_{m\to\infty}\max_{n<m} h_{n,m} = 0, \tag{B.14}$$

then $\{x_n\}$ also converges.

We need to prove that $R^N f$ is a Cauchy sequence. We consider for $N < M$:

$$\left\|R^N f - R^M f\right\|^2 = \left\|R^N f\right\|^2 - \left\|R^M f\right\|^2 - 2\langle R^N f - R^M f, R^M f\rangle.$$

Let denote

$$h_{N,M} = \left|\langle R^N f - R^M f, R^M f\rangle\right|.$$

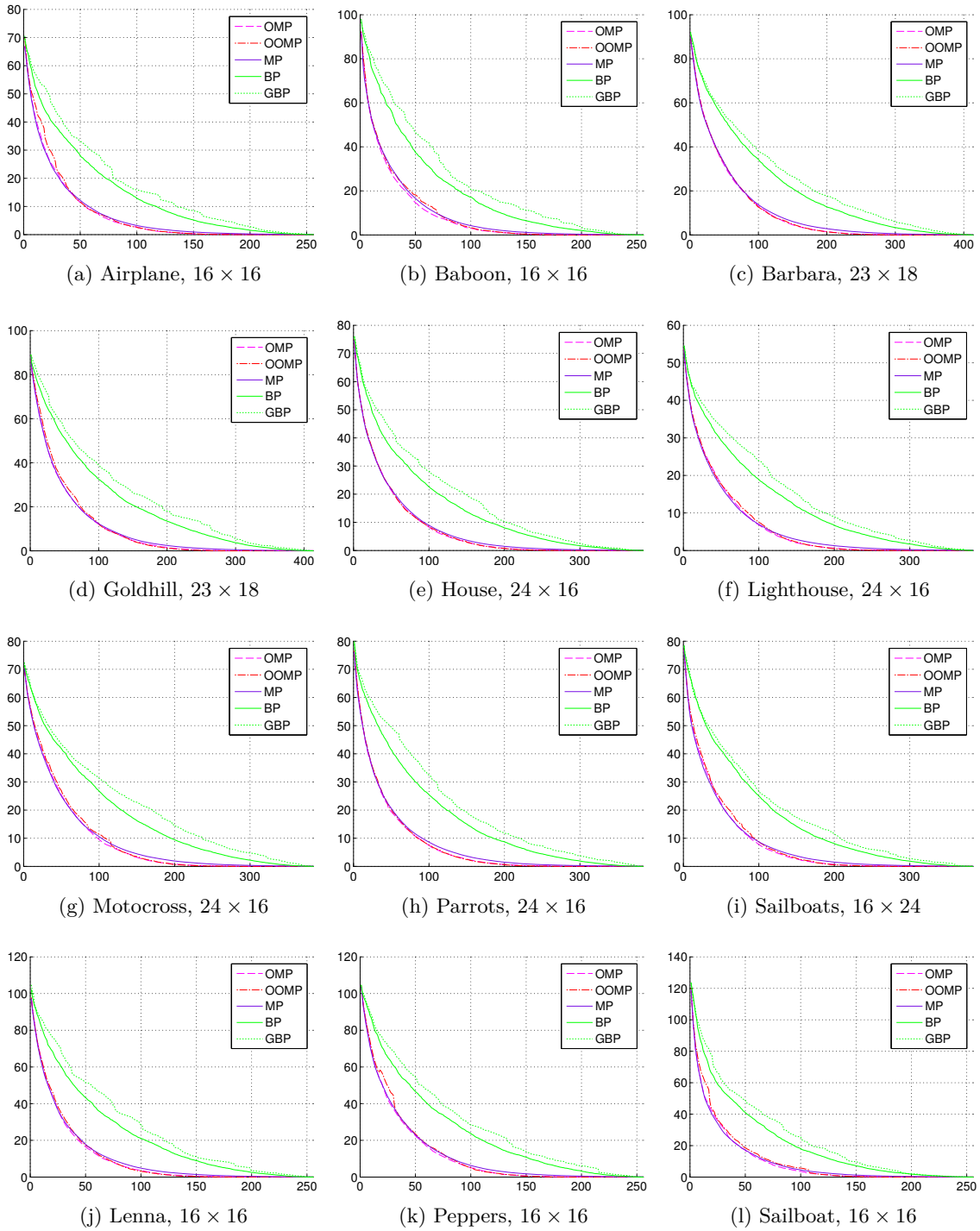
As $\left\|R^N f\right\|^2$ converges as a decreasing sequence of positive numbers we only need to prove Lemma B.3.4. The following can be derived for QMP based on the update step and inequality (5.7) from Chapter 5 using inner product properties and triangle inequality (B.4):

$$h_{N,M} = \left|\langle \sum_{j=N+1}^{M} A_j g_{\lambda_j}, R^M f\rangle\right| \leq |a_M| \sum_{j=N+1}^{M}|A_j| \leq (1+\theta)|a_M|\sum_{j=1}^{M}|a_j|.$$

It holds for all N and M such as $N < M$ so also for $\max_{N<M} h_{N,M}$. The right hand side satisfies Lemma B.3.1 which finishes the proof. □

C Comparison of Basis Selection Methods

Results for Dictionary $\mathcal{D}_{16}$.



(a) Airplane, 16 × 16 (b) Baboon, 16 × 16 (c) Barbara, 23 × 18

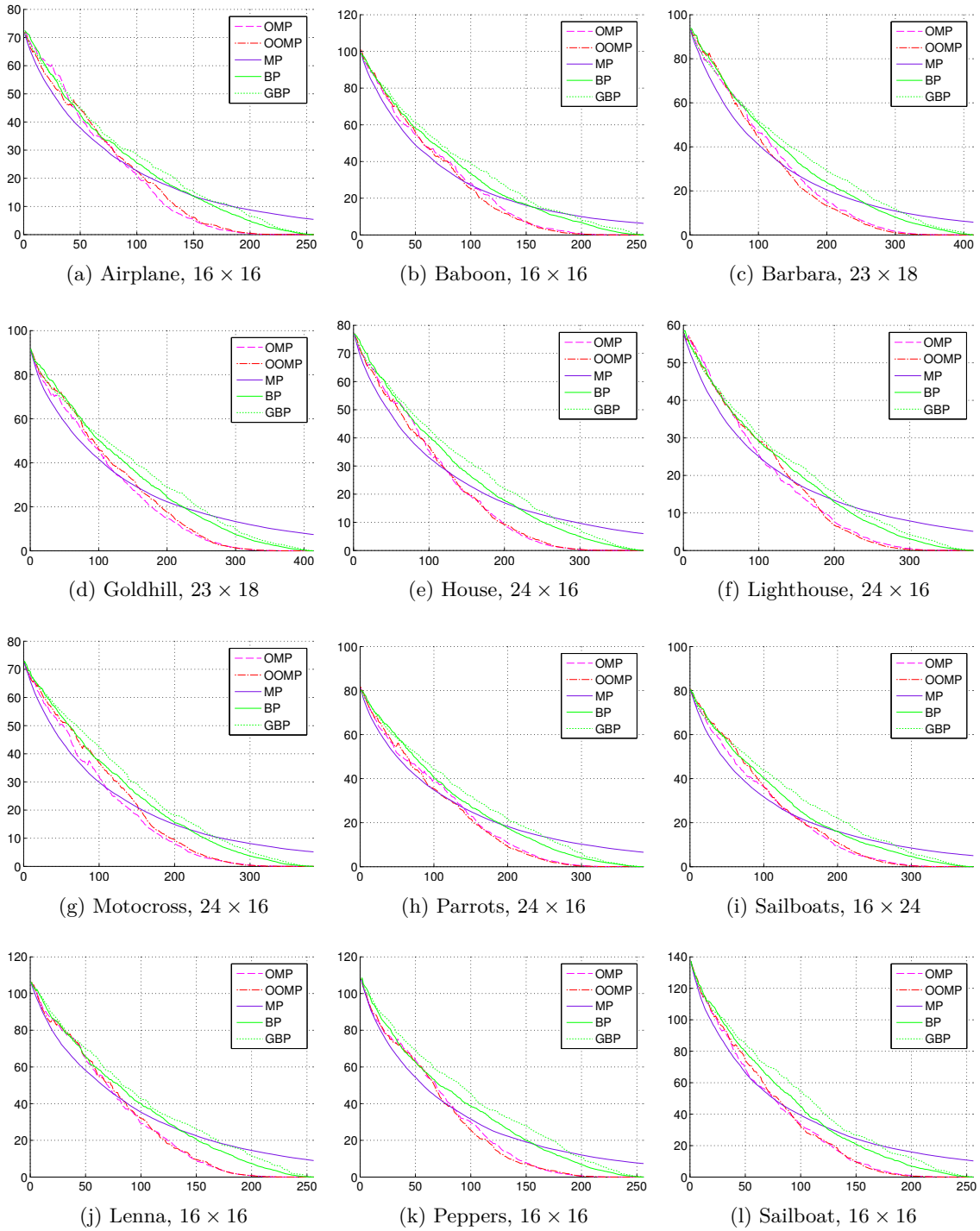
(d) Goldhill, 23 × 18 (e) House, 24 × 16 (f) Lighthouse, 24 × 16

(g) Motocross, 24 × 16 (h) Parrots, 24 × 16 (i) Sailboats, 16 × 24

(j) Lenna, 16 × 16 (k) Peppers, 16 × 16 (l) Sailboat, 16 × 16

Figure C.1: RMSE (y-axis) as a function of number of atoms (x-axis) for the lowest frequency subband for all test images and dictionary $\mathcal{D}_{16}$.

Results for Random Dictionary.



(a) Airplane, 16 × 16 (b) Baboon, 16 × 16 (c) Barbara, 23 × 18

(d) Goldhill, 23 × 18 (e) House, 24 × 16 (f) Lighthouse, 24 × 16

(g) Motocross, 24 × 16 (h) Parrots, 24 × 16 (i) Sailboats, 16 × 24

(j) Lenna, 16 × 16 (k) Peppers, 16 × 16 (l) Sailboat, 16 × 16

Figure C.2: RMSE (y-axis) as a function of number of atoms (x-axis) for the lowest frequency subbands for all test images and random uniform dictionary.

Bibliography

[1] O. K. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A. Zakhor. Video compression using matching pursuits. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(1):123–143, 1999.

[2] A. M. Amin. A novel approach for image compression using matching pursuit signal approximation and simulated annealing. Master's thesis, Cairo University, Giza, Egypt, 2005.

[3] M. Andrle and L. Rebollo-Neira. Improvement of orthogonal matching pursuit strategies by backward and forward movements. In *Proc. of International Conference on Acoustics Speech and Signal Processing*, volume 3, pages 480–495, 2006.

[4] M. Andrle, L. Rebollo-Neira, and E. Sagianos. Backward-optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 11(9):705–708, 2004.

[5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, 1992.

[6] H. B. Barlow. Redundancy reduction revisited. *Netw. Comput. Neural Syst.*, (12):241–253, 2001.

[7] F. Bergeaud and S. Mallat. Matching pursuit of images. In *Proc. of International Conference on Image Processing*, volume 1, pages 53–56, 1995.

[8] O. Bryt and M. Elad. Compression of facial images using the K-SVD algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–282, 2008.

[9] S.G. Chang, B. Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, 2000.

[10] S.G. Chang, B. Yu, and M. Vetterli. Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Transactions on Image Processing*, 9(9):1522–1530, 2000.

[11] S.G. Chang, B. Yu, and M. Vetterli. Wavelet thresholding for multiple noisy image copies. *IEEE Transactions on Image Processing*, 9(9):1631–1635, 2000.

[12] S. S. Chen and D. L. Donoho. Basis pursuit. In *Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 41–44, 1994.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[14] Chun-Hsien Chou and Kuo-Cheng Liu. Perceptually optimal JPEG2000 coding of color images. In *Proc. of Eighth IEEE International Symposium on Multimedia*, pages 549–556, 2006.

[15] Chun-Hsien Chou and Kuo-Cheng Liu. Colour image compression based on the measure of just noticeable colour difference. *IET Image Processing*, 2(6):304–322, 2008.

[16] Chun-Hsien Chou, Kuo-Cheng Liu, and Chien-Sheng Lin. Perceptually optimized JPEG2000 coder based on CIEDE2000 color difference equation. In *Proc. of International Conference on Image Processing*, volume 3, pages 1184–1187, 2005.

[17] C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG 2000 still image coding system: An overview. *IEEE Transactions on Consumer Electronics*, 46(4):1103–1127, 2000.

[18] C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, 2001.

[19] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising. In *Lecture Notes in Statistics*, volume 103, pages 125–150. Springer-Verlag, 1995.

[20] P. Czerepinski, C. Davies, N. Canagarajah, and D. Bull. Matching pursuits video coding: Dictionaries and fast implementation. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(7):1103–1115, 2000.

[21] F. Daly, D. J. Hand, M.C Jones, A. D. Lunn, and K. J. McConway. *Elements of Statistics*. Addison-Wesley Publishing Company, Wokingham, 1995.

[22] C. De Vleeschouwer and B. Macq. Subband dictionaries for low-cost matching pursuits of video residues. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):984–993, 1999.

[23] C. De Vleeschouwer and A Zakhor. In-loop atom modulus quantization for matching pursuit and its application to video coding. *IEEE Transactions on Image Processing*, 12(10):1226–1242, 2003.

[24] D. L. Donoho. Adaptive signal representations: How much is too much? In *Proc. of IEEE-IMS Workshop on Information Theory and Statistics*, page 53, 1994.

[25] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[26] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1305, 2006.

[27] David L. Donoho and I. M. Johnstone. Ideal denoising in an orthonormal basis chosen from a library of bases. *Comptes Rendus Acad. Sci., Ser. I*, 319:1317–1322, 1994.

[28] P. J. Durka, D.Ircha, and K. J. Blinowska. Stochastic time-frequency dictionaries for matching pursuit. *IEEE Transactions on Signal Processing*, 49(3):507–510, 2001.

[29] M. Elad. *Sparse and Redundant Representations*. Springer, 1st edition, 2010.

[30] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[31] M. Elad, M. A. T. Figueiredo, and Yi Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010.

[32] K. Engan, K. Skretting, and J. H Husøy. Denoising of images using designed signal dependent frames and matching pursuit. In *Proc. of International Conference on Acoustics Speech and Signal Processing*, volume 2, pages 653–656, 2005.

[33] M. D. Fairchild. *Color Appearance Models*. Wiley, 2nd edition, 2005.

[34] R. M. Figueras i Ventura. *Sparse image approximation with application to flexible image coding*. PhD thesis, EPFL, Lausanne, 2005.

[35] R. M. Figueras i Ventura, O. Divorra Escoda, and P. Vandergheynst. A matching pursuit full search algorithm for image approximations. Technical report, EPFL, 2004.

[36] R. M Figueras i Ventura, P. Vandergheynst, and P. Frossard. Low-rate and flexible image coding with redundant representations. *IEEE Transactions on Image Processing*, 15(3):726–739, 2006.

[37] R. M. Figueras i Ventura, P. Vandergheynst, P. Frossard, and A. Cavallaro. Color image scalable coding with matching pursuit. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 53–56, 2004.

[38] P. Frossard, P. Vandergheynst, R. M. Figueras i Ventura, and M. Kunt. A posteriori quantization of progressive matching pursuit streams. *IEEE Transactions on Signal Processing*, 52(2):525–535, 2004.

[39] E. Gershikov, E. Lavi-Burlak, and M. Porat. Correlation-based approach to color image compression. *Image Communications*, 22(9):719–733, 2007.

[40] E. Gershikov and M. Porat. On color transforms and bit allocation for optimal subband image compression. *Image Communications*, 22(1):1–18, 2007.

[41] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 5th edition, 1992.

[42] M. Ghanbari. *Standard Codecs: Image Compression to Advanced Video Coding*. The Institution of Engineering and Technology, illustrated edition, 2003.

[43] L. Goffman-Vinopal and M. Porat. Color image compression using inter-color correlation. In *Proc. of International Conference on Image Processing*, volume 2, pages 353–356, 2002.

[44] S. W. Golomb. Run-length encodings. *IEEE Transactions on Information Theory*, 12(3):399–401, 1966.

[45] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2007.

[46] V. K. Goyal, M. Vetterli, and N. T. Thao. Quantized overcomplete expansions in $\Re^N$: analysis, synthesis, and algorithms. *IEEE Transactions on Information Theory*, 44(1):16–31, 1998.

[47] D. J. Graham and D. J. Field. Efficient coding of natural images. *New Encyclopedia of Neuroscience*, 2007.

[48] M. Grgic, M. Ravnjak, and B. Zovko-Cihlar. Filter comparison in wavelet transform of still images. In *Proc. of International Symposium on Industrial Electronics*, volume 1, pages 105–110, 1999.

[49] S. Grgic, M. Grgic, and B. Zovko-Cihlar. Performance analysis of image compression using wavelets. *IEEE Transactions on Industrial Electronics*, 48(3):682–695, 2001.

[50] P. Hao and Q. Shi. Comparative study of color transforms for image coding and derivation of integer reversible color transform. In *Proc. of International Conference on Pattern Recognition*, volume 3, pages 224–227, 2000.

[51] Y. He, T. Gan, and H. J. Wang. Efficient matching pursuit image coding based on block partitioning. *Electronics Letters*, 45(14):733, 2009.

[52] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[53] P. S. Huggins and S. W. Zucker. Greedy basis pursuit. *IEEE Transactions on Signal Processing*, 55(7):3760–3772, 2007.

[54] K. Imamura, Y. Koba, and H. Hashimoto. A fast matching pursuits algorithm using sub-band decomposition of video signals. In *Proc. of International Conference on Multimedia and Expo*, number 4, pages 729–732, 2006.

[55] ITU-CCIT. Digital compression and coding of continous-tone still images - Requirements and guidelines, ITU-CCIT recommendation T.81, 1992.

[56] A. K. Jain. A sinusoidal family of unitary transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(4):356–365, 1979.

[57] B. Jeon and S. Oh. Fast matching pursuit with vector norm comparison. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(4):338–342, 2003.

[58] A. A. Kassim and W. S. Lee. Embedded color image coding using SPIHT with partially linked spatial orientation trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(2):203–206, 2003.

[59] E. Khan and M. Ghanbari. Efficient SPIHT-based embedded colour image coding techniques. *Electronics Letters*, 37(15):951–952, 2001.

[60] Beong-Jo Kim, Zixiang Xiong, and W. A. Pearlman. Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT). *IEEE Transactions on Circuits and Systems for Video Technology*, 10(8):1374–1387, 2000.

[61] Hyun Mun Kim, Woo-Shik Kim, and Dae-Sung Cho. A new color transform for RGB coding. In *Proc. of International Conference on Image Processing*, volume 1, pages 107–110, 2004.

[62] N. Kingsbury and T. Reeves. Redundant representation with complex wavelets: How to achieve sparsity. In *Proc. of International Conference on Image Processing*, volume 1, pages 45–48, 2003.

[63] N. G. Kingsbury and T. Reeves. Iterative image coding with overcomplete complex wavelet transforms. In *Society of Photo-Optical Instrumentation Engineers Conference Series*, volume 5150, pages 1253–1264, 2003.

[64] Nick Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Applied and Computational Harmonic Analysis*, 10(3):234–253, 2001.

[65] D. T. Lee. JPEG 2000: Retrospective and new developments. *Proceedings of the IEEE*, 93(1):32–41, 2005.

[66] D. Lemire and O. Kaser. Reordering columns for smaller indexes. *Information Sciences*, 181(12):2550–2570, 2011.

[67] A. Lutoborski and V. N. Temlyakov. Vector greedy algorithms. *Journal of Complexity*, 19(4):458–473, 2003.

[68] R. Maciol, Y. Yuan, and I. T. Nabney. Colour image coding with matching pursuit in the spatio-frequency domain. In *Proc. of International Conference on Image Analysis and Processing*, volume 1, pages 306–317, 2011.

[69] R. Maciol, Y. Yuan, and I. T. Nabney. Grayscale and colour image codec based on matching pursuit in the spatio-frequency domain. Technical report, Aston University, availabale at: http://eprints.aston.ac.uk/15194/, 2011.

[70] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.

[71] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way.* Academic Press, 3 edition, 2009.

[72] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[73] A. Moffat and L. Stuiver. Binary interpolative coding for effective index compression. *Inf. Retr.*, 3:25–47, 2000.

[74] A. Moinuddin, E. Khan, and M. Ghanbari. An efficient wavelet based embedded color image coding technique using block-tree approach. In *Proc. of International Conference on Image Processing*, pages 1889–1892, 2006.

[75] A. Moinuddin, E. Khan, and M. Ghanbari. Low complexity, efficient and embedded color image coding technique. *IEEE Transactions on Consumer Electronics*, 54(2):787–794, 2008.

[76] D. M. Monro. Basis picking for matching pursuits image coding. In *Proc. of International Conference on Image Processing*, volume 4, pages 2495–2498, 2004.

[77] D. M. Monro and W. Poh. Improved coding of atoms in matching pursuits. In *Proc. of International Conference on Image Processing*, volume 3, pages 759–62, 2003.

[78] D. M. Monro, W. Xiaopeng, H. Wei, and A. N. Evans. Merge coding of atoms for wavelet/matching pursuits image compression. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 1053–1056, 2007.

[79] D. M. Monro and Y. Yuan. Bases for low complexity matching pursuits image coding. In *Proc. of International Conference on Image Processing*, volume 2, pages 249–252, 2005.

[80] V. K. Nath, D. Hazarika, and A. Mahanta. A 3-D block transform based approach to color image compression. In *Proc. of Region 10 Conference*, pages 1–6, 2008.

[81] V. K. Nath, D. Hazarika, and A. Mahanta. A novel approach to color image compression using 3-D Discrete Cosine Transform (DCT). In *Proc. of Workshop on Machine Learning for Signal Processing*, pages 205–210, 2008.

[82] R. Neff. *New Methods for Matching Pursuit Video Compression.* PhD thesis, University of California at Berkeley, 2000.

[83] R. Neff and A. Zakhor. Very low bit-rate video coding based on matching pursuits. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):158–171, 1997.

[84] R. Neff and A. Zakhor. Modulus quantization for matching-pursuit video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(6):895–912, 2000.

[85] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[86] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997.

[87] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 40–44, 1993.

[88] 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*, 14(11):1219–1235, 2004.

[89] M. Pedersen and J. Y. Hardeberg. Survey of full-reference image quality metrics. Technical Report 5, Høgskolen i Gjøviks rapportserie, 2009.

[90] Soo-Chang Pei and Jian-Jiun Ding. Reversible integer color transform with bit-constraint. In *Proc. of International Conference on Image Processing*, volume 3, pages 964–967, 2005.

[91] Soo-Chang Pei and Jian-Jiun Ding. Reversible integer color transform. *IEEE Transactions on Image Processing*, 16(6):1686–1691, 2007.

[92] N. Ponomarenko, F. Battisti, K. Egiazarian, J. Astola, and V. Lukin. Metrics performance comparison for color image database. *Fourth international workshop on video processing and quality metrics for consumer electronics*, 2009.

[93] Ch. Poynton. *Digital video and HDTV: Algorithms and interfaces*. Elsevier, 3rd edition, 2003.

[94] W. Pratt. Spatial transform coding of color images. *IEEE Transactions on Communication Technology*, 19:980–992, 1971.

[95] K. R. Rao and Yip P. C. *The Transform and Data Compression Handbook*. CRC Press, 2001.

[96] L. Rebollo-Neira. Highly nonlinear approximations for sparse signal representation. *http://nonlinear-approx.info/*. Last access: 2012.03.01.

[97] L. Rebollo-Neira. Oblique matching pursuit. *IEEE Signal Processing Letters*, 14(10):703–706, 2007.

[98] L. Rebollo-Neira and D. Lowe. Optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 9(4):137–140, 2002.

[99] 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*, 6(3):243–250, 1996.

[100] Xing San, Hua Cai, and Jiang Li. Color image coding by using inter-color correlation. In *Proc. of International Conference on Image Processing*, pages 3117–3120, 2006.

[101] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6):123–151, 2005.

[102] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, 1993.

[103] G. Sharma and H. J. Trussell. Digital color imaging. *IEEE Transactions on Image Processing*, 6(7):901–932, 1997.

[104] H. R. Sheikh, M. F. Sabir, and A. C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451, 2006.

[105] K. Shen and E. J. Delp. Color image compression using an embedded rate scalable approach. In *Proc. of International Conference on Image Processing*, volume 3, pages 34–37, 1997.

[106] Y. Shi, Y. Ding, R. Zhang, and J. Li. Structure and hue similarity for color image quality assessment. In *Proc. of International Conference on Electronic Computer Technology*, pages 329–333, 2009.

[107] A. Shoa and S. Shirani. Adaptive rate-distortion optimal in-loop quantization for matching pursuit. *IEEE Transactions on Image Processing*, 17(9):1616–1623, 2008.

[108] M. Siotani. Order statistics for discrete case with a numerical application to the binomial distribution. *Annals of the Institute of Statistical Mathematics*, 8:95–104, 1956.

[109] K. Skretting and K. Engan. Image compression using learned dictionaries by rls-dla and compared with k-svd. In *Proc. of International Conference on Acoustics Speech and Signal Processing*, pages 1517–1520, 2011.

[110] Alvy Ray Smith. Color gamut transform pairs. *SIGGRAPH Comput. Graph.*, 12(3):12–19, 1978.

[111] SPIHT. *http://www.cipr.rpi.edu/research/SPIHT/*. Last access: 2009.09.10.

[112] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet - sRGB. 1996. Version 1.10.

[113] D. Taubman. Kakadu JPEG 2000. *http://www.kakadusoftware.com/*. Last access: 2009.11.10.

[114] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170, 2000.

[115] ISO/IEC 15 444-1: Information Technology. JPEG 2000 Image Coding System - Part 1: Core Coding System, 2000.

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

[117] V. N. Temlyakov. Weak greedy algorithms. *Advances in Computational Mathematics*, 12(2-3):213–227, 2000.

[118] V. N. Temlyakov. Nonlinear methods of approximation. *Foundations of Computational Mathematics*, 3(1):33–107, 2003.

[119] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[120] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[121] G. K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):18–34, 1992.

[122] Z. Wang and A. C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.

[123] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.

[124] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error measurement to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.

[125] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, CA, 2nd edition, 1999.

[126] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.

[127] Chun-Ling Yang, Lai-Man Po, Chun-Ho Cheung, and Kwok-Wai Cheung. A novel ordered-SPIHT for embedded color image coding. In *Proc. of International Conference on Neural Networks and Signal Processing*, volume 2, pages 1087–1090, 2003.

[128] J. Yang, Y. Wang, W. Xu, and Q. Dai. Image Coding Using Dual-Tree Discrete Wavelet Transform. *IEEE Transactions on Image Processing*, 17(9):1555–1569, 2008.

[129] Y. Yuan, A. N. Evans, and D. M. Monro. Low complexity separable matching pursuits [video coding applications]. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 725–728, 2004.

[130] Y. Yuan and D. M. Monro. 3D wavelet video coding with replicated matching pursuits. In *Proc. of International Conference on Image Processing*, volume 1, pages 69–72, 2005.

[131] Y. Yuan and D. M. Monro. Improved matching pursuits image coding. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 201–204, 2005.

[132] J. Zepeda, C. Guillemot, and E. Kijak. Image compression using the iteration-tuned and aligned dictionary. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 793–796, 2011.