MATCHING PURSUIT AND RESIDUAL VECTOR QUANTIZATION

MATCHING PURSUIT AND RESIDUAL VECTOR QUANTIZATION: APPLICATIONS IN IMAGE CODING

By

ABBAS EBRAHIMI-MOGHADAM, B.Sc., M.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (2008) MCMASTER UNIVERSITY

(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: **Matching Pursuit and Residual Vector Quantization: Applications in Image Coding**

AUTHOR: Abbas Ebrahimi-Moghadam

B.Sc., M.Sc.

SUPERVISOR: Dr. Shahram Shirani

NUMBER OF PAGES: xxiii, 137

To my wife Homa, my son Arad
my mother, and the memory of my father

Abstract

In this thesis, novel progressive scalable region-of-interest (ROI) image coding schemes with rate-distortion-complexity trade-off based on residual vector quantization (RVQ) and matching pursuit (MP) are developed. RVQ and MP provide the encoder with multi-resolution signal analysis tools, which are useful for rate-distortion trade-off and can be used to render a selected region of an image with a specific quality. An image quality refinement strategy is presented in this thesis, which improves the quality of the ROI in a progressive manner. The reconstructed image can mimic foveated images in perceptual image coding context. The systems are unbalanced in the sense that the decoders have less computational requirements than the encoders. The methods also provide interactive way of information refinement for regions of image with receiver's higher priority. The receiver is free to select multiple regions of interest and change his/her mind and choose alternative regions in the middle of signal transmission.

The proposed RVQ and MP based image coding methods in this thesis raise a couple of issues and reveal some capabilities in image coding and communication. In RVQ based image coding, the effects of dictionary size, number of RVQ stages and the size of image blocks on the reconstructed image quality, the resulting bit rate, and the computational complexity are investigated. The progressive nature of the resulting bit-stream makes RVQ and MP based

image coding methods suitable platforms for unequal error protection.

Researchers have paid lots of attention to joint source-channel (JSC) coding in recent years. In this popular framework, JSC decoding based on residual redundancy exploitation of a source coder output bit-stream is an interesting bandwidth efficient approach for signal reconstruction. In this thesis, we also addressed JSC decoding and error concealment problem for matching pursuit based coded images transmitted over a noisy memoryless channel. The problem is solved on minimum mean squared error (MMSE) estimation foundation and a suboptimal solution is devised, which yields high quality error concealment with different levels of computational complexity. The proposed decoding and error concealment solution takes advantage of the residual redundancy, which exists in neighboring image blocks as well as neighboring MP analysis stages, to improve the quality of the images with no increase in the required bandwidth. The effects of different parameters such as MP dictionary size and number of analysis stages on the performance of the proposed soft decoding method have also been investigated.

Acknowledgements

Like most major achievements in life, the completion of this thesis would not have been possible without the support and guidance of mentors, family, friends and colleagues. I would like to thank all of them.

First and foremost, I would like to express my deep appreciations to my wife, Homa, for her unconditional love, support, and patience. If she had not encouraged me to pursue my PhD, this thesis would not have been written. It is my duty and greatest pleasure to thank my mother for her lifelong love and support and to dedicate this work to the memory of my father who taught me determination and perseverance in life. In addition to my family, I would like to express my sincere gratitude to my supervisor Dr. Shahram Shirani for the expert guidance he gave and the confidence he showed in me throughout the long course of this research work. His encouragement, patience, and support during the years of my study in McMaster University will always be remembered. I am also grateful to Professor David W. Capson, Professor Xiaolin Wu, and Dr. Christopher Anand, members of my supervisory committee, for their valuable feedbacks. Also, I would like to acknowledge ECE department staves, specially Ms. Cheryl Gies and Ms. Helen Jachna for their cares and tireless administrative helps and Mr. Terry Greenlay and Mr. Cosmin Coroiu for their expert technical support.

List of Acronyms

2D	Two Dimensional
AWGN	Additive White Gaussian Noise
BEP	Bit Error Probability
BER	Bit Error Rate
bpp	bits per pixel
BCH	Bose-Chaudhuri-Hocquenghem (codes)
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
CD	Compact Disc
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation [12]
DVD	Digital Versatile Disc
EZW	Embedded Zero-tree Wavelet
dB	decibel(s)
GF	Galois Field
GLA	Generalized Lloyd Algorithm
H.263	ITU-T video coding standard [13]

HVS	Human Visual System
JORVQ	Jointly Optimized Residual Vector Quantization
JPEG	Joint Photographic Experts Group [2]
JPEG-2000	Image coding standard by JPEG [3]
JPIP	JPEG-2000 Interactive Protocol
JSC	Joint Source-Channel (coding/decoding)
LBG	Linde-Buzo-Gray (VQ codebook design algorithm)
MAP	Maximum *A-Posteriori* (estimation)
MP	Matching Pursuit
MPEG	Moving Picture Expert Group [14]
MRVQ	Mean Removed Vector Quantization
MSE	Mean Squared Error
MMSE	Minimum Mean Squared Error
PDF	Probability Density Function
PNN	Pairwise Nearest Neighbor
PSNR	Peak Signal to Noise Ratio
ROI	Region of Interest
RS	Reed-Solomon (coding)
RVQ	Residual Vector Quantization
SBC	Sub-Band Coding
SNR	Signal to Noise Ratio
SPIHT	Set Partitioning In Hierarchial Trees
UEP	Unequal (Uneven) Error Protection
VQ	Vector Quantization
VLC	Variable Length Coding

List of Notations and Symbols

$\|\cdot\|$	Euclidian norm of a vector
$\lceil x \rceil$	the smallest integer larger or equal to x
$\lfloor x \rfloor$	the largest integer smaller or equal to x
$\langle \mathbf{X}, \mathbf{Y} \rangle$	inner product of two vectors $\mathbf{X}$ and $\mathbf{Y}$
Γ	set of all MP dictionary indices
ϕ	phase shift parameter
$\sigma_{C^{(n)}}$	standard deviation of inner product coefficients at n^{th} MP analysis stage
τ	delay parameter
ξ	frequency modulation parameter
B_D	number of binary digits representing dictionary indices
B_Q	number of binary digits representing quantized values
$\mathcal{C}$	codebook set
$C^{(i)}$	inner product coefficient associated to i^{th} MP stage
$\tilde{C}^{(i)}$	scalar quantized version of $C^{(i)}$
$\mathscr{D}$	MP dictionary
$D^{(r)}$	average distortion in r^{th} LBG iteration
$d_j^{(i)}$	the distance of j^{th} image vector from the i^{th} ROI center

$\mathrm{E}[\cdot]$	expectation operator
f_c	cut-off frequency
$\mathbf{g}_\gamma$	γ^{th} element of MP dictionary
$\mathbf{g}_{I^d}$	MP dictionary element associated with I^d index
$H(i,j)$	Hamming distance between i and j
$\mathscr{I}$	set of codebook or dictionary indices
$\underline{I}$	a sequence of indices
I_n^a	index associated with average value of n^{th} image block
$I_n^{c(r)}$	index representing an MP inner product
	coefficient for r^{th} MP stage associated with n^{th} image block
$I_n^{d(r)}$	index representing an MP dictionary element
	for r^{th} MP stage associated with n^{th} image block
I_n	index associated with n^{th} image block
I^r	index representing residual vector
IQ_i	i^{th} stage inverse quantizer
J	received index from channel
L	vector dimension
$L^2(\mathbb{R})$	finite energy functions
N_B	number of image block in an image
N_c	the constraint number of paths in the JORVQ analysis
N_D	number of elements in shrunk MP dictionary
$N_{\mathscr{D}}$	number of elements in the original huge MP dictionary
N_Q	number of quantization steps
N_{ROI}	number of ROIs

N_s	maximum available RVQ or MP stages for signal analysis
p_e	probability of error
P_{ROI}	number of refinement stages of point of interest
	before background information starts being refined
Q_i	i^{th} stage quantizer
$\mathbb{R}^L$	L-dimensional Euclidian space
$\mathcal{R}^{(r)}\mathbf{x}$	r^{th} order residual vector resulted by MP analysis
R_k	radius of virtual circle at k^{th} stage of ROI image coding refinement
s	scaling parameter in time-frequency MP dictionaries
$\mathcal{T}$	VQ training set
u	translation parameter
V_i	i^{th} Voronoi cell
V	viewing distance
(x_f, y_f)	coordinates of fixation point
$\mathbf{X}$	signal vector
$\tilde{\mathbf{X}}$	estimated version of $\mathbf{X}$
$\mathbf{X}_{MR}$	mean removes version of vector $\mathbf{X}$
$\mathbf{X}_n$	n^{th} image block in a raster scan
$\mathbf{X}_{n<r>}$	for $r = 1, 2, 3, 4$ represent image blocks
	over, on left, right and under $\mathbf{X}_n$, respectively
$\tilde{\mathbf{X}}$	VQ or MP reconstructed version of $\mathbf{X}$
$\mathbf{Y}_i$	i^{th} VQ codebook element
$\tilde{\mathbf{Y}}_n^{(i)}$	The selected dictionary index in i^{th} stage of MP analysis of $\mathbf{X}$

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We are primarily visual creatures and receive most of the information about our surroundings by our eyes. We do not tolerate vision defects and try to fix them by every possible means such as spectacles and laser treatments, while we may accept some level of hearing loss before using hearing aid tools and we cannot do much about imperfections in other senses of taste, smell, and touch. Capturing memorable or important events by taking photos has now become commonplace. We often judge the success of deep space missions by the quality of returned images. This bias toward senses in our daily life is extended to how we try to perceive or "imagine" technical results. Today, scientific or technical instruments, make images to communicate with their operators rather than making sounds or emitting smells! We also try to extend the range of our natural vision by using microscopes or telescopes and go beyond the visible portion of the electromagnetic spectrum and take images of what used to be invisible. This partially explains why communication of visual signals is now so important.

For centuries, human's need to communicate has driven growing advancements in communication methods from early smoke signals to recent YouTube.

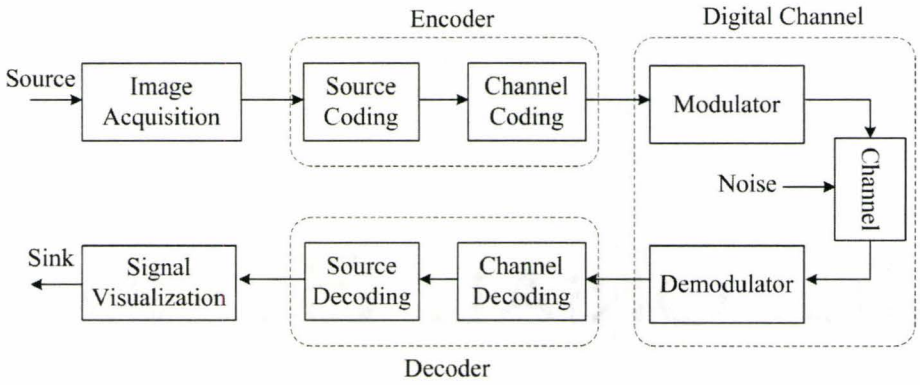


Figure 1.1: Digital communication of a signal in image form

We now live at the age of information and seek communication developments more than ever. Nowadays, new inventions in information technology have given an unprecedented access to the ever growing knowledge, entertainment and business contents. New inventions improve existing methods and open doors to new technologies. The goal of this thesis is to address some issues in image communication systems by considering two different multi-stage signal analysis methods as image coding basis.

1.1 Image Communication Overview

Fig. 1.1 displays a schematic block diagram of a typical digital image communication system. The signal is initially captured in image form by a proper *image acquisition* tool whose characteristics are closely related to the nature of the signal. A photo camera is the most apparent example of image acquisition tools. Images can also be acquired, for example, by computer graphics, medical diagnosis instruments, and scientific measuring tools. For digital communication, the image has to be in a digital form. A digital image is regarded as a two-dimensional signal made up of an array of $M \times N$ samples

or pixels. As the smallest individual element in a digital image, a pixel holds quantized value of grey level in grey-scale images or quantized values that represent colors in color images[†].

The encoder part in Fig. 1.1 is responsible for transforming the digital image into a bit-stream so that the signal can be efficiently and reliably delivered by the digital channel. According to information theory fundamentals, the pure essence or the existing information in a signal can be measured by the level of uncertainty in it, namely entropy. The *source encoder* or data compressor, attempts to remove all of the redundancies and tries to deliver this pure information to the next stage in a sequence of binary digits. Most of the source encoders, however, leave some redundancies due to technical difficulties or imperfect knowledge about the source model. There are two different types of data compression. Lossless compression, which removes redundancies of a signal in such a way that the reconstructed signal is identical to the original one. Huffman coding and arithmetic coding [1] are examples of lossless compression methods. Lossy compression methods, on the other hand, tolerate some loss of fidelity in the reconstructed signal and yield more signal compaction in return. Lossy compression is also regarded as perceptual coding, since it exploits how we perceive the signal, and removes less important information. For example, our eyes are not sensitive to very subtle color variations in an image and color information can be rounded off without any perceptual effect. In some cases, signal fidelity[‡] in a lossy compression is noticeably sacrificed to

[†]The discussion in this thesis is limited to grey-scale images, however, it is not a difficult task to extend it to color images.

[‡]There are different ways to compare the reconstructed image with the original signal and measure fidelity. The most common fidelity measure in image communication is called peak signal to noise ratio (PSNR). PSNR is defined as PSNR (dB) $= 10 \log_{10} \frac{x_{\max}^2}{\text{MSE}}$, where $x_{\max}$ is the maximum possible value of signal samples and MSE is the mean-squared error between the original signal sequence $\mathbf{X} = (x_1, x_2, \ldots, x_L)$ and the reconstructed one $\mathbf{Y} = (y_1, y_2, \ldots, y_L)$, i.e., MSE $= \frac{1}{L} \sum_{i=1}^{L} (x_i - y_i)^2$.

effectively reduce the amount of transferable data. Transform coding, such as DCT or wavelet transform coding, followed by quantization is a common lossy compression approach and is used in JPEG [2] and JPEG-2000 [3] standards. In addition to the compression quality, the source coding building block is involved with some other remarkable issues. Some source coding schemes provide *scalability*. In image coding context, scalability is a desirable property of the encoded bit-stream that enables the image communication system to support various channel bandwidth constraints or receiver's quality needs. The signal bit-stream in a quality scalable image encoder provides progressive quality improvements and successively refines the reconstructed image. An image encoder may also enable region of interest (ROI) coding, in which the viewer's most important regions of the image have higher transmission priority.

The *channel coder* in Fig. 1.1 transforms the source coder output into a sequence of symbols to combat the effect of the noisy transmission channel. In most instances, the channel coder output symbols are also binary digits. Unlike source coding, channel coding is involved with expanding the size of the transferable data to control decoding errors. A very simple, but unpopular, example of channel coding is to send every symbol r times with the hope that the symbol is received correctly most of the time. There are, of course, more sophisticated and reliable channel coding approaches such as, linear block coding, convolutional coding, and turbo coding [4].

The *modulator* converts the channel coder symbols into waveforms that are proper for transmission through the channel. The channel conveys the sequence of waveforms and imposes various noise disturbances. The *demodulator* processes the received waveforms and produces a sequence of symbols for *channel decoder*. The last three communication building blocks, i.e. modulator,

channel, and demodulator, are often regarded as digital channel in the literature. The channel decoder receives the possibly corrupted symbols and, based on the channel coding strategy and the noise characteristics of the channel, transforms them into a binary sequence called estimated information sequence. Finally, *source decoder* transforms this estimated sequence into an estimate of source coder output and delivers this estimate to image visualization unit.

In this thesis, some capabilities of two different signal analysis methods, namely residual vector quantization (RVQ) and matching pursuit (MP), in digital image communication systems are investigated. These two methods have similar residual and layered structures which suit quality scalability and ROI coding. By choosing each of these two signal analysis methods for progressive and ROI image coding, several issues in the image communication system of Fig. 1.1 have to be addressed. The source coder characteristics, the channel coding issue, and the decoding of the noise effected bit-stream are the main topics in this thesis.

1.2 Overview of the Thesis

1.2.1 Motivation

Data compression schemes operate more efficiently when they take the models of both the source and the receiver into account. Perceptual coding is a lossy compression approach, which takes advantage of discarding irrelevant imperceptible information in order to achieve high compression levels. Image and video compression based on *foveation*, modeling of the fovea of the human eye, is an example of perceptual coding. The works in this thesis were motivated by the author's primary research on foveation and the human visual

system (HVS) model. Foveation can be regarded as one of the natural vision system actions while sending visual information from the retina to the brain, which provides multi-resolution perception of an image. When we look at a picture, we do not see the image with uniform details and the image resolution is highest at the point of gaze and falls quickly toward the periphery. After examining different methods for video foveation and implementing a multi-resolution method based on filter banks for foveated H.263 video coding standard, which is explained briefly in appendix A, the author searched for other ways of implementing the more general idea of region of interest image coding.

Residual vector quantization (RVQ) and matching pursuit (MP) are two methods that can provide multi-resolution image representation. Region of interest image coding was the initial goal of using these signal representation methods, however, research progress revealed new aspects of image coding and communication based on the selected foundations. RVQ and MP are able to provide progressive image refinement when they are used in source coder. The fact that earlier parts of the source coder bit-stream have more impact on the quality of the reconstructed image, is the reason that an unequal error protection (UEP) scheme may protect the source coder output better than ordinary equal error protection codes. Besides, the residual information left in the source coder output can be exploited for error concealment at the decoder.

1.2.2 Outline

The rest of this dissertation is organized as follows. In chapter 2, required theoretical background for vector quantization and matching pursuit signal representation are provided. Chapter 3 introduces our new VQ based ROI image coding. In this chapter, RVQ and its variants are adopted as source

coding basis and the issue of rate-quality-complexity trade-off is investigated. Also, UEP effect on the resilience of the reconstructed image quality at the presence of channel error is examined. In chapter 4, image coding based on MP is explored. ROI image coding and quality scalability are investigated and different key players in this image coding scheme are studied. Chapter 5 provides a sub-optimal MMSE based joint source/channel decoding method. In this chapter, the required mathematical formulations for soft decoding of the received MP based bit-stream are developed and different decoding scenarios are investigated. This thesis is concluded in chapter 6, which contains concluding remarks and future directions.

1.3 Contributions and Publications

The results of primary research of the author on the topic of foveation may equip current DCT based video coding standards with a new perceptual coding tool. The original work on fixation point tracking and video foveation was awarded by the Communications and Information Technology Ontario (CITO) in 2004.

The author has contributed to the area of image coding and transmission by the following original developments:

- New Region of interest image coding scheme based on residual vector quantization and its variants.
- Investigation of key elements of RVQ based image coding for rate-quality-complexity trade-off of the source encoder.
- Unequal error protection investigation for the RVQ based image coding bit-stream.

- Novel ROI image coding based on matching pursuit.
- New MP dictionary size reduction method and rate-quality-complexity trade-off investigation for MP based image coding.
- New Joint source/channel decoding method for MP based image coding and transmission system, which can also be applied to the proposed RVQ based image coding scheme.

These contributions are contained within the chapters of the thesis as well as four conference and three journal papers. Most of the contents of chapter 3 have been published in *IEEE Transactions of Multimedia* [5]. An earlier exposition of some parts of this chapter was also presented in *IEEE DCC-2003* [6]. The UEP method and its results for RVQ based ROI image coding bit-stream was presented in *IEEE CCECE 2004* [7]. The contents of chapter 4 have been published in *IEEE Transactions on Image Processing* journal [8] and summarized in *IEEE ICME-2006* [9]. The work in chapter 5 has been submitted for publication to *IEEE Transactions on Image Processing* [10]. Portions of this work was presented in *IEEE ICME-2008* [11]

Chapter 2

Background

In this chapter, two major topics regarding this thesis are explained; vector quantization and matching pursuit methods. The topics are given emphasis regarding to their use in this thesis and not their relative research and application value.

2.1 Vector Quantization

Over the past few decades, the large amount of data associated with digital signals has stimulated a large number of researchers to work on signal compression algorithms. Quantization can be viewed as an essential building block in many lossy signal compression systems, including DPCM [12], JPEG [2], H.263 [13], MPEG-4 [14], and JPEG-2000 [3], where a controlled loss of fidelity in the decompressed signal is tolerated. In a quantization process, a large and possibly infinite set of source output values are represented by a much smaller set of representatives, so the original values are lost forever. The source output values and their representatives can be considered as elements of an L-dimensional Euclidian space $\mathbb{R}^L$. If $L = 1$, the quantization process is

called *scalar quantization*, where every source output value is represented by one of the limited reconstruction levels. For a comprehensive review of different scalar quantization strategies, readers may refer to [1]. In case $L > 1$, we deal with *vector quantization*, commonly denoted by VQ. According to Shannon's rate-distortion theory [15, 16], coding vectors instead of scalars always increases the best achievable compression performance†, even when the data source is memoryless. This implies that a vector quantization strategy, that involves quantizing sequences of source samples, can provide better compression performance than a scalar quantization system.

2.1.1 VQ Process

Prior to and as input to VQ process, the source output is grouped into vectors of size L. This can be done, for example, by treating L consecutive digital audio samples, or a block of L pixels of a digital image as L-dimensional vectors. Vector quantization can be viewed as two distinct operations namely encoding and decoding. Both encoder and decoder of a VQ system have a set of L-dimensional vectors known as *codebook* whose elements, known here as *code-vectors*, are representatives of all the input vectors $\mathbf{X} = (x_1 x_2 \dots x_L) \in \mathbb{R}^L$. Each code-vector $\mathbf{Y}_i \quad i \in \mathscr{I} = \{1, 2, \dots, K\}$ is represented by its index i, where K is the number of code-vectors in codebook $\mathcal{C}$ and $\mathscr{I}$ is VQ index set. VQ encoding is a mapping operation from the infinite L-dimensional space $\mathbb{R}^L$ into the finite index set $\mathscr{I}$, while VQ decoding is a mapping operation from the index set $\mathscr{I}$ into the finite reproduction set $\mathcal{C} \subset \mathbb{R}^L$. Therefore, the whole VQ process can be viewed as the following mapping operation:

$$\text{VQ}: \quad \mathbb{R}^L \rightarrow \mathcal{C} \tag{2.1}$$

†In a signal compression system, better performance can be considered as a lower distortion for a given rate, or a lower rate for a given distortion.

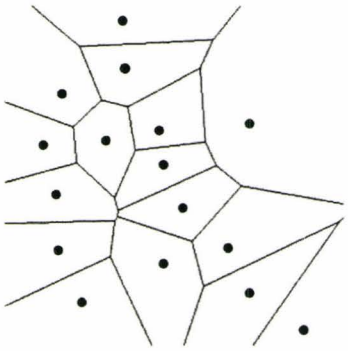


Figure 2.1: Voronoi partition cells

Vector quantization divides the Euclidian space $\mathbb{R}^L$ into K partitions or cells. Each partition is represented by a code-vector. VQ encoder identifies which cell the input vector falls in, and assigns an index based on that indication to the vector. Usually, vector quantizers employ the nearest neighbor or Voronoi cell geometry, in which the closest code-vector to each input vector is selected by VQ encoder. For the nearest neighbor VQ, the geometric description of the cells are not required to be stored. A Voronoi cell $V_i \quad i \in \{1, 2, \ldots, K\}$ is a partition of Euclidian space, which is represented by $\mathbf{Y}_i \in \mathcal{C}$, and is defined by

$$V_i = \{\mathbf{X} \in \mathbb{R}^L : d(\mathbf{X}, \mathbf{Y}_i) \leq d(\mathbf{X}, \mathbf{Y}_j) \quad \forall i \neq j\} \tag{2.2}$$

where $d(.,.)$ is a numerically computable distortion measure. This notion of partitioning can be visualized in two dimensions, where each input vector is a point in a two dimensional space (plane). Fig. 2.1 illustrates a 2-D Voronoi partitioning with dots denoting the code-vectors. Euclidian distance as a measure for $d(.,.)$ is commonly used for vector quantization. Therefore, (2.2) can be written by the following expression

$$V_i = \{\mathbf{X} \in \mathbb{R}^L : \|\mathbf{X} - \mathbf{Y}_i\| \leq \|\mathbf{X}, \mathbf{Y}_j\| \quad \forall i \neq j\} \tag{2.3}$$

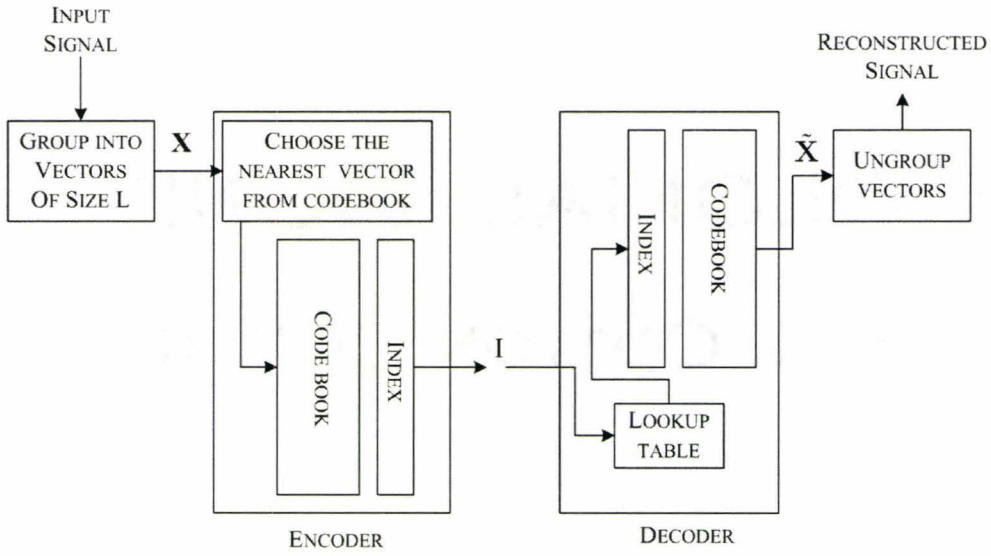


Figure 2.2: Vector quantization procedure

where

$$\|\mathbf{X}\|^2 = \sum_{m=1}^{L} x_m^2. \tag{2.4}$$

The nearest neighbor VQ encoder performs an exhaustive search algorithm to find the best match for every input vector. In this regard, the encoder sequentially computes the distortion caused by every code-vector in the codebook, using mean squared error formula $\|\mathbf{X}-\mathbf{Y}_i\|^2$, and selects the code-vector with least resulting distortion. In the context of digital communication, the index of the selected code-vector is sent to the receiver, where the decoder performs a table lookup to reconstruct an approximate version of the input vector. Fig. 2.2 illustrates this process. VQ encoder performs a considerable amount of computations, which involves $K \times L$ multiplications and $K \times (L-1)$ additions for K distortion evaluation $\|\mathbf{X} - \mathbf{Y}_i\|^2 \quad i = 1, 2, \ldots K$. However, the operation associated with VQ decoder is very simple, involving only table lookup. Therefore, when the available computational resources for decoding is much less than that for the encoding, vector quantization becomes more

attractive.

For a lossy compression system, rate and distortion play important roles. Here, distortion, which represents performance penalty associated with reproducing $\mathbf{X}$ by $\tilde{\mathbf{X}}$, is measured by mean squared error $\|\mathbf{X} - \tilde{\mathbf{X}}\|^2$ and rate R is the number of binary digits that represents a source sample and is measured by the following relation

$$R = \frac{\lceil \log_2 K \rceil}{L} \quad \text{bits per sample} \tag{2.5}$$

where $\lceil x \rceil$ is the smallest integer larger or equal to x. It is clear that using larger dictionaries increases the possibility of finding a better representative for input vector $\mathbf{X}$, however, this increases the required rate. For a given dictionary size and vector dimensionality, determining the codebook for a given source is a key element in reducing VQ distortion.

2.1.2 Codebook Design

VQ *Codebook design* is usually referred to the process of selecting K points in the L-dimensional Euclidian space $\mathbb{R}^L$ as code-vectors. One way of codebook selection is to choose K points randomly in $\mathbb{R}^L$ space. Clearly, this approach does not exploit source statistical characteristics and yields poor VQ performance. A more reasonable approach toward codebook design is to place the reproduction points (code-vectors) in $\mathbb{R}^L$ space wherever the source vectors are most likely to gather . In order to indicate where the source output vectors cluster, a large set of source vectors, known as *training set*, is required. One simple codebook design method is *pairwise nearest neighbor* (PNN) [17] algorithm. In this algorithm, we start with the training set and systematically merge the elements of this set stage by stage, until we form a set with the desired number of elements (K). More specifically, at each stage, two closest

vectors in the set are replaced by their mean and the set cardinality is reduced by one. PNN algorithm effectively combines those vectors that would result in least amount of distortion. The most widely cited VQ codebook design method is the algorithm attributed to Linde, Buzo, and Gray, popularly known as LBG algorithm [18].

2.1.2.1 LBG Algorithm

This algorithm is similar to the k-means algorithm [19], a well known clustering method in pattern recognition. LBG algorithm is also known as generalized Lloyd algorithm (GLA), since it is a generalization of the codebook design algorithm for scalar quantization proposed by Stuart Lloyd [20]. LBG algorithm takes a large training set and K initial representatives of Voronoi cells (as initial code-vectors) and maps the training set vectors on Voronoi partitions $V_i \quad i = 1, 2, \ldots K$, so the average resulting distortion is minimized. The steps of LBG algorithm are intuitive and straightforward. The first step is to find the closest Voronoi cell representative for each training vector and place the vector in that cell. Once all the vectors of the training set are assigned to Voronoi cells, each cell representative is updated by the *centroid vector*, the arithmetic average of the vectors in that cell. This process repeats iteratively and the Voronoi cells and their representatives (as code-vectors) are modified, until a condition on distortion measure is met. Having a certain training set and a set of K initial vectors, the process decreases the average distortion in each stage [21], however, the convergence of the process is rather slow near the point of convergence. A normalized distortion difference measure is calculated in each LBG iteration and is used as the criterion to terminate the codebook design process. In r^{th} LBG iteration, distortion measure is defined as $\frac{D^{(r)}-D^{(r-1)}}{D^{(r)}}$, where $D^{(r)}$ is the average distortion (for example mean

squared error) between the training vectors and their representative reconstruction vectors. If this normalized difference distortion measure falls below a specified threshold ε, i.e. $\frac{D^{(r)}-D^{(r-1)}}{D^{(r)}} < \varepsilon$, LBG design process terminates and the Voronoi cell representatives are selected as final code-vectors.

In LBG process, the average distortion is guaranteed to decrease from one iteration to the next. The process, however, does not guarantee an optimal codebook as the output. The result of LBG codebook design process is closely related to the initial set of K representative vectors. There have been a large number of suggested methods for LBG initialization. One method is to pick randomly K vectors from the training set [22]. Another method, presented by Linde-Buzo, and Gray in their original work [18], is referred to as *splitting technique*. In this method, we start with a single-element codebook. The code-vector is the average of all training set. By adding a fixed perturbation vector, we create another code-vector and double the number of codebook elements. Using LBG process on the training set and the set of two initial vectors, we design a 2-element codebook. By employing perturbation on each code-vector, the initial set of vectors for 4-element codebook is generated. This process continues until K initial vectors are obtained. A two dimensional animated illustration for this process is presented in [23]. The previously discussed PNN algorithm is another LBG initialization method, which usually yields better LBG codebook design among other methods. However, none of the above LBG initialization methods, including PNN algorithm, always provides a globally optimal solution. More LBG initialization methods are presented in [1, 21].

There have been a number of attempts to find alternatives for LBG codebook design algorithm. Neural network based VQ design approach, can have less sensitivity to initialization of the codebook, faster convergence, and better rate-distortion performance [24]. Some codebook design approaches find

the ultimate globally optimum codebook, e.g. [25–27], however due to their computational complexity, LBG still remains the most popular VQ codebook design algorithm.

2.1.3 Structured Vector Quantization

The computational complexity of a VQ encoder (number of arithmetic operations in exhaustive codebook search) and the storage requirements in both encoder and decoder are proportional to the size of VQ codebook. For a given rate R bit per sample and vector dimension L, the size of codebook is $K = 2^{RL}$. For example, for $R = 1$ bit per sample and $L = 64$, the codebook would have $K = 2^{64} \simeq 1.84 \times 10^{19}$ code-vectors! Since reducing the dimension of vectors diminishes the possibility of exploiting the existing statistical dependencies of the source output, ordinary unstructured vector quantizers operate at low bit rates. This may sacrifice the quality of the reconstructed signal.

One approach to solve the VQ complexity problem is to impose some structural constraints on VQ codebook, so the complexity problem (due to nearest neighbor search) and codebook storage are mitigated. By applying these structures and constraints, vector quantization with large vector dimension and codebook size becomes practical, without hitting the complexity and storage barrier. In an unstructured VQ codebook design, K code-vectors are located in $\mathbb{R}^L$ space wherever they minimize the overall quantization distortion, while the code-vectors in structured VQ are distributed in a restricted manner which facilitates the nearest neighbor search process. Although an unstructured VQ with large dimensionality, that is optimally designed for a certain source, yields higher rate-distortion performance "in theory", a structured VQ provides a useful performance and complexity trade-off and makes a high quality signal representation realizable.

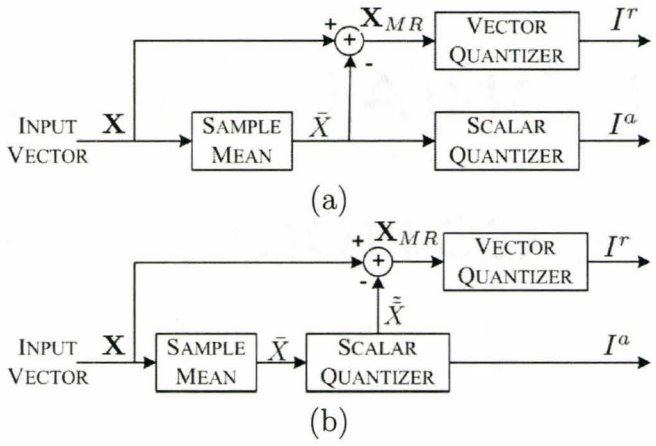


Figure 2.3: Mean-removed vector quantizer. (a) original mean-removed VQ structure (b) alternative structure

One popular way of handling an unmanageably difficult problem is to decompose it into smaller ones and try to solve those sub-problems. This is the essence of what is called *product code technique* [21, 28]. In vector quantization, the encoding process can be viewed as the difficult problem. There are several methods to decompose the VQ encoding task into easier ones. One way of doing so is to divide the vector dimension into smaller ones. For example, instead of considering whole image as an input vector, we can divide the image into image blocks and encode each image block separately. Following are some other structured VQ approaches based on product code technique, which have been used in this thesis. For a comprehensive review of more structured vector quantization methods, readers may refer to [1, 21, 29]

2.1.3.1 Mean-Removed VQ

The average value of vector components, i.e.

$$\bar{X} = \frac{1}{L}\sum_{r=1}^{L} x_r \tag{2.6}$$

where $X = (x_1, x_2, \ldots, x_L)$, is often regarded as statistically independent from the vector component variations around this value. For example, the average pixel values of an image signal represent the level of picture illumination which can be changed without drastic effect on actual image content. The mean-removed residual vector, defined as $\mathbf{X}_{MR} = \mathbf{X} - \bar{X}$, is in the form of a vector of dimension L and represents the pattern content of the original vector $\mathbf{X}$. In a *mean-removed* VQ [21], sometime denoted as MRVQ, both the scalar mean value $\bar{X}$ and the mean-removed residual vector $\mathbf{X}_{MR}$ are quantized separately in order to represent the original vector $\mathbf{X}$ as $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}_{MR} + \tilde{\bar{X}}$.

Figure 2.3(a) illustrates this vector quantization mechanism. The vector quantizer and scalar quantizer blocks in this figure have codebooks $\mathcal{C}_r$ of size K_r and $\mathcal{C}_a$ (scalar quantizer for average value of vector $\mathbf{X}$) of size K_a respectively. The equivalent codebook for the whole mean-removed VQ structure is the product codebook $\mathcal{C} = \mathcal{C}_r \times \mathcal{C}_a$ consisting of $K = K_r \times K_a$ code-vectors. The codebook that is designated for mean-removed residual vector is designed based on the statistical characteristics of those vectors, i.e. using a training set of mean-removed residual vectors. Figure 2.3(b) displays an alternative structure for mean-removed VQ. In this structure, the quantized mean value of each vector is subtracted from the vector to compute residual vector. In the second structure, the input to the vector quantizer depends on the performance of the scalar quantizer and the system can be viewed as a sequential product code, where the overall quantization error between the input and output vectors is equal to the quantization error introduced in the last stage, i.e., the vector quantizer unit. An extensive comparison of these two structures is presented in [21].

2.1.3.2 Residual Vector Quantization

Sequential search product code VQ is a class of structured VQ that simultaneously reduces the necessary memory and computation cost. In this class of quantizers, different layers of the quantizer system represent different features of the input signal. A special type of product code vector quantizer is one with direct sum structure [30], in which the equivalent codebook is constrained to be the direct sum of a number of smaller constituent codebooks. This type of structured VQ can be used in a very memory-complexity efficient fashion. For a P stage direct sum VQ with each stage having codebooks of size K_i, where $i = 1, 2, \ldots, P$, code-vectors, only $\sum_{i=1}^{P} K_i$ code-vectors are required to be stored while, the equivalent unstructured codebook contains $K = \prod_{i=1}^{P} K_i$. Since the exhaustive nearest neighbor search examines all the stored code-vectors, the computation burden is also proportional to $\sum_{i=1}^{P} K_i$, much less than $\prod_{i=1}^{P} K_i$ of the equivalent unstructured VQ.

Residual vector quantizer (RVQ), also known as multi-stage vector quantizer [31–33], is a simple sequential search product code VQ based on direct sum codebook structure which includes multiple successive stages, where each stage encodes the residual (error) vector of the previous stage. Figure 2.4 displays a schematic block diagram of this quantization method. According to this figure, the encoding task is divided into successive stages. The first RVQ encoding stage (Q_1) performs a crude approximation of the input vector using a relatively small dictionary $\mathcal{C}_1$. The index $I^{(1)}$ of the reproduction vector is then transmitted to the receiver. The vector approximation is then refined by the second RVQ quantization stage (Q_2). The residual error vector $\mathbf{E}_2 = \mathbf{X} - Q_1(\mathbf{X})$, the difference between the input vector of the previous RVQ and its coarsely quantized version, is treated as the input to the second RVQ stage. The index $I^{(2)}$ of the selected code-vector from codebook $\mathcal{C}_2$ is sent to

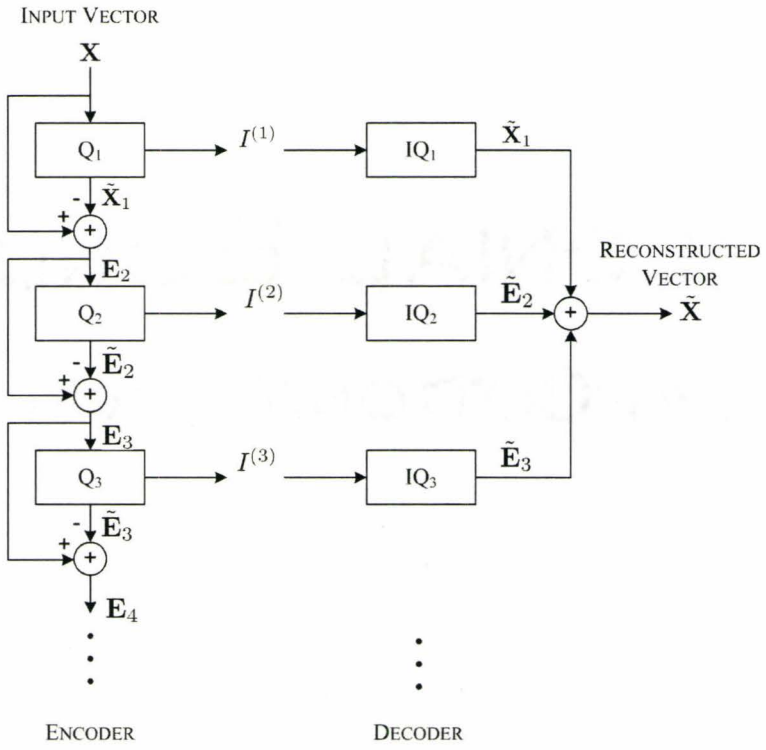


Figure 2.4: RVQ encoder and decoder

the decoder. The third RVQ stage quantizes the second stage residual error vector and further refines the approximation and so on.

The overall quantization error of an RVQ scheme is the error introduced by the last RVQ stage. Hence, in a P-stage RVQ system with a perfect channel, RVQ decoder reconstructs the input vector using summation

$$\tilde{\mathbf{X}} = \tilde{\mathbf{X}}_1 + \sum_{i=2}^{P} \tilde{\mathbf{E}}_i \tag{2.7}$$

and the overall reconstruction error is $\mathbf{E}_{P+1}$. Therefore, signal to quantization noise power (SNR) for this RVQ system can be defined by

$$\text{SNR} = \frac{\text{E}[\|\mathbf{X}\|^2]}{\text{E}[\|\mathbf{E}_{P+1}\|^2]} \tag{2.8}$$

where E[·] is expectation operator. This equation can be expanded into the

following form

$$\begin{aligned} \text{SNR} &= \frac{\text{E}[\|\mathbf{X}\|^2]}{\text{E}[\|\mathbf{E}_2\|^2]} \times \frac{\text{E}[\|\mathbf{E}_2\|^2]}{\text{E}[\|\mathbf{E}_3\|^2]} \cdots \frac{\text{E}[\|\mathbf{E}_P\|^2]}{\text{E}[\|\mathbf{E}_{P+1}\|^2]} \\ &= \text{SNR}_1 \times \text{SNR}_2 \times \ldots \times \text{SNR}_P \end{aligned} \tag{2.9}$$

In logarithmic scale (dB) we have

$$(\text{SNR})_{\text{dB}} = \sum_{i=1}^{P} (\text{SNR}_i)_{\text{dB}} \tag{2.10}$$

For an ideally designed VQ, SNR in dB grows linearly with the number of representative index bits [21]. Therefore, if each RVQ stage is designed perfectly, we may expect the same performance from RVQ as its equivalent unstructured counterpart. This, however, is not quite true. The compression performance of a vector quantizer closely depends on the correlation of components in an input vector and this correlation declines stage by stage in an RVQ quantizer. Therefore, for a given rate, an unstructured VQ always outperforms RVQ scheme. As explained earlier, for a given complexity and memory cost, RVQ method provides superior compression performance compared to unstructured VQ. Residual vector quantization scheme is also suitable for applications that require progressive transmission or successive approximation since it renders the input vector stage by stage and progressively.

RVQ codebook is designed stage by stage. This means that a set $\mathcal{T}_1$ of training vectors is employed to design the codebook of the first RVQ stage. Using this designed codebook $\mathcal{C}_1$, all the vectors of $\mathcal{T}_1$ are quantized and the residual errors of those vectors are placed in a second set $\mathcal{T}_2$ with the same size and dimensionality as $\mathcal{T}_1$. This new training set represents the statistical characteristics of the vectors applied to the second stage of RVQ. Therefore, $\mathcal{T}_2$ is used as the training set for the codebook design of the second RVQ stage ($\mathcal{C}_2$). The rest of the successive stage codebooks are designed by repeating

this process. RVQ codebook design complexity is much lower compared to the codebook design of its unstructured counterpart, another advantage of RVQ scheme. For applications in which the required training set is not large enough for an unstructured VQ and applications that requires codebook transmission as side-information, a multistage VQ with small stage codebook is beneficial [34]. Codebook design for RVQ with many stages and optimality issues are investigated in [33]. An alternative RVQ codebook design based on neural networks is presented in [35].

2.2 Matching Pursuit

Suppose you know the grammar of a language, but your vocabulary is very basic. It is difficult to write about a topic in that language, since it requires lengthy statements to express unavailable words and subtle differences between close concepts. Take another example. In theory, it is possible to paint a natural scene using a palette consisting only of red, green, and blue paints, since every color can be made by a combination of these three primary colors. However, the artist would be frustrated by the amount of required efforts of constantly mixing the paints on canvas for making non primary colors. A more convenient and time saving approach is to use a palette rich of different colors.

The above idea also applies to the classic problem of representing a signal (vector) $\mathbf{X} \in \mathbb{R}^L$ as a linear combination of predefined unit norm vectors $\{\mathbf{g}_\gamma\}_{\gamma\in\Gamma}$ from a set $\mathscr{D}$ called *dictionary*, i.e.:

$$\mathbf{X} = \sum_{\gamma\in\Gamma} C_\gamma \mathbf{g}_\gamma \tag{2.11}$$

where Γ is the set of dictionary elements indices. The artist in the above example is analogous to the algorithm, which combines dictionary elements,

and his/her color palette is similar to dictionary $\mathscr{D}$. Similar to a palette of three primary colors, the dictionary can be restricted to contain the minimum number of elements so that any signal $\mathbf{X} \in \mathbb{R}^L$ can be perfectly represented by the above linear combination. We call this dictionary a *complete* set. The above examples of rich vocabulary and color palette make us think if an *over-complete* dictionary, containing more elements than a complete set, might improve signal representation.

In this thesis, we deal with compact coding of image signals for transmission over a channel, where a suitable lossy representation of signal with low bit-rate and in a progressive manner is desirable. For this kind of signal analysis-synthesis problem, a small subset of a rich, over-complete dictionary may more accurately approximate a signal than a complete dictionary. This sparse signal approximation is in the following form

$$\mathbf{X} \approx \tilde{\mathbf{X}} = \sum_{\gamma \in \Gamma' \subset \Gamma} C_\gamma \mathbf{g}_\gamma \tag{2.12}$$

where Γ' is a small subset of much larger index set Γ. Over-complete signal representation, like a natural language with a rich vocabulary, is able to represent a signal with the least number of dictionary elements. A linear expansion on bases (such as Fourier and wavelet bases) is not able to completely identify the signal patterns and efficiently represent a signal with minimum number of constituent elements. This is due to the dilution of information on the whole basis expansion. Now, the fundamental problem is how to efficiently represent a signal with the over-complete dictionary with a reasonable computational burden. This is how we turn into a technique called *matching pursuit* (MP).

There are other methods, which are extremely complex, such as basis pursuit [36], that provide global optimum solution for (2.12) in an over-complete

signal representation and minimize the following approximation error

$$\varepsilon(P) = \|\mathbf{X} - \sum_{\gamma \in \Gamma'} C_\gamma \mathbf{g}_\gamma\| \tag{2.13}$$

where P is the the cardinality of index set Γ' . It has been shown [37] that global optimal solution for (2.13) belongs to the class of computationally intractable NP-hard problems. It is unproven but widely believed that the number of computational operations required for an NP-hard problem grows faster than any polynomial with order of the size of input vector [38]. Matching pursuits (MP) is closely related to a statistical parameter estimation method called projection pursuit [39,40], which drastically reduces the analysis complexity by tolerating a suboptimal solution [41]. This signal analysis algorithm also has similarities to multi-stage (residual) gain-shape vector quantization [21]. MP algorithm was first introduced by Mallat and Zhang in their original paper [42], where they developed it for one dimensional signal analysis. The algorithm has been employed for an increasing number of applications such as speech coding [43], image coding [44], video coding [45, 46], pattern recognition [47], medical applications [48, 49], and industrial machining [50]. In recent years, video coding based on matching pursuit algorithm has shown better coding efficiency and perceptual quality at low bit rates than its DCT-based counterpart [51–53].

2.2.1 MP Algorithm

Matching pursuits is a *greedy algorithm*[†], which progressively extracts signal structures that are more coherent[‡] with respect to the chosen over-complete

[†]Greedy algorithms are straightforward problem solving strategies. These algorithms take the best immediate (locally optimal) decisions using the information at hand without worrying if the results are globally optimum. However, sometimes they yield globally optimal results [54].

[‡]See [55] sub-section 10.5 for the concept of "coherence"

dictionary. Thus, the convergence rate of this algorithm is related to the coherence of the input signal and the MP dictionary. In the matching pursuit approach, more significant signal structures are extracted earlier during the course of signal analysis. This is the key to an embedded progressive signal transmission, where bit-stream truncation at a limited bit budget or an acceptable signal quality is desirable. The successive approximation of matching pursuit algorithm for image coding applications means that the most visible features of the signal tend to be extracted first. This is a very important property, which will be exploited in this thesis for progressive ROI image coding.

Although using matching pursuit is not restricted to just one dimensional signals, this section explains the algorithm on one dimensional vectors just for the sake of notation simplicity. Generalizing the matching pursuit analysis algorithm to higher dimensions is straightforward. This algorithm is explained for continuous time signals in an infinite Hilbert space framework in [42].

Assume $\mathscr{D} = \{\mathbf{g}_{I^d}\}_{I^d \in \mathscr{I}^d}$ as an over-complete set with finite number of normalized members ($\|\mathbf{g}_{I^d}\| = 1$) in $L^2(\mathbb{R})$ that spans a space of dimension L ($\mathscr{I}^d$ is a finite dictionary index set of N_D elements and I^d is called dictionary index). Each L dimensional vector $\mathbf{X}$ of the space can be projected on a dictionary element $\mathbf{g}_{I^d} \in \mathscr{D}$ to approximate the vector in that direction:

$$\mathbf{X} = \langle \mathbf{X}, \mathbf{g}_{I^d} \rangle \mathbf{g}_{I^d} + \mathcal{R}\mathbf{x} \tag{2.14}$$

where $\langle \mathbf{X}, \mathbf{g}_{I^d} \rangle$ is called the inner product of vector $\mathbf{X}$ with dictionary element $\mathbf{g}_{I^d}$ and $\mathcal{R}\mathbf{x}$ is residual vector or approximation error vector. For two tall vectors $\mathbf{X}_1$ and $\mathbf{X}_2$, inner product operation is defined as $\langle \mathbf{X}_1, \mathbf{X}_2 \rangle = {\mathbf{X}_1}^T \times \mathbf{X}_2$ and is a measure of how much a vector is in the direction of the other one. Since the residual vector $\mathcal{R}\mathbf{x}$ does not have any component left in $\mathbf{g}_{I^d}$ direction, i.e. the approximation error vector is orthogonal to the direction of the selected

dictionary element, using Pythagorean theorem we have:

$$\|\mathbf{X}\|^2 = |\langle \mathbf{X}, \mathbf{g}_{I^d} \rangle|^2 + \|\mathcal{R}\mathbf{x}\|^2 \tag{2.15}$$

where $\|\mathcal{R}\mathbf{x}\|^2$ is the energy of the approximation error. In order to minimize the residual energy, $\mathbf{g}_{I^d} \in \mathscr{D}$ must be selected such that the projection energy $|\langle \mathbf{X}, \mathbf{g}_{I^d} \rangle|^2$ is maximized.

Matching pursuit is an iterative algorithm that successively decomposes a residual vector, using the above-mentioned orthogonal projection, onto the most similar element of MP dictionary $\mathscr{D}$. To start the MP successive projection, assume the initial residual vector to be the original vector $\mathbf{X}$, i.e. $\mathcal{R}^{(0)}\mathbf{x} = \mathbf{X}$ [†]. Also suppose we have successively decomposed the vector $\mathbf{X}$, $r-1$ times and the $(r-1)^{th}$ order residue $\mathcal{R}^{(r-1)}\mathbf{x}$ is already computed. As r^{th} MP iteration, to decompose $\mathcal{R}^{(r-1)}\mathbf{x}$ in $\mathbf{g}_{I^{d(r)}}$ direction we have:

$$\mathcal{R}^{(r-1)}\mathbf{x} = \langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^{d(r)}} \rangle \mathbf{g}_{I^{d(r)}} + \mathcal{R}^{(r)}\mathbf{x} \tag{2.16}$$

where

$$\langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^{d(r)}} \rangle = \sup_{I^d \in \mathscr{I}^d} \langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^d} \rangle \tag{2.17}$$

By induction and after η MP iteration we have

$$\mathbf{X} = \sum_{r=1}^{\eta} \langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^{d(r)}} \rangle \mathbf{g}_{I^{d(r)}} + \mathcal{R}^{(\eta)}\mathbf{x} \tag{2.18}$$

In equation (2.18) the original vector $\mathbf{X}$ is approximated by η dictionary elements. With a complete dictionary set, as η tends to infinity, the residual norm exponentially tends to zero [41, 55], in other words

$$\lim_{\eta \to \infty} \|\mathcal{R}^{(\eta)}\mathbf{x}\| = 0 \tag{2.19}$$

[†]As we can see later in chapter 4, in order to represent a vector by minimum number of MP analysis parameters, the mean value of the original signal can be removed. The mean-removed vector can then be treated as $\mathcal{R}^{(0)}\mathbf{x}$.

therefore

$$\mathbf{X} = \sum_{r=1}^{\infty} \langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^{d(r)}} \rangle \mathbf{g}_{I^{d(r)}} \tag{2.20}$$

The approximation error decays faster if the MP dictionary is more coherent to the signal being analyzed. Signal decomposition in r^{th} MP iteration is characterized by the inner product coefficient $\langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^{d(r)}} \rangle$ and the index of the selected dictionary element $I^{d(r)}$. The balance of information between the inner product coefficients and dictionary indices in each MP iteration changes with the size of dictionary. If the dictionary is reduced to N dimensional complete orthonormal basis, all the information is concentrated in inner product coefficients. On the other hand, in case of a dictionary with all possible unit vectors, all the analysis information is compacted in the chosen dictionary index [42].

In real life, MP decomposition iteration has to be terminated at some point. The residual projection on dictionary elements may be iterated up to a predefined number of times. Alternatively, matching pursuit analysis algorithm may continue until the residual energy becomes less than a positive threshold, i.e.

$$\|\mathcal{R}^{(r)}\mathbf{x}\|^2 \leq \mu^2 \times \|\mathbf{X}\|^2 \qquad 0 < \mu < 1 \tag{2.21}$$

For the purpose of using MP decomposition for signal encoding, each inner product coefficient should be quantized in each MP iteration and the resulting quantized value can then be employed for residual vector calculation. Therefore, a signal $\mathbf{X}$ can be approximated by a P-element linear combination:

$$\mathbf{X} \approx \tilde{\mathbf{X}} = \sum_{r=1}^{P} \mathrm{Q}(\langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^{d(r)}} \rangle) \mathbf{g}_{I^{d(r)}} \tag{2.22}$$

where $\mathrm{Q}(\cdot)$ is scalar quantization operator.

Selecting the closest dictionary element (best match) to the residual error in each MP iteration is computationally the most intensive part of the algorithm

since the inner product operation $\langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^d} \rangle$ must be calculated for all dictionary elements (N_D times). Mallat suggests a method [42] to reduce this computational complexity by storing the inner product of dictionary elements with each other. By taking inner product operation on both sides of equation (2.16) with $\mathbf{g}_{I^d}$ we have the following iterative updating formula:

$$\langle \mathcal{R}^{(r)}\mathbf{x}, \mathbf{g}_{I^d} \rangle = \langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^d} \rangle - \langle \mathcal{R}^{(r-1)}\mathbf{x}, \mathbf{g}_{I^{d(r)}} \rangle \times \langle \mathbf{g}_{I^{d(r)}}, \mathbf{g}_{I^d} \rangle \tag{2.23}$$

In this equation, the dictionary inner products $\langle \mathbf{g}_{I^{d(r)}}, \mathbf{g}_{I^d} \rangle$ can be computed and stored in advance to be used as lookup table later during the course of MP analysis. Equation (2.23) updates the inner product of the residual error with all dictionary elements without employing direct inner product calculation. This drastically reduces the computational complexity of the matching pursuit analysis [42].

All the previously described matching pursuits formulations are also valid for a 2D signal (like an image) if the inner product operation is properly defined. For this purpose and in MP analysis context, we define a 2D inner product of two matrices $\mathbf{A}$ and $\mathbf{B}$ of identical size $M \times N$ as the following summations:

$$\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{A}(i,j) \cdot \mathbf{B}(i,j) \tag{2.24}$$

A more computationally complex variant of matching pursuit, namely *orthogonal matching pursuit* [41, 43] provides a faster converging signal approximation than the original matching pursuit by removing the selected dictionary element not only from the signal but also from other elements in the dictionary using Gram-Schmidt orthogonalizing procedure. It has been shown [41] that for a vector $\mathbf{X}$ of size L, an orthogonal matching pursuit decomposition results in zero residual energy after $\eta \leq L$ iterations. In addition to its excessive computations both in analysis as well as synthesis stages, orthogonal MP

method deforms MP dictionary elements and their patterns during the course of decomposition of a vector.

2.2.2 MP Dictionary

An interesting property of matching pursuit signal decomposition is that it imposes no restriction on the choice of dictionary. Mallat and Zhang in their original work [42] employed a dictionary of Gabor and wavelet functions, however the algorithm itself does not require dictionary with any specific structure, as long as the completeness condition is satisfied. The dictionaries can be application specific and tailored according to the signal statistical properties in order to increase the convergence speed of the MP algorithm. A class of MP dictionaries, which is very popular in over-complete signal analysis is *time-frequency dictionary* class [42].

2.2.2.1 Time-Frequency Dictionary

A Fourier basis poorly represents signals that are localized well in time domain (spatial domain for image signals) and wavelet basis does not properly represent signals with high frequency and narrow Fourier transform support [55]. Signal energy compaction (localization) of a function $g(t) \in L^2(\mathbb{R})$ in time and frequency can be measured by the variance of the function in time (σ_t) and frequency (σ_ω) defined by the following relations

$$\sigma_t^2 = \frac{1}{\|g\|^2} \int_{-\infty}^{\infty} (t - u_0)^2 \, |g(t)|^2 \, dt \tag{2.25}$$

$$\sigma_\omega^2 = \frac{1}{\|\hat{g}\|^2} \int_{-\infty}^{\infty} (\omega - \xi_0)^2 \, |\hat{g}(\omega)|^2 \, d\omega \tag{2.26}$$

where $\hat{g}(\omega)$ is Fourier transform of $g(t)$. In the above equations, u_0 and ξ_0 can be viewed as energy center point of g in time and frequency domains and

defined respectively as the following equations

$$u_0 = \frac{1}{\|g\|^2} \int_{-\infty}^{\infty} t \, |g(t)|^2 \, dt \tag{2.27}$$

$$\xi_0 = \frac{1}{\|\hat{g}\|^2} \int_{-\infty}^{\infty} \omega \, |\hat{g}(\omega)|^2 \, d\omega \tag{2.28}$$

For signals with widely varying time-frequency localization, an over-complete dictionary consisting of dilated, modulated, and translated versions of a single windowed function contains a large variety of elements (also known as time-frequency *atoms*) with different time-frequency properties. Suppose $g(t) \in L^2(\mathbb{R})$ is a normalized ($\|g(t)\|^2 = \int_{-\infty}^{\infty} |g(t)|^2 dt = 1$), even function, with nonzero value at the origin ($g(0) \neq 0$) and nonzero mean (i.e. $\int_{-\infty}^{\infty} g(t) \neq 0$). Elements of the triplet $\gamma = (s, \xi, u) \in \mathbb{R}^+ \times \mathbb{R}^2$ represent scaling, modulation, and translation parameters, respectively. A time-frequency atom is defined as

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g(\frac{t-u}{s}) e^{j\xi t} \tag{2.29}$$

This function is normalized by $\frac{1}{\sqrt{s}}$ term and its energy is concentrated around its central point $t = u$ in a neighborhood of size proportional to s. Time-frequency atom g_γ in Fourier domain is

$$\hat{g}_\gamma(\omega) = \int_{-\infty}^{\infty} g_\gamma(t) e^{-j\omega t} dt = \sqrt{s}\hat{g}\left(s(\omega - \xi)\right) e^{-j(\omega-\xi)u} \tag{2.30}$$

which is localized around frequency $\omega = \xi$ on a frequency neighborhood proportional to $1/s$. According to (2.29) and (2.30), small values of scaling parameter s results in well localized signal in time domain and poorly concentrated signal in frequency domain, while atoms are well localized in frequency domain and poorly concentrated in time domain for large values of this parameter. This fact complies with Heisenberg uncertainty principle, which implies that a signal cannot be compact both in time and frequency domain. This principle

states that the temporal and frequency variances of a function $g(t) \in L^2(\mathbb{R})$ are restricted by the following inequality [55]

$$\sigma_t^2 \sigma_\omega^2 \geq \frac{1}{4} \tag{2.31}$$

It has been shown that this inequality becomes equality if and only if there exist $(\upsilon, \varphi, a, b) \in \mathbb{R}^2 \times \mathbb{C}^2$ such that

$$g(t) = ae^{j\varphi t - b(t-\upsilon)^2} \tag{2.32}$$

This function can provide the highest temporal and frequency localization. For an interesting case of $(\upsilon, \varphi, a, b) = (0, 0, \sqrt[4]{2}, -\pi)$ we have Gaussian function $g(t) = \sqrt[4]{2}e^{-\pi t^2}$. This function satisfies the aforementioned conditions for being a time-frequency prototype. The time-frequency dictionaries generated by this function are very popular in MP analysis and are called Gabor dictionaries.

For representing real signals, (2.29) can to be modified to

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g(\frac{t-u}{s}) cos(\xi t + \phi) \tag{2.33}$$

In order to generate dictionaries of finite cardinality for decomposition of vectors with limited size, time-frequency atoms must be sampled in time. The scaling, modulation, and translation parameters should also be quantized.

Chapter 3

Region-of-Interest Image Communication Using VQ

Rate scalability of still-image coding is the ability of extracting the visual information at continuously varying data rate from a single compressed bit stream. This property allows transmission for more than one reconstructed image quality by generating a bit stream in a manner in which the target bit rate or reconstruction quality need not be known at the time of compression. Image coding with progressive quality refinement provides rate and quality scalability for a given image coding scheme. A coarse version of image is transmitted first, decoded, and displayed at the receiver end. At this stage, the user can decide either to further refine the entire image or to determine region(s) of interest to be refined first. Obviously, in order to exploit the available bandwidth optimally, the refinement information should be complementary to the previously transmitted image data, and this can be achieved via progressive image coding.

As the Internet becomes increasingly popular, more and more high resolution images are brought online. NASA is putting its collection of space photos

online. The Microsoft terra server provides access to high resolution satellite/aero photographs over the Internet. Museums are digitizing and moving their collectibles online. Even individuals are sharing their personal photos over the web. It is very enjoyable to watch a high-resolution image on the screen, but it is equally painful to download the huge image over the slow Internet. Although the backbone of the Internet keeps improving, the content and the number of users also grow. Efficient delivery of large image is thus crucial to provide enjoyable image browsing experiences on the Internet.

In the early days, compressed images were downloaded entirely from the network before their contents were rendered. Newer version of the browsers support progressive streaming such as JPEG 2000 [56] and progressive JPEG [57] standards, where views are rendered repeatedly as more and more bit-stream arrives. Progressive streaming improves the experience of image browsing. However, for a large image, the size of the compressed bit-stream is still too big to be streamed efficiently. One solution to this problem is to send a part of the image which is of more interest at a higher quality and if the bandwidth permits, improve the quality of the rest of the image. Moreover, in many applications (e.g., image streaming) the end user is browsing through a set of images and would prefer to receive a coarse version of the image first and have the option of interactively refining part or parts of the coarse received image.

Image compression schemes operate more efficiently when they take into account the region of interest (ROI) of the end-user on the image to be compressed. Transmitter can send the information of the ROI with higher quality than that for the background. The functionality of this method of image coding is significant in applications where some specific parts of the image are of higher degree of importance than others.

In this chapter VQ-based methods for progressive and interactive ROI image coding are presented. The proposed schemes are computationally unbalanced and the receiver is much simpler than the transmitter, a property which is sometimes favorable in commercial image archiving. In order to get a progressive coding scheme with bit-rate scalability, an RVQ based algorithm is presented which can automatically provide a useful trade-off between rate and distortion in accordance with the network situation. With this RVQ method, receiver is able to change the ROI without losing the previously received information. Furthermore, the receiver is able to request multiple ROIs and the transmitter progressively sends a version of image with higher quality in the regions of interest. Jointly sub-optimized RVQ without any increase in bit-rate ameliorates the reconstructed image quality at the expense of more encoding complexity. In order to optimally reduce the effect of the channel error on the reconstructed image quality, a variable error protection scheme is applied to the output bit-stream of our source encoder. Since the bit-stream portions at the output of the proposed coder have different levels of importance in terms of their contribution to the final reconstructed image quality (in an MMSE sense), variable number of redundancy bits are added to the bit-stream portions to protect them against the channel error accordingly. For unequal error protection (UEP) simulations, we use Reed-Solomon forward error protection coding method.

This chapter is organized as follows: Section 3.1 presents our progressive interactive RVQ based ROI image coding methods. Mean-removed RVQ based ROI image coding and another alternative for the RVQ based ROI image coding scheme, which compromises the reconstructed image quality with the encoder complexity, are also introduced in this section. Unequal error protection using Reed-Solomon codes is presented in section 3.2. Section 3.3 presents

the simulation results which gives examples of ROI coded images with each of the aforementioned methods and illustrates the rate-distortion behavior of the schemes. This section is also includes some simulation results that reflects unequal error protection performance of the proposed ROI image transmission.

3.1 ROI Image Coding Using VQ

3.1.1 Vector Quantization of Image

Image compression by vector quantization consists of forming the image vectors (by grouping this two dimensional signal into blocks of pixels of M columns and N rows, and treating them as an $L = M \times N$ dimensional vector), comparing the vectors with the predefined vectors in the codebook, transmitting the index of the closest vector to the image vectors, and reconstructing the image with the vectors whose indices have been sent [1,21] (Fig. 2.2). For maximum exploitation of two dimensional correlation between image samples, the image blocks are generally selected as squares [1], i.e. $M = N$.

3.1.2 A Basic Intuition

The intuition for ROI image compression based on vector quantization comes from the fact that there is a close relationship between the quality of the reconstructed image and the corresponding codebook size, i.e. reconstructed image with higher quality can always be obtained by using larger codebooks. Thus, a simple method for ROI image coding is to use larger codebook for region of interest than that for the background. A low quality image is sent once with an initial small sized codebook. The user who receives the image

determines his/her point(s) of interest and sends this information to the transmitter. The transmitter quantizes the ROI information by means of a bigger codebook. This simple method is not progressive, i.e., the information of the low quality image cannot be employed in rendering the higher quality image of the ROI.

3.1.3 ROI Image Coding Using RVQ

As a result of serious complexity barrier in unstructured VQ, practical direct use of VQ is limited to rather modest codebook size and vector dimensions. While for a given bit-rate, smaller vector dimension tends to have lower complexity and codebook size, bigger vector dimension takes advantage of the correlations between more vector elements (pixels). As an ROI coding technique, the unstructured VQ does not enjoy progressive and embedded transmission in a sense that any request of improvement in the ROI is always replied by complete retransmission of image vectors with a bigger codebook VQ scheme. This problem can be solved by employing residual vector quantization. RVQ can provide a multi-resolution representation of image vectors, an attractive property for progressive ROI image coding. In this structured VQ method, codebook enlargement is realized using more quantization stages.

Fig. 2.4 shows the general configuration of a residual vector quantization system. In the first stage of a residual vector quantizer, a coarse approximation of the input image is generated by using a low-rate vector quantizer Q_1 and the indices of the chosen codebook vectors are sent to the receiver. As the second stage, the errors between the original input vectors and their coarse representation are quantized by Q_2 and the indices of the chosen vectors from the second stage codebook are transmitted. In this manner, the errors between the original image vectors and the approximated vectors obtained from the

outputs of the preceding $(n-1)$ stages are applied to the n^{th} stage and the indices are transmitted. By providing different quality layers, this technique enables progressive encoding of the original image.

The main idea for RVQ based ROI image coding is to use more quantization stages for the image vectors closer to the point(s) of interest. A coarse version of image using the first stage of RVQ is sent to the receiver. The receiver indicates its point(s) of interest and sends the coordinates of the point(s) to the transmitter. The transmitter starts refining the image by sending the indices of residues of image vectors inside the ROI using the next RVQ stages. After some stages of refinement (P_{ROI}) and transmitting the corresponding RVQ indices, the vectors in the background are also started being refined. This improvement routine for each image vector continues and whenever the last RVQ refinement layer is done, the quantization and transmission for that vector stops but the process is continued for other image vectors until all the image vectors are rendered using all available RVQ stages.

This progressive method of streaming provides quality and bit-rate scalability. The transmission can be terminated in the middle of refinement process and the receiver is still able to reconstruct a version of the image with a quality corresponding to the received information.

The aforementioned quantization method can be used in order to implement an RVQ based ROI image coding such that the quality of image gradually reduces from the ROI to the background. In order to achieve this, the encoder finds d_j, the distance of the j^{th} image block from the point of interest, and determines P_j, the number of RVQ stages by which the j^{th} vector is to be quantized, where $j = 1, 2, \ldots, N_B$ represents the image vector indices and N_B is the number of blocks in the image. The RVQ proceeds on j^{th} image vector and encodes it with stage $P_j + 1$ if d_j is less than a metric R_k defined for k^{th}

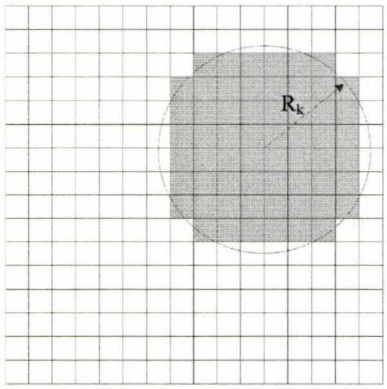


Figure 3.1: The entitled image blocks for quality refinement

stage of image refinement. The distance metric R_k is the radius of a virtual circle, with the center of the point of interest, containing the centers of the image blocks that are going to be refined at the k^{th} refinement stage, as illustrated in Fig. 3.1. This quantization strategy enables gradual image quality change from the ROI to the background. If the network situation is taken into account for the definition of R_k, we can have a bit-rate scalable coding scheme.

More specifically, our proposed method of ROI coding based on progressive RVQ operates as follows. Using the first stage of RVQ, a coarse version of the image is transmitted. The receiver sends the ROI information (the coordinates of the center of ROI). The encoder investigates all the image blocks to check whether the distances of their centers from the center of ROI (d_j) is less than an initial value of R_1. In case the distance (d_j) is more than R_1, nothing is sent for j^{th} image vector. If the distance d_j is less than R_1 , the residue of the j^{th} image vector is quantized using the second stage of the RVQ codebook. In this manner, all the image vectors are investigated and the next stage of RVQ encoding is completed. To further improve the image quality in the ROI, using the relation

$$R_{k+1} = \alpha \times R_k \tag{3.1}$$

the encoder obtains the new distance metric. The above procedure is repeated for all image vectors using R_{k+1} as the distance metric. This process forms the successive layers of the image residues to be sent with emphasizing on the ROI.

In (3.1), α determines how gradual the background quality is increased toward the ROI. The value which is assigned to α plays an important role to give greater importance to the ROI visual information. Taking into account the bit budget of image transmission and the importance of the ROI compared to the background, a proper value has to be assigned to α. In the case of absolute ROI importance and low bit budget, α is set to 1, which means the refinement bits are sent only for the ROI. If a larger value is assigned to α the distance metric R_k grows rapidly. On the other hand the circle with radius R_1 and the center indicated by the point of interest is an initial region whose image vectors are sent with maximum priority. Since bigger R_1 means more image vectors with this privilege, there would be no ROI if R_1 exceeds the size of the image. Practical values for R_1 can be in the range of 1/10 to 1/2 of the size of image. Thus, if network cannot supports a high bit-rate transmission, by choosing α properly close to 1, and assigning a suitable value to R_1 a higher priority of transmission can be given to the ROI image vectors. The transmission can be terminated whenever an acceptable image quality is provided at the receiver end.

The gradual quality changes around ROI imitates the foveation property of the human visual system (HVS) [58]. If the streaming is terminated by the receiver in the middle of transmission, the receiver already has a foveated image version. For more information about different foveation methods see appendix A and [59–62].

This RVQ based method allows us to change the ROI without losing the

previously received information. To add this ability to the proposed method, we need to make a record of the number of RVQ stages by which each image vector has been encoded so far. Receiver should send its stage number, by which it was previously receiving and refining the image, and the coordinates of new point of interest. Using these information, the transmitter can refine the visual information around newly defined point of interest without sending the information twice. Let $P_j^{(-1)}$ denote the number of RVQ stages by which the j^{th} block is coded. If the distance d_j is less than R_k in the $(k+1)^{th}$ RVQ stage, encoder checks $P_j^{(-1)}$ and if $P_j^{(-1)} > k+1$ transmitter sends nothing for that image block, otherwise, the residue of the vector is quantized using the $(k+1)^{th}$ stage of the RVQ codebook and the index is transmitted and $P_j^{(-1)}$ is updated.

Besides having the flexibility of changing the point of interest, multiple ROI can also be handled in the proposed scheme. In this case d_j is calculated by

$$d_j = \min_i d_j^{(i)} \quad i = 1, 2, ..., N_{\text{ROI}} \quad j = 1, 2, ..., N_B \tag{3.2}$$

where $d_j^{(i)}$ is the distance of the j^{th} image vector from i^{th} point of interest and N_{ROI} is the number of ROIs.

3.1.4 ROI Image Coding Using Mean Removed RVQ

The vast variation of the background illuminations causes the codebook in a VQ to be very much dispersed. As a result, two identical images with different background illuminations may have different VQ representation. To reduce this effect, the mean values of image vectors can be removed and sent separately using a scalar quantizer and the resulting image vectors can be quantized with a mean removed codebook [21].

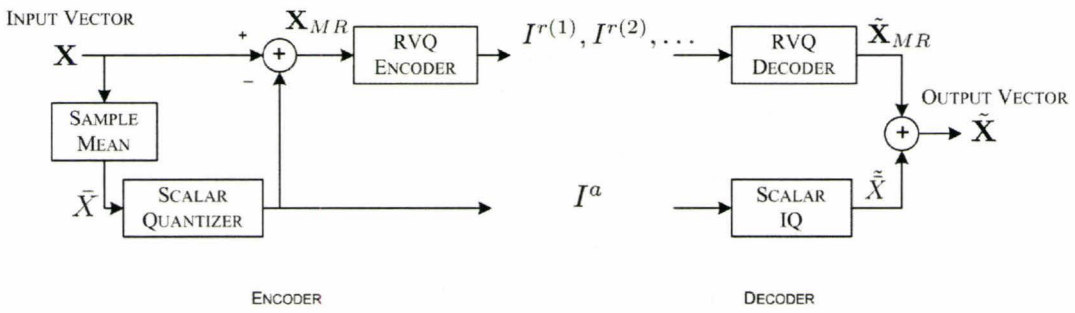


Figure 3.2: Mean removed VQ encoder and decoder

Removal of image block means will not impact the performance of the previously described RVQ based ROI image coding scheme. Figure 3.2 shows a schematic diagram for the mean removed RVQ method. In this method a coarse version of the image, reconstructed by image block mean values, is sent using a scalar quantizer. Since the quantized mean of each vector (and not the exact mean) is removed from the original image, no additional error caused by scalar quantization is introduced. Using a scalar quantizer for the transmission of the mean value of the image reduces the minimum achievable bit per pixel for the coarse image representation.

3.1.5 ROI Coding Using Jointly optimal RVQ

Jointly optimized RVQ (JORVQ) is an RVQ paradigm which was introduced to effectively minimize the overall quantization error [32, 33, 63, 64]. It can be shown that a sequential single-path search through RVQ stages cannot, in general, utilize all available code vectors. JORVQ employs M-search, an efficient multi-path tree search algorithm, to search the stage codebooks. Fig. 3.3 is an example of optimal path through a lattice. According to this figure, the decision for choosing the proper codebook vectors is made after examining the quantization error caused by all the possible paths and the path with

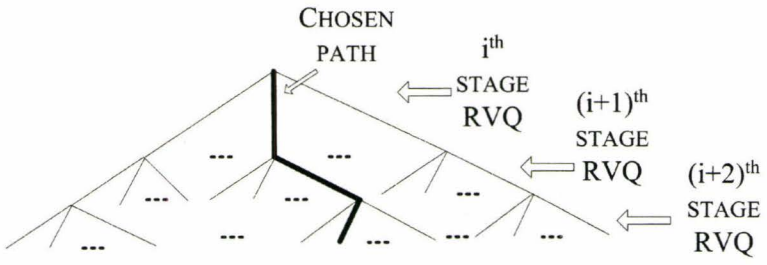


Figure 3.3: JORVQ with $M = 3$ search

minimum quantization error is chosen. This design results in improvement in reconstructed image quality. However, this increase in rate-distortion performance comes at the expense of additional computation. For RVQ systems with identical codebook size $N = 2^K$ for each stage, the computation and memory cost for M-search (exhaustive) JORVQ is proportional to 2^{MK}, while the cost for ordinary RVQ to calculate indices of M consecutive stages is $M \times 2^K$. So by choosing the value of M very large, the computational cost of JORVQ may be prohibitive.

It is clear that for JORVQ if M, the number of RVQ stages jointly considered for optimization, is large, the computational and memory cost is enormous and unmanageable. To address this problem we can consider a constraint of maximum number of best paths which are under consideration. If the number of paths, which their track should be kept, exceeds this constraint, best N_c paths (with lowest resulting quantization error) are kept and the rest are thrown away. $N_c \geq 1$ is the constraint number for maximum paths and its selection is a trade-off between optimality level and the memory and computational cost. This method gives a jointly suboptimal RVQ scheme and improves our previously described RVQ. For RVQ systems with identical stage codebook size $N = 2^K$, the computational cost for M-search (exhaustive) JORVQ with constraint number N_c is proportional to $M \times N_c \times 2^K$. It is clear that this

computational cost is less than that for JORVQ, specially for large values of M. Since there is nothing to be changed in the structure of the receiver, the transmitter can switch either to RVQ or to JORVQ methods. Also the transmitter may use JORVQ for the ROI and RVQ for the background.

3.2 Unequal Error Protection of RVQ Coded Bit-Stream

It is clear that in the embedded bit-stream resulted by the proposed ROI coding based on RVQ procedure, the earlier portions of the bit-stream have more contribution in the reconstructed image quality than the later portions. Indeed, by incorrectly decoding the information of the i^{th} RVQ stage of an arbitrary image vector, correctly decoding of the $j^{th}, (j > i)$ RVQ stage information is ineffective. Thus to protect the output bit-stream of the RVQ based coder, an unequal error protection (UEP) [65] (in a sense to protect earlier parts of the bit-stream more than the later parts) results in better final rate-distortion performance than an equal error protection scheme when the bit-stream is transmitted over a noisy channel. In the proposed RVQ based image coding, the initial coarse version of the image is protected with maximum amount of error protection, dictated by the channel error rate condition. The remaining bit-stream is divided into some sub-bitstreams and each sub-bitstream is error protected no less than the next sub-bitstreams.

Reed-Solomon codes are popular error correcting tools for protecting the transmitted bit-streams against the errors. By assigning a larger number of redundancy bits to the output codewords of the source coder, larger number of bit errors can be corrected [4]. A brief description of Reed-Solomon codes

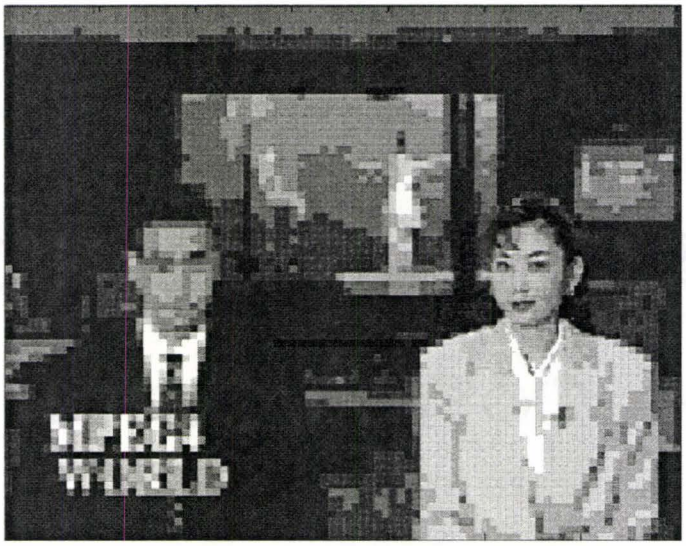


Figure 3.4: "news" image using ROI coding by multiple codebook size (unstructured VQ) with 0.3 bpp.

is given in appendix B. Although unequal error protection can be realized by most forward error correcting methods, Reed-Solomon codes are employed for simulation results due to their popularity.

3.3 Experimental Results

For the following experimental results codebooks are built using Linde-Buzo-Gray (LBG) algorithm. As the training set, five different images (not including "news" image) are employed. For the multiple stage VQ schemes, the error vectors in a given stage form the training set for the next stage (Figure 2.4). For the first stage of RVQ, we use original training set (formed by five different images) and for mean-removed RVQ, we use the mean removed version of that training set.

Here, we used 8 bits per pixel (8 bpp) gray scale "news" image to demonstrate the performance of our methods. Fig. 3.4 shows an example of the

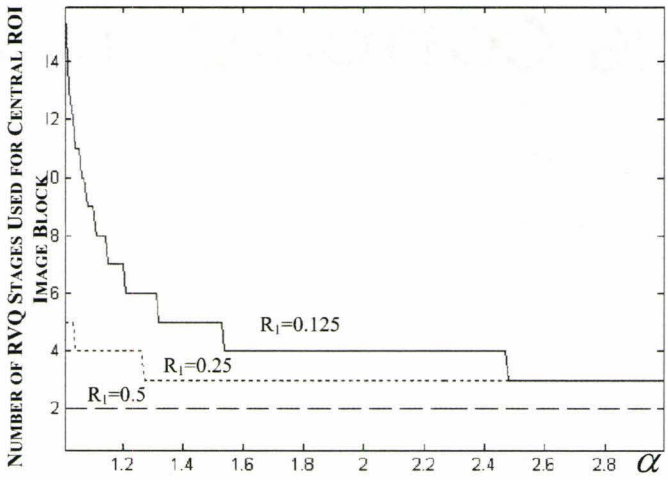


Figure 3.5: Number of RVQ (or JORVQ) stages used for the representation of the point of interest image block with respect to α and for different initial radius R_1. The final number of the transmitted bits is constrained to 0.15 bpp. The point of interest is considered the centre of a 288×352 pixels image.

ROI image coding using proposed unstructured VQ based scheme. This figure demonstrates the ROI image coding using unstructured VQ with vector dimension of 16 (blocks of 4×4 pixels). Codebook sizes used for this image are 8 and 512 for the background and the ROI, respectively, and the overall rate is 0.3 bpp.

Fig. 3.5 shows the relation between the number of RVQ (or JORVQ) stages used for the representation of the point of interest image block and parameter α for three different values of the initial distance metric R_1 (the unit for measuring R_1 is the maximum between length and width of the image). In this figure the transmitted bits per pixel is constrained to 0.15 bpp. The point of interest is assumed to be the center of a 288×352 pixels image. According to this figure, the closer the value of α to 1 means the larger RVQ stages assigned to the ROI for some given transmission bit budget. The figure also shows how the initial distance metric R_1 plays a role for giving more priority to the ROI quality refinement.

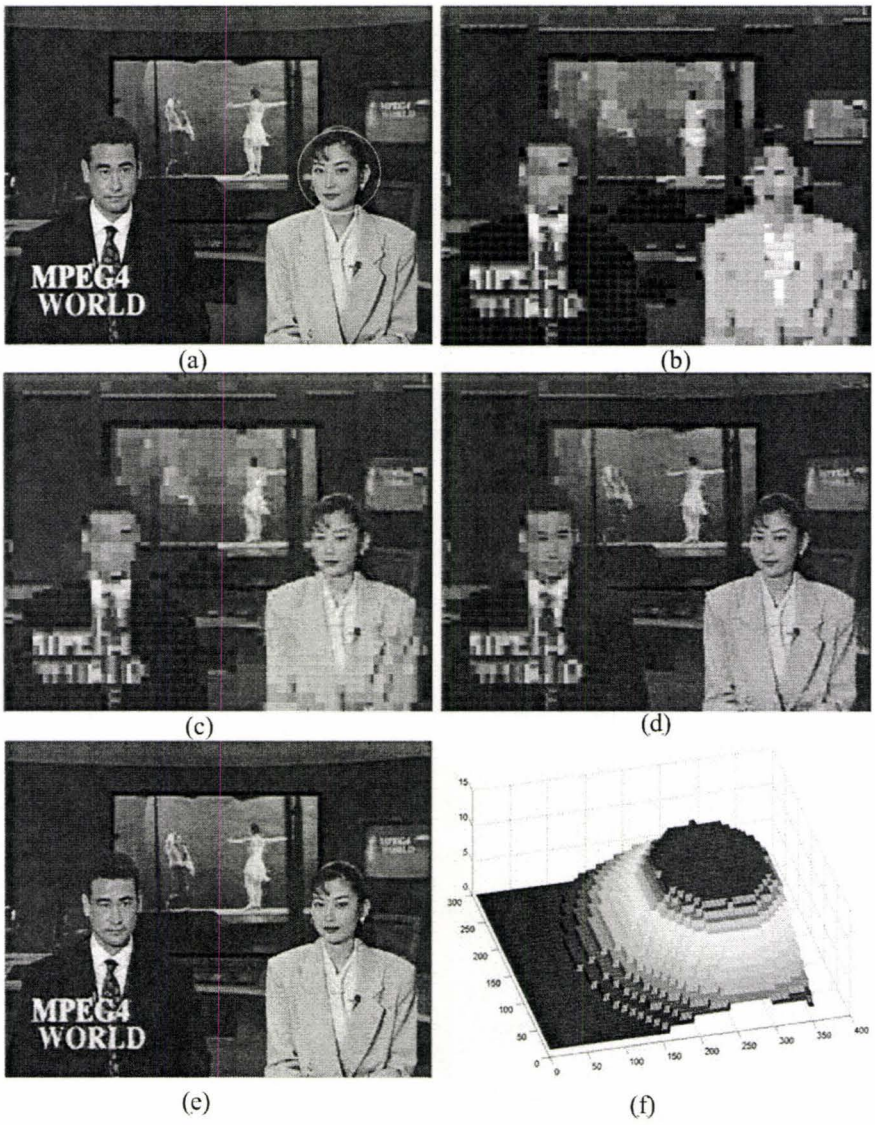


Figure 3.6: Progressive ROI image coding results for "news" image using RVQ based method. Vector dimension is 64 (blocks of 8 × 8 pixels) and codebook size in each stage is 32. The maximum number of stages is 15. (a) original "news" image 8 bpp with indicated ROI . (b) Initial coarse image which is received by end-user and is employed to delineate the ROI (c) Image is reconstructed progressively during the course of transmission. the reconstructed image by the progressive method in the middle of transmission. $R_1 = 0.25$ and $\alpha = 1.1$ with 0.16 bpp. (d) Reconstructed image by the progressive method in the middle of transmission and with more received information than in (c). $R_1 = 0.25$ and $\alpha = 1.1$ with 0.46 bpp. (e) Final reconstructed image by 15-stage RVQ scheme, with 1.17 bpp (f) Spatial RVQ stage status for reconstructed image in part d

Fig. 3.6 shows examples of progressive RVQ based ROI image coding. Fig. 3.6(a) shows the original 8 bpp "news" image with indicated region of interest. To implement the method, we used image vectors with dimension of $L = 64$ (blocks of 8×8 pixels). Codebook size in each stage is 32 and the maximum number of RVQ stages is 15. Fig. 3.6(b) shows the initial coarse image which is transmitted to the end-user in order to make an interaction with the receiver for ROI indication. Assume the ROI is the one that is indicated in Fig. 3.6(a). This coarse image is sent approximately with 0.078 bpp. Fig. 3.6(c) shows one incompletely refined version of reconstructed image at the receiver by the upcoming stream of progressive image information. In this figure $R_1 = 0.25$, $\alpha = 1.1$, and at this stage the image has been received with 0.16 bpp. If we let the streaming to continue, the image quality will improve. Fig. 3.6(d) shows the more refined version of received image with the same ROI image coding parameters but with 0.46 bpp. After all layers of refinement (15 RVQ stage in this case) the final version of reconstructed image is obtained (with 1.17bpp). Fig. 3.6(e) shows this final reconstructed image. Fig. 3.6(f) shows the status of the RVQ refinement stages in each part of the image. This figure is associated with Fig. 3.6(d). This figure demonstrates the ROI oriented priority for progressive streaming. The priority of refinement is approximately an exponential function of distance from the center of ROI. This function is closely related to α and R_1.

In Fig. 3.7 we show another example for the progressive RVQ-based ROI image coding. In this figure we used image vectors with dimension of $L = 16$ (blocks of 4×4). Fig. 3.7(a) shows the original image with 4 indicated ROIs. A coarse version of image using first RVQ stage is sent to the receiver with 0.32 bpp [Fig. 3.7(b)]. Fig. 3.7(c) demonstrates a coarse version of image which is reconstructed at the end-user. As this figure shows, all four ROIs

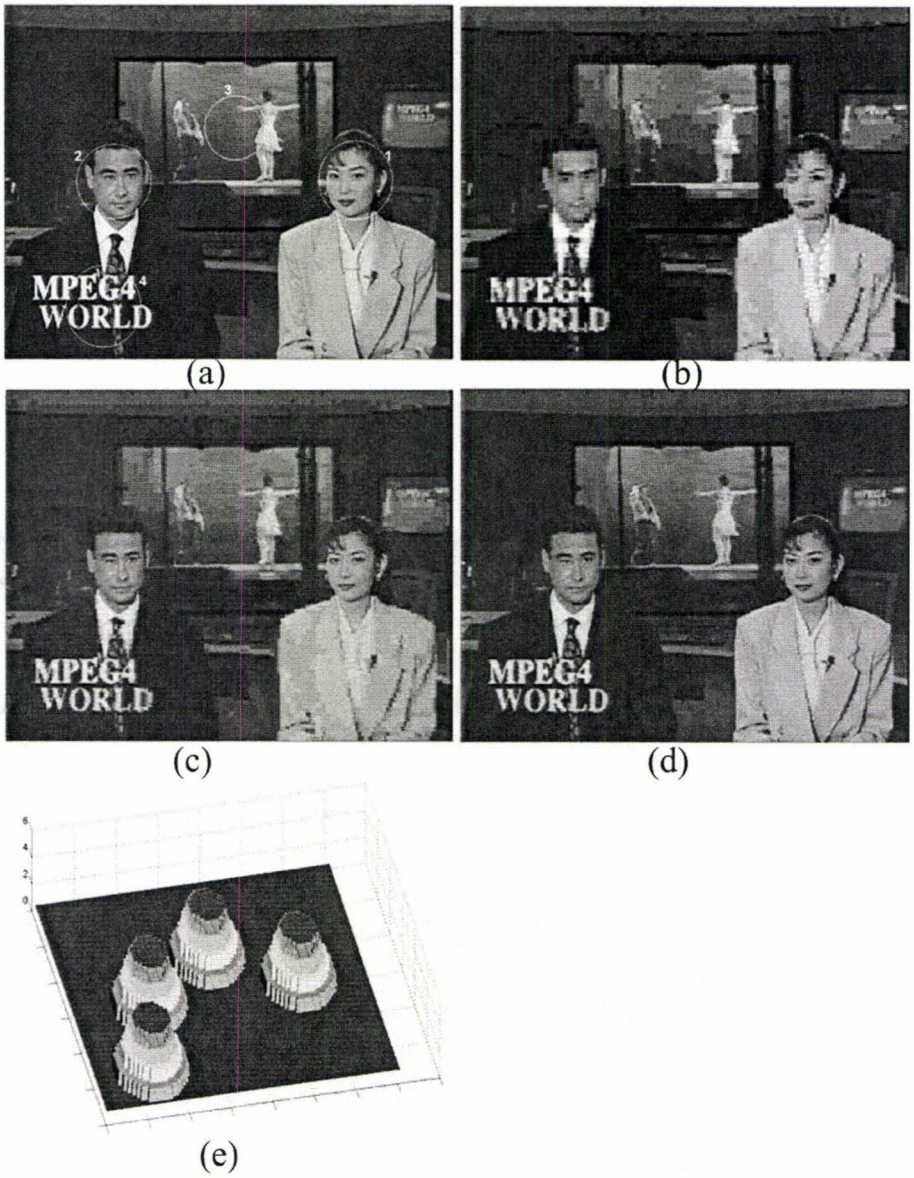


Figure 3.7: Progressive multiple ROI image coding results for "news" image using RVQ based method. Vector dimension is 16 (blocks of 4 $\times$ 4 pixels) and codebook size in each stage is 32. The maximum number of stages is 5. (a) Original "news" image 8 bpp with 4 indicated ROIs . (b) Initial coarse version of image with 0.32 bpp (c) Reconstructed image by the progressive method in the middle of transmission. $R_1 = 0.15$ and $\alpha = 1.1$ with 0.65 bpp. (d) Final reconstructed image by 5-stage RVQ scheme, with 1.6 bpp (e) Spatial RVQ stage status for reconstructed image in part c

have been reconstructed at higher quality compared to the rest of the image. Here $\alpha = 1.1$ and $R_1 = 0.15$ and the image is reconstructed by 0.65 bpp. Fig. 3.7(d) shows the final reconstructed image by 1.6 bpp. The number of RVQ layers used in Fig. 3.7(c) is shown in Fig. 3.7(e).

Fig. 3.8 shows the rate-distortion behavior of the RVQ and mean removed RVQ schemes for two different images, that were not used for training VQ codebook. Here we use $\alpha = 1$, $R_1 = 0.25$, and we change the bit-rate by increasing the RVQ stages used in ROI. According to Fig. 3.8(a) and (b), RVQ with 8×8 image blocks can provide a lower minimum affordable rate than that for RVQ with 4×4 image blocks (the left end of each curve represents the rate and the distortion corresponding to the coarsest image version provided by one layer RVQ). Since the coarse first layer of mean-removed RVQ is sent by scalar quantizer, the minimum obtainable rate by this method is less than the rate obtained by ordinary RVQ method. The overall rate-distortion performance of mean-removed RVQ is also better than that of the ordinary RVQ. Similar results were obtained for the second test image "mobile" as shown in Fig. 3.8(c) and (d).

Fig. 3.9 shows the effect of search number (M) in jointly optimized RVQ on the rate distortion behavior of this quantization scheme for the "news" image. Here, we have considered the maximum path number of $N_c = 30$. It can be seen that the rate-distortion performance of 2-stage JORVQ is considerably better than 1-stage JORVQ (i.e. ordinary RVQ). But the difference between performance of 3-stage JORVQ and 2-stage RVQ is not that much. On the other hand, increase in the search number M escalates the computation and the memory cost drastically. Thus, the figure represents a rate-distortion-complexity behavior for JORVQ.

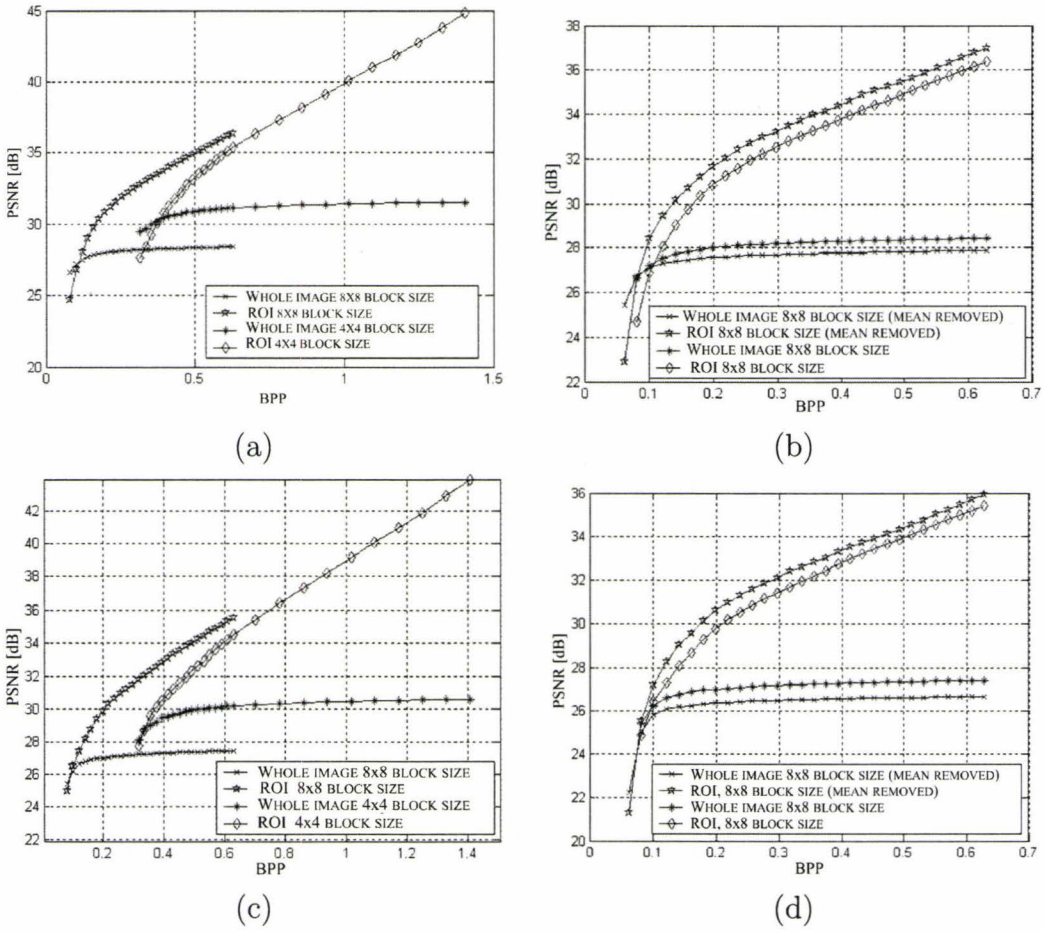


Figure 3.8: Rate-distortion behavior of the RVQ and mean removed RVQ schemes applied on two different images (a) RVQ rate-distortion performance for "news" image (b) Comparison between rate-distortion of RVQ and mean removed for "news" image RVQ (c) and (d) Same as (a) and (b) respectively but for "mobile" test image

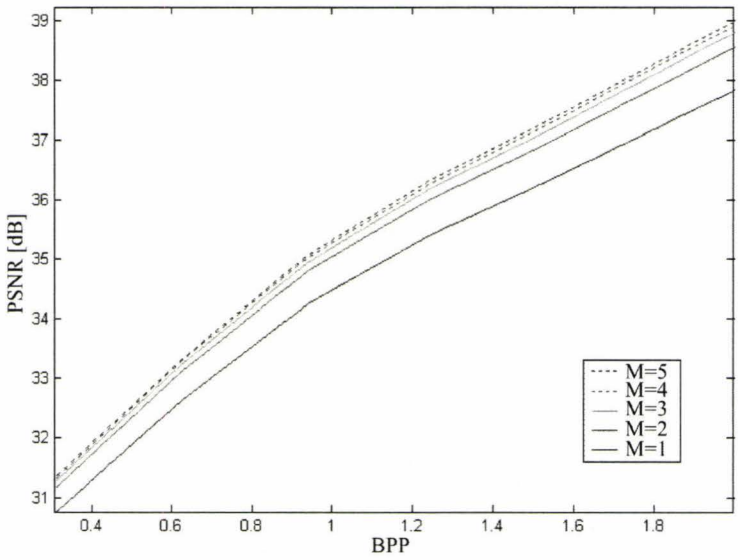


Figure 3.9: Rate-distortion-complexity behavior of JORVQ. $N_c = 30$ for the "news" image

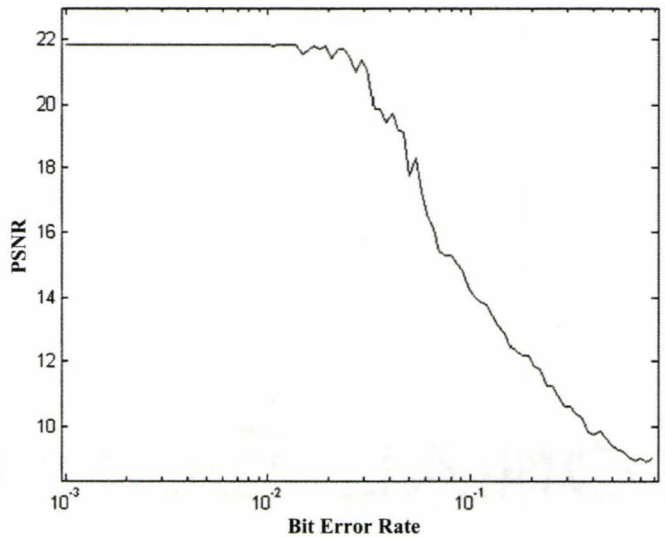


Figure 3.10: Reconstructed image quality behavior of an unequal error protected stream in the presence of channel bit error

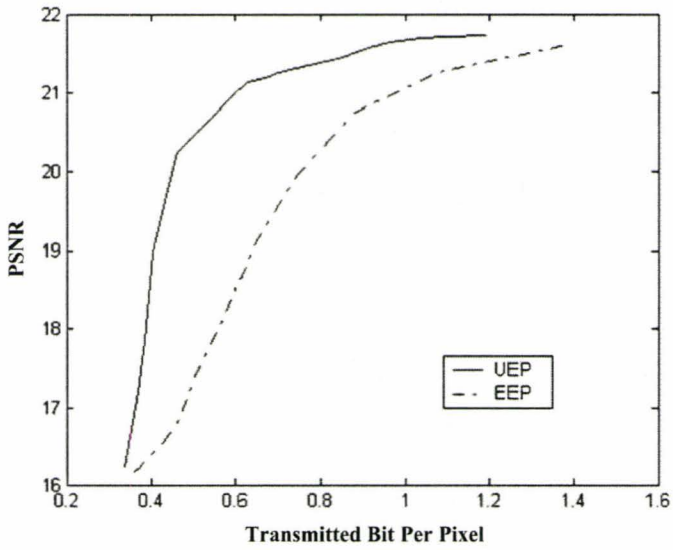


Figure 3.11: Rate-distortion behavior of UEP against equal error protection

Index assignment of the VQ codebook can be an efficient attempt for reducing the effect of channel error with no computation or rate cost. In this regard, index assignment is in such a way that the indices with the Hamming distances of one (different in just one bit) should represent vectors with maximum possible similarities. However, for further protection of bit-streams against channel noise, we may use unequal error protection. Fig. 3.10 shows the behavior of the reconstructed image quality in the presence of channel bit error. Bit-stream belongs to the truncated ROI image coded in Fig. 3.6(c) with rate of 0.16 and PSNR=21.8 dB. The bit-stream is divided into three sub-bit-streams. The first, the second ,and the third sub-bitstreams are unequally error protected by Reed-Solomon codes rs(31,15), rs(31,21), rs(31,27), respectively. Reed-Solomon encoding and decoding are done by MATLAB functions using symbols of length $m = 5$ bits. The figure shows a perfect protection of the bit-stream for the bit error rates less than 0.01.

Fig. 3.11 illustrates the advantage of UEP to the equal error protection in

terms of rate-distortion behavior. The input to the channel encoder is the bit-stream used for the pervious figure. For the UEP case, the bit-stream is divided into three sub-bitstreams. Each sub-bitstream is then channel coded using Reed-Solomon codes and error protected more than the next sub-bitstream. For the equal error protection case, the whole bit-stream is applied to Reed-Solomon coder. The channel error rate for this figure is 0.05.

Chapter 4

MP based ROI Image Coding

The work in this chapter focuses on a novel scalable, and progressive ROI image compression scheme based on matching pursuit (MP) algorithm [42,55]. Like residual vector quantization in the previous chapter, Matching pursuit (MP) is a multi-stage signal analysis method and can be used to render a selected region of an image with a specific quality. The method is capable of providing a trade off between rate, distortion, and complexity. The method also provides an interactive way of information refinement for regions of image with higher receiver's priority. By selecting a proper subset of the huge initial MP dictionary, using the method described in this work, the complexity burden of MP analysis can be adapted to the computational power of the image encoder.

This chapter is organized as follows. Section 4.1 details the proposed MP-based ROI image-coding scheme. In this section, an interactive approach to the quality refinement of an image is introduced. Rate-distortion-complexity trade-off for the ROI image-coding scheme is also discussed in this section. Simulation results are presented in section 4.2. Also, the effect of MP coefficient quantization and the size of MP dictionary on the rate-distortion

performance of the proposed method is reported in this section.

4.1 ROI Image Coding

4.1.1 System Overview

Section 2.2 comprises a detailed description for MP algorithm and its fundamentals. Here, an overview of MP-based image-coding system is presented. An image is initially divided into blocks or tiles and each image block is treated as an input signal (vector) to the MP analysis/synthesis system. Fig. 4.1(a) shows a schematic block diagram of MP based encoder and decoder. An image block (input signal) $\mathbf{X} = \mathcal{R}^{(0)}\mathbf{x}$ is applied to the first MP analysis stage (MP1), where $\mathcal{R}^{(0)}\mathbf{x}$ is the initial MP residual vector. Here, the signal is compared with all dictionary members $\mathbf{g}_{I^d}$ and dictionary element $\mathbf{g}_{I^{d(1)}}$, which yields the maximum magnitude inner product with $\mathcal{R}^{(0)}\mathbf{x}$, is chosen. Since we can only handle the transmission of analysis parameters with finite precision, the inner product coefficient

$$C^{(1)} = \langle \mathbf{X}, \mathbf{g}_{I^{d(1)}} \rangle = \sup_{I^d \in \mathcal{I}^d} \langle \mathcal{R}^{(0)}\mathbf{x}, \mathbf{g}_{I^d} \rangle \tag{4.1}$$

is quantized to $\tilde{C}^{(1)}$. In order to reduce the quantization error effect of inner product coefficients on MP analysis performance, the quantized version of inner product coefficient $\tilde{C}^{(1)}$ is then employed to calculate the residual signal $\mathcal{R}^{(1)}\mathbf{x} = \mathbf{X} - \tilde{C}^{(1)}\,\mathbf{g}_{I^{d(1)}}$. The quantized coefficient $\tilde{C}^{(1)}$ along with the index of the selected dictionary element $I^{d(1)}$ is sent to the decoder. Now, $\mathcal{R}^{(1)}\mathbf{x}$ is the input to the second MP analysis stage (MP2). The resulting $\tilde{C}^{(2)}$ and $I^{d(2)}$ are transmitted and $\mathcal{R}^{(2)}\mathbf{x} = \mathcal{R}^{(1)}\mathbf{x} - \tilde{C}^{(2)}\,\mathbf{g}_{I^{d(2)}}$ is employed for the next MP analysis stage. This procedure continues and the corresponding analysis parameters are sent. Fig. 4.1(b) illustrates the above process at the r^{th} MP

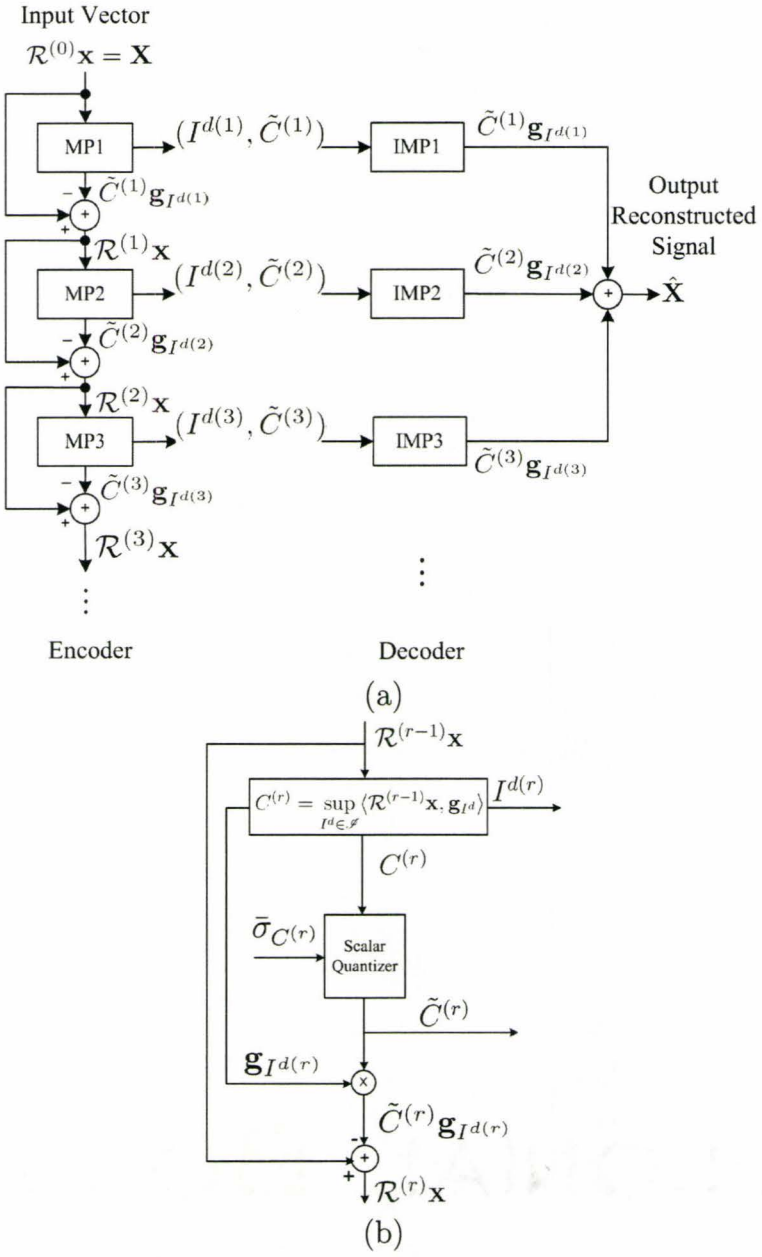


Figure 4.1: (a) Schematic block diagram of matching pursuit encoder and decoder (b) More detailed block diagram of r^{th} stage of MP analyzer

analysis stage. As shown in this figure, the scalar quantizer in each stage is adapted based on the standard deviation of the inner product coefficients in that MP analysis stage. Details of the employed quantization method in this chapter are given in the next section. At the receiver side the signal $\mathbf{X}$ is reconstructed using the received information $\tilde{C}^{(i)}$, and $I^{d(i)}$ by the following linear combination formula:

$$\tilde{\mathbf{X}} = \sum_{i=1}^{m} \tilde{C}^{(i)} g_{I^{d(i)}} \tag{4.2}$$

where m is the number of MP analysis stages.

4.1.2 MP Based ROI Image Coding

The capability of providing a multiresolution approximation for image blocks makes matching pursuit algorithm a proper choice for ROI image coding. The basic idea for MP based ROI image coding is to extract more structural features for the regions of the image with higher level of viewer's interest. To apply this intuitive idea, an image is divided to image blocks or tiles. The level of interest for an image block is considered to be a function of the distance of the image block from the point of interest. Our proposed interactive MP based ROI image coding works as follows. A coarse version of the image is transmitted to the receiver. The receiver then indicates his/her point of interest (i.e., center of ROI) and sends the corresponding coordinates of the point to the transmitter. Following this interaction, the transmitter sends refinement information for each image block according to its level of importance. The level of importance is measured by the distance of image block centroid from the point of interest. To generate a progressive bit-stream, the refinement information related to image blocks inside the ROI has to have higher transmission priority. After MP analysis and transmission of image

blocks belonging to ROI, the background image blocks are also refined. This progressive streaming algorithm provides bit-rate and quality scalability, i.e., the bit-stream can be truncated at a desired point in the middle of refinement process and a quality corresponding to the received information is obtained.

The above-mentioned approach can be implemented for MP based ROI image coding in such a way that the quality of image gradually reduces from the ROI to the background. To achieve this image refinement manner, the encoder calculates the distance of the j^{th} image block from the point of interest (d_j), where $j = 1, 2, 3, \ldots, N_B$ is image block index and N_B is the total number of blocks in the image. Let P_j represents the number of MP stages by which the j^{th} image block is coded. The MP algorithm proceeds on j^{th} image block and encodes it with $(P_j + 1)^{th}$ MP stage if d_j is less than a distance metric R_k which is defined for k^{th} level of image refinement. R_k may be assumed as the radius of a virtual circle whose center locates at the point of interest (see Fig. 3.1). The circle encloses centroids of image blocks that are eligible for the k^{th} stage of MP refinement. This analysis scheme provides gradual image quality change from the ROI to the background. The distance metric R_k can be adjusted based on the network condition.

The proposed progressive ROI image coding based on matching pursuit is very similar to the one proposed for RVQ based ROI image coding in the previous chapter. In order for this chapter to be self-contained, the details of the proposed progressive ROI coding method is also included in this chapter. By transmitting the mean values of image blocks, a coarse version of the image is reconstructed at the receiver. The receiver sends the ROI information (the coordinates of the center of ROI). The encoder checks all the image blocks to see if the distances of their centroid from the point of interest (d_j) is less than an initial distance value of R_1. If $d_j > R_1$, no refinement information is

sent for j^{th} image block. If $d_j \leq R_1$, the residue of the j^{th} image block with respect to the existing version of it at the receiver is approximated using the first stage of the MP analysis. To further improve the quality of the ROI, using the relation

$$R_{k+1} = \alpha \times R_k \tag{4.3}$$

the encoder updates the R_k with R_{k+1}. The above routine is repeated for all image blocks using R_{k+1} as the new distance measure. This process results in generation of successive MP refinement layers of the image with more emphasize on the region of interest. The updating multiplier α, in equation (4.3), specifies how gradual the image quality is reduced from ROI toward the background. This parameter plays an important role in emphasizing the ROI visual information. An appropriate value has to be assigned to α in order to meet the image transmission bit budget and, at the same time, give a desired level of reconstruction fidelity to the ROI compared to the background. If the ROI has an absolute transmission priority and the bit budget is small, α can be set to 1. This implies that the refinement bits are sent merely for the region of interest. If a larger value is assigned to α the distance metric R_k grows more rapidly and more image blocks become eligible for the refinement. The virtual circle with radius R_1 and centered at the point of interest is the initial region which its image blocks are enhanced with the highest priority. Since a bigger R_1 means more initial image blocks with this privilege, there would be no ROI if R_1 exceeds the size of the image. Practical values for R_1 can be in the range of 1/10 to 1/2 of image size. Thus, in case of low bit budget or low bit-rate transmission, α is set close to 1 and R_1 is selected in the above mentioned range and a higher priority of transmission will be given to the ROI image blocks.

The complete bit stream generated by above algorithm conveys refinement information for all image blocks with a final quality dictated by the maximum MP analysis stages. The bit stream can be truncated whenever a desirable image quality is provided at the receiver. In this case, the reconstructed image has better quality at region(s) of interest.

The proposed MP-based ROI image coding method allows changing the position of point of interest without losing the previously received refinement data. In order for the transmitter to handle the ROI change, the number of MP stages that each image block has been analyzed with, should be recorded at the encoder. Receiver sends the location of the new point of interest. After this interaction, the transmitter sends the refinement data for image blocks around the newly defined point of interest without the need to retransmit the already sent information. More specifically, let $P_j^{(-1)}$ denote the number of MP stages by which the j^{th} block is analyzed. Let d_j represent the distance of the j^{th} image block from the new point of interest. If $d_j < R_k$, encoder checks $P_j^{(-1)}$ and if $P_j^{(-1)} \geq k$ no transmission for j^{th} image block is required, otherwise, the residue of the image block is approximated using the k^{th} stage of the matching pursuits and the index of chosen dictionary element and the corresponding inner product coefficients are transmitted and $P_j^{(-1)}$ is updated.

Besides the possibility of ROI location change during the course of transmission, multiple ROI can also be coded in the proposed MP-based ROI image coding scheme. In the multiple points of interest case d_j is calculated as follows:

$$d_j = \min_i d_j^{(i)} \quad i = 1, 2, ..., N_{ROI} \quad j = 1, 2, ..., N_B \tag{4.4}$$

where $d_j^{(i)}$ is the distance of the j^{th} image block from i^{th} point of interest and N_{ROI} is the number of ROIs.

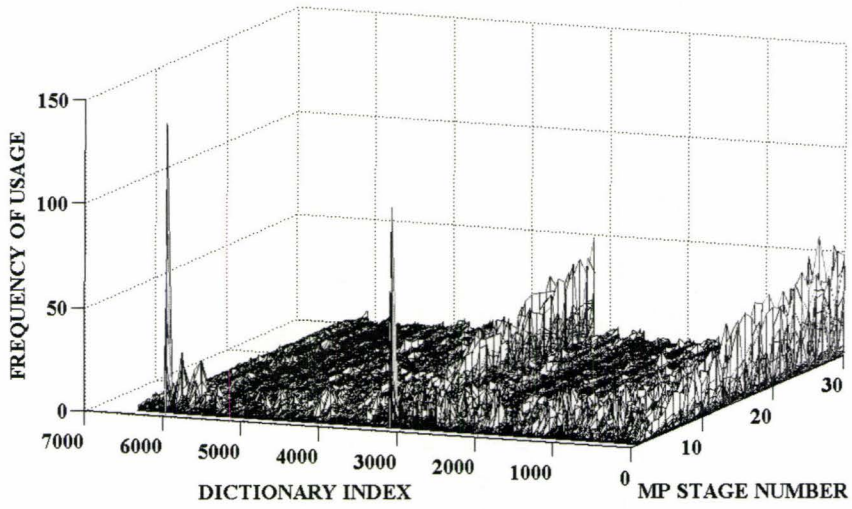


Figure 4.2: Dictionary elements histogram for different MP stages when MP applied on a set of 8 different images

4.1.3 Computational Complexity Reduction

The size of MP dictionary has a direct effect on the computational complexity of the MP algorithm. A larger dictionary requires more bits to represent the dictionary index than that for a smaller one. On the other hand, a properly designed larger dictionary would better represent signal structures. Thus, when we choose a dictionary size for MP analysis, there are trade-offs between rate, quality, and complexity.

Here, a method for reducing the computational complexity of MP-based ROI image coding is proposed. A huge initial dictionary is used for MP analysis of a large number of test images. According to the residual patterns of image blocks, in each MP stage, different subsets of the dictionary are used more often. We can notice this fact by observing the histogram of dictionary elements when a set of different test images are analyzed using matching pursuit method.

Fig. 4.2 shows the usage frequency (histogram) of dictionary elements as a function of matching pursuit stage number when MP is applied to a set of 8 different images for a Gabor dictionary with 6400 element of 8×8 pixels. As this figure displays, in each MP analysis stage, the usage frequency of dictionary elements is largely skewed and a significant number of elements are almost never used. This is due to the fact that in each stage of MP analysis of an image block, different structures exist that are not necessarily similar to structures in other stages and in order to extract these structures, different subsets of dictionary are used more frequently. Using this prior knowledge, the computational complexity of MP-based ROI image coding can be reduced. The idea is to sort the dictionary for each MP stage based on the usage frequency of dictionary elements for that stage. The original and sorted dictionaries have exactly the same elements, but elements of the latter are sorted by their usage frequency in a descending manner. According to our rate-quality-complexity choice of working point, an N_D element subset of the original dictionary can be selected in each MP stage simply by taking N_D first elements of the corresponding sorted dictionary. Only the corresponding sub-dictionary is searched for the most similar element to the residual signal at each MP stage. Since saving multiple sorted versions of the original huge dictionary for each MP stage is inefficient, in terms of memory requirement, for each stage of matching pursuit, a transition vector is used to map the index from the original dictionary to the sub-dictionary for that stage. This transition vector reduces the storage requirement for the sorted dictionary and the dictionary inner products (used in Mallat's fast algorithm, presented in section 2.2) to only the original ones. The i^{th} element of the transition vector for the k^{th} MP stage is a pointer to an element in the original dictionary which is identical to the i^{th} element in the sorted dictionary for the k^{th} MP stage. In other words, in order to point

to the i^{th} element of the sorted dictionary in k^{th} MP stage the index in the original dictionary (i_{orig}) can be found using the following relation:

$$i_{orig} = trans^k(i) \tag{4.5}$$

In (4.5), $trans^k(\cdot)$ represents the transition vector for k^{th} MP stage. The formula can also be used when the elements of the dictionary inner product matrix are required for the Mallat's fast matching pursuit algorithm. To point to the element (i, j) of the inner product matrix of the sorted dictionary at k^{th} MP stage, we simply find the element (i_{orig}, j_{orig}) of inner product matrix associated to the original dictionary, without the necessity of storing another version of the original huge $N_{\mathscr{D}} \times N_{\mathscr{D}}$ matrix ($N_{\mathscr{D}}$ is the number of elements in the original dictionary). This task can be done using

$$(i_{orig}, j_{orig}) = (trans^k(i), trans^k(j)) \tag{4.6}$$

According to Mallat's fast updating formula (2.23), each inner product coefficient $\langle \mathcal{R}^{(r+1)}\mathbf{x}, \mathbf{g}_{I^d} \rangle$ (for every $I^d \in \mathscr{I}^d$) can be calculated only if $\langle \mathcal{R}^{(r)}\mathbf{x}, \mathbf{g}_{I^d} \rangle$ is already calculated in the previous MP stage. Thus, in case a given $\mathbf{g}_{I^d}$ is not included in the sub-dictionary for the stage r, the fast algorithm is not able to calculate $\langle \mathcal{R}^{(r+1)}\mathbf{x}, \mathbf{g}_{I^d} \rangle$ and this value should be computed using direct inner product operation. Fig. 4.2 shows that higher MP stages behave almost the same as far as the usage frequency of dictionary elements is concerned. Thus for higher MP stages the same transition vector, and sub-dictionary, can be used. Our dictionary truncation method not only reduces the computational complexity, it also lowers the required bit-rate by diminishing the number of bits representing each dictionary index.

4.2 Experimental Results

In this section we present the simulation results for the proposed MP based ROI image coding technique. First, the choice of MP dictionary and the inner product coefficient quantization method are explained.

4.2.1 Dictionary characteristics

Matching pursuit technique in general and the proposed MP-based ROI image coding in particular place no restriction on the choice of dictionary. For the following experimental results we choose over-complete 2-D separable Gabor dictionaries with different dimensionality (i.e., 4×4, 8×8, and 16×16) similar to what is used in [67]. We do not claim this choice of dictionary is optimum, it has been used in order to show the functionality of the proposed ROI image coding method based on matching pursuit analysis. However, Gabor 2-D dictionaries are very popular for matching pursuit analysis purpose. Besides, by choosing a very large dictionary, and selecting subsets of it for each MP analysis stage (as characterized previously), the coherence of the dictionary elements with the residues in each MP stage can be increased, if a dictionary with a specific size is targeted.

The 2-D ($N \times N$) separable Gabor functions are defined in terms of a Gaussian prototype window

$$g(t) = \sqrt[4]{2} e^{-\pi t^2}. \tag{4.7}$$

By employing this function, a set of scaled, modulated, phase shifted, and translated two dimensional functions is generated

$$G_{\vec{\beta}^{(1)},\vec{\beta}^{(2)}}(i,j) = K_{\vec{\beta}^{(1)},\vec{\beta}^{(2)}} g_{\vec{\beta}^{(1)}}(i) g_{\vec{\beta}^{(2)}}(j) \tag{4.8}$$

	4×4 image blocks			8×8 image blocks			16×16 image blocks		
k	s_k	ξ_k	ϕ_k	s_k	ξ_k	ϕ_k	s_k	ξ_k	ϕ_k
1	1	0	0	1	0	0	1	0	0
2	3	0	0	5	0	0	5	0	0
3	6	0	0	9	0	0	10	0	0
4	10	0	0	14	0	0	30	0	0
5	1.3	1	$\pi/2$	20	0	0	60	0	0
6	4	1	$\pi/2$	1.4	1	$\pi/2$	120	0	0
7	7	1	$\pi/2$	5	1	$\pi/2$	1.4	1	$\pi/2$
8	10	1	$\pi/2$	12	1	$\pi/2$	10	1	$\pi/2$
9	5	1	$\pi/4$	16	1	$\pi/2$	55	1	$\pi/2$
10	10	1	$\pi/4$	20	1	$\pi/2$	140	1	$\pi/2$

Table 4.1: Gabor 2-D dictionary parameters associated to 4×4, 8×8, and 16×16 block sizes

where

$$g_{\vec{\beta}^{(l)}}(i) = g\left(\frac{i - u^{(l)}}{s^{(l)}}\right) \cos\left(\frac{2\pi\xi^{(l)}(i - u^{(l)})}{N} + \phi^{(l)}\right) \tag{4.9}$$

and $i, j \in \{0, 1, \ldots, N-1\}$. $\vec{\beta}^{(l)} = (s^{(l)}, \xi^{(l)}, \phi^{(l)}, u^{(l)})$ is a quarter consisting of scale $s^{(l)}$, modulation frequency $\xi^{(l)}$, phase shift $\phi^{(l)}$, and translation $u^{(l)}$, where $l = 1$ or 2. Here, $K_{\vec{\beta}^{(1)},\vec{\beta}^{(2)}}$ is a normalization factor for each $(N \times N)$ two dimensional dictionary element. Table 4.1 shows a set of 10 values for the first three elements of $\vec{\beta}$ for different dimensionality of image blocks, i.e., 4×4, 8×8, and 16×16 pixels. The translation parameter u has N different values from 0 to $N - 1$, thus it is not entered into this table. Fig. 4.3 displays the 2-D, 8×8 Gabor dictionary elements when $u^{(1)} = u^{(2)} = N/2$, i.e., centralized 2-D Gaussian elements. For image blocks of size 4×4, 8×8, and 16×16, the dictionaries consist of 1600, 6400, and 25600 2-D Gabor functions respectively.

4.2.2 Quantization of MP inner product coefficients

The resulting inner product coefficients of MP analysis can be delivered to a digital medium only with finite precisions. Therefore, quantization of the

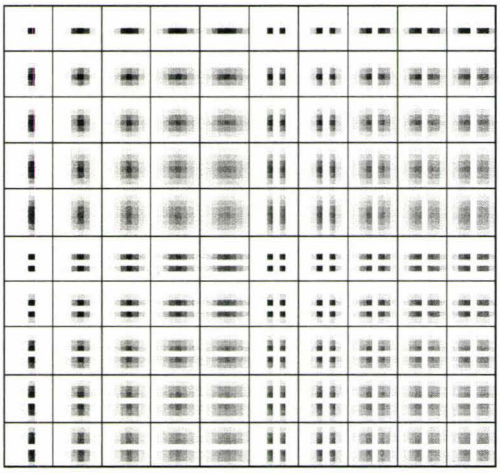


Figure 4.3: 2-D, 8×8 Gabor dictionary elements at fixed values of $u_1 = u_2 = N/2$, where $N = 8$

coefficients is an integral part of the proposed MP based ROI image coding. There are a lot of different quantization schemes to choose from and here a simple adaptive uniform scalar quantization scheme is employed. The step size of the scalar quantizer of each MP stage is adapted based on the standard deviation of the inner product coefficients in that stage.

According to [42], the energy of the MP inner product coefficients is an exponentially decreasing function of the number of MP analysis stages. Fig. 4.4 demonstrates experimentally calculated variance of inner product coefficients as a function of MP stage number for different image block sizes. To create this figure, the statistical data is extracted from MP analysis of a set of 8 different images. Since a larger image block involves with larger number of image pixels for 2-D inner product calculation than a smaller image block, 16×16 image block can take the highest values of variance for MP coefficients among other experimented block sizes i.e. 8×8 and 4×4 block sizes. This figure shows that variance of MP coefficients changes significantly as a function of MP stage. Therefore, it would be a better choice to adapt the step size

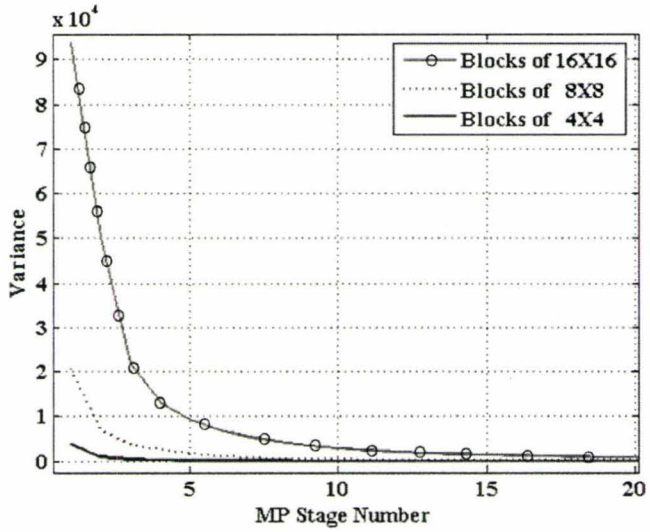


Figure 4.4: Variance of inner product coefficient as a function of MP analysis stage number for different sizes of image blocks

of scalar quantizer based on MP stage number.

For the experimental results, a mid-tread quantizer [21] with $N_Q = 16$ quantization steps is employed. Therefore, each MP inner product coefficient is represented by $B_Q = \log_2(N_Q) = 4$ bits. The granular region [1] of the quantizer is from $-3\sigma_{C^{(n)}}$ to $3\sigma_{C^{(n)}}$, where $\sigma_{C^{(n)}}$ is the standard deviation of inner product coefficients at n^{th} MP analysis stage. If the statistical distribution for the inner product coefficients conforms to a Gaussian model, the above-mentioned range is where 99% of the statistical data are located.

4.2.3 Region of interest image coding results

Fig. 4.5 demonstrates examples of progressive MP-based ROI image coding. Fig. 4.5(a) shows the original monochrome 8 bpp "news" image with indicated region of interest. To apply our method of ROI image coding, the image is divided into 8×8 non-overlapping image blocks. The MP dictionary contains 6400 2-D Gabor elements with size of 8×8. Fig. 4.5(b) is the coarse

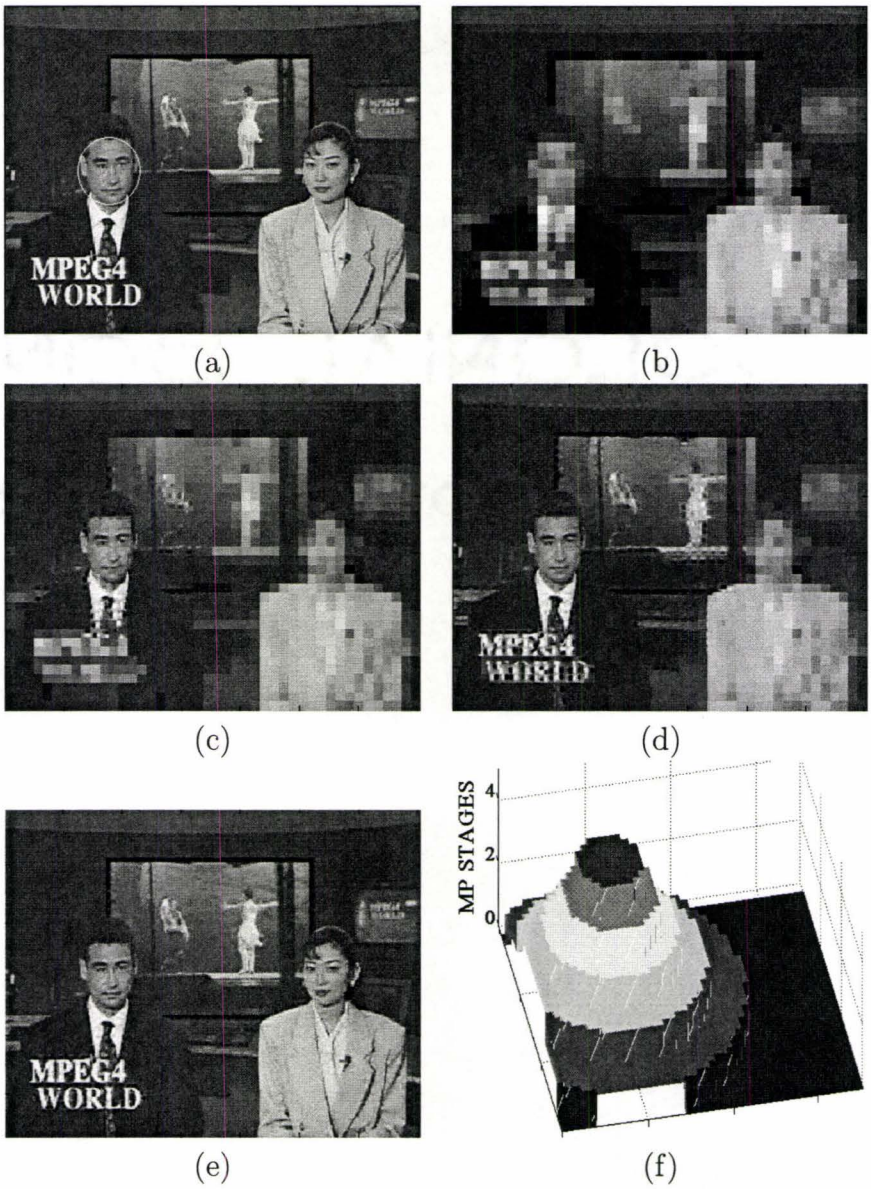


Figure 4.5: Progressive image enhancement using MP-based ROI image coding. The test image is "news" image, and image blocks are 8×8 pixels. The MP dictionary has 6400 2-D Gabor elements with the size of image blocks. Here the MP analysis is restricted to maximum 5 stages. MP inner products are quantized with a uniform 4-bit quantizer (a) Original monochrome 288×352 pixels "news" image represented by 8 bpp. The region of interest is indicated by a circle. (b) A coarse version of image, generated by mean value of each image block, is sent to the receiver with 0.0625 bpp. (c) MP-based ROI image coded version of image in early part of transmission with $R_1 = 0.125$ and $\alpha = 1.4$ with 0.1457 bpp. (d) Previous image after receiving more refinement bits at 0.3981 bpp. (e) Completely refined image by 5 stages of matching pursuit analysis bit-stream at 1.3628 bpp. (f) Spatial status of MP stages for part (d)

version of "news" image formed by mean value of each image block at 0.0625 bpp. The receiver reconstructs this version of image and indicates the region of interest by sending the coordinates of the point of interest to the transmitter. At this stage, the transmitter starts sending refinement bits generated by the proposed MP based ROI image coding. In figure 4.5(c), the reconstructed image at an early stage of progressive transmission is shown. The analysis parameters are $R_1 = 0.125$ (the unit here is the width of the image), and $\alpha = 1.4$. The image is progressively reconstructed at 0.1457 bpp. As time passes, more refinement bits participate in image enhancement. Fig. 4.5(d) shows how the reconstructed image looks when the overall rate becomes 0.3981 bpp. In Fig. 4.5(e), we receive the complete bit-stream at 1.3628 bpp. In this figure, all image blocks are analyzed with a maximum of $N_s = 5$ MP stages. Fig. 4.5(f) demonstrates the number of MP analysis stages that have been used in different locations of the image. This MP stage status presentation is associated with Fig. 4.5(d).

The proposed method of ROI image coding is able to handle multiple regions of interest. Fig. 4.6 demonstrates this ability of our MP-based ROI image coding scheme. The original "news" image is shown in fig. 4.6(a) with three different regions of interest indicated by circles. The MP-based ROI image coding setting is exactly like the one in Fig. 4.5 (i.e., $\alpha = 1.4$, $R_1 = 0.125$, 8×8 image blocks, and a 6400-element MP dictionary of 2-D Gabor functions). A coarse version of image generated by the mean values of image block is reconstructed at the receiver end [Fig.4.5(b)]. According to this version of image, the receiver requests more enhancement bits by sending the coordinates of the regions of interest. The refinement information is sent to the receiver in a progressive fashion and with the priority of the ROIs. Fig. 4.6(b) shows the received version of image in the early stages of transmission.

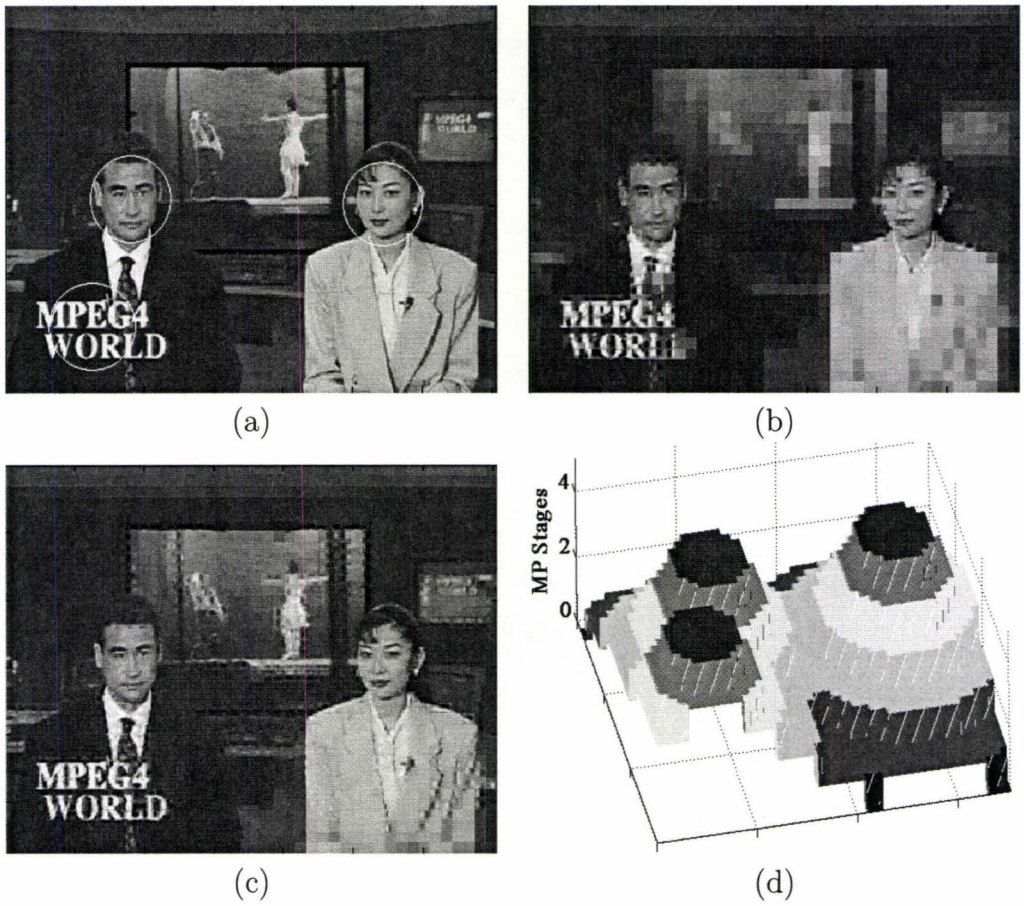


Figure 4.6: Multiple-ROI image coding based on MP (a) Original test image "news", with three regions of interest marked by circles. (b) The reconstructed image in early stages of transmission at 0.1566 bpp. The analysis parameters are $\alpha = 1.4$ and $R_1 = 0.125$. (c) Another version of the reconstructed image at 0.5826 bpp. (d) Spatial MP stage status associated with (c)

Here the image is reconstructed at 0.1566 bpp. Fig. 4.6(c) and (d) show a higher quality version of the image at 0.5826 bpp and its spatial MP stage status presentation, respectively.

The proposed MP-based ROI image coding scheme is also able to handle the change of ROI choice during the course of transmission. Fig. 4.7 illustrates this capability of the method. Fig. 4.7(a) shows the original "news" test image with two successive regions of interest marked by circles. The left hand side

ROI is the first choice. The analysis parameters are $\alpha = 1.4$, and $R_1 = 0.125$. In Fig. 4.7(b) the image reconstructed based on the first choice of ROI, when the receiver requests ROI change, is displayed. At this point the image is reconstructed at 0.1424 bpp. Fig. 4.7(c) shows the reconstructed image after receiving some refinement bits according to the second choice of ROI at 0.4544 bpp. The corresponding MP stage status in different parts of image is shown in Fig. 4.7(d).

4.2.4 Effects of image block size

To implement the proposed ROI image coding, different sizes of image blocks may be chosen. Fig. 4.8 shows examples of MP-based ROI image coding for different choices of 4×4, 8×8, and 16×16 for image blocks. The MP dictionary for each image block size is tailored to that specific size. The size of 2-D Gabor dictionaries associated to 4×4, 8×8, and 16×16 image blocks are 1600, 6400, and 25600 respectively. The bit-streams are truncated in such a way that the total rates are almost 0.3 bpp. Figures 4.8(a), (b), and (c) are the reconstructed images for 4×4, 8×8, and 16×16 image blocks and with the overall PSNR of 23.8732 dB, 25.3791 dB, and 26.5264 dB respectively. The bit rate required to receive the initial coarse version of the image for each of the above mentioned block sizes are 0.25 bpp, 0.0625 bpp, and 0.0156 bpp respectively. As it can be seen from the figures, for a given rate, the reconstruction quality in the region of interest for 16×16 case is much higher than that for the 4×4 case while, the background quality for the latter is higher than the former.

Fig. 4.9 shows the effect of dictionary size change on the quality of the reconstructed image, analyzed by the proposed MP-based ROI image coding.

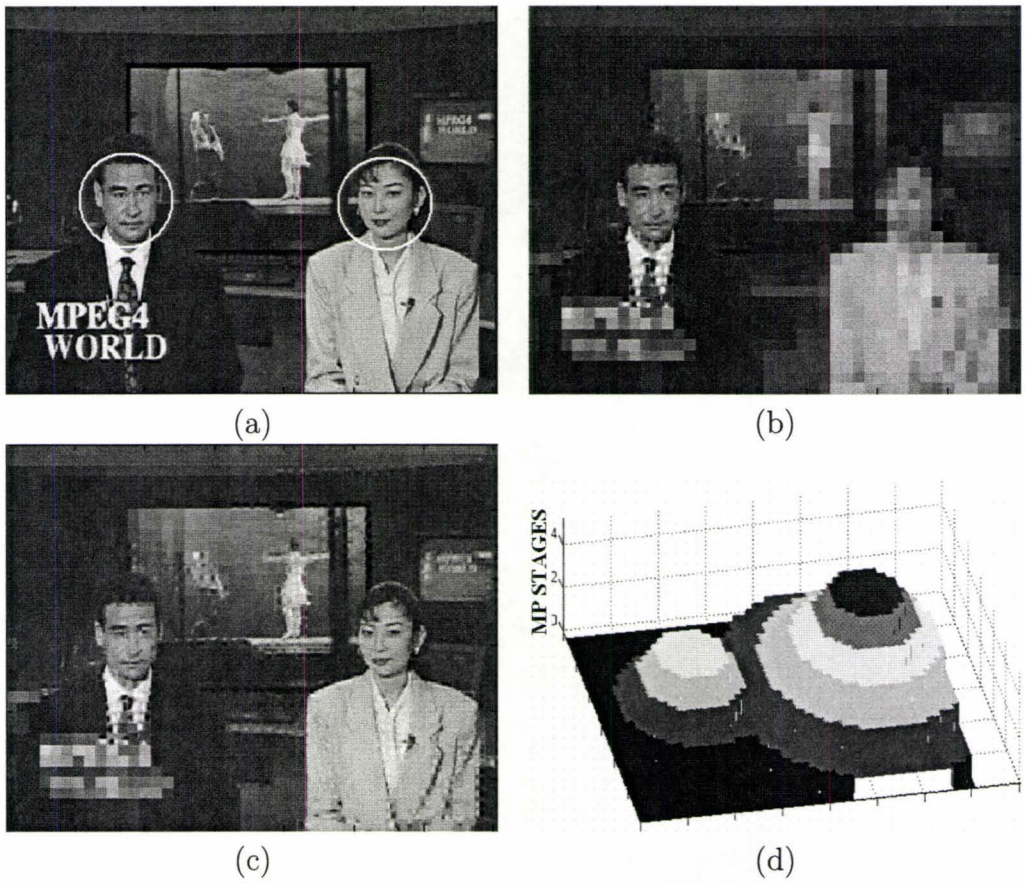


Figure 4.7: Changing the point of interest during the course of transmission. (a) Original "news" test image with two regions of interest marked by circles. (b) The refinement information according to the first ROI (left hand side ROI) is transmitted. At this point the receiver changes the ROI choice to the right hand side ROI. MP-based ROI analysis parameters are: $\alpha = 1.4$, and $R_1 = 0.125$. The image is reconstructed at 0.1424 bpp. (c) The refinement information is sent according to the new choice of ROI. The image is reconstructed at 0.4544 bpp. (d) Spatial MP stages status when image (c) is received

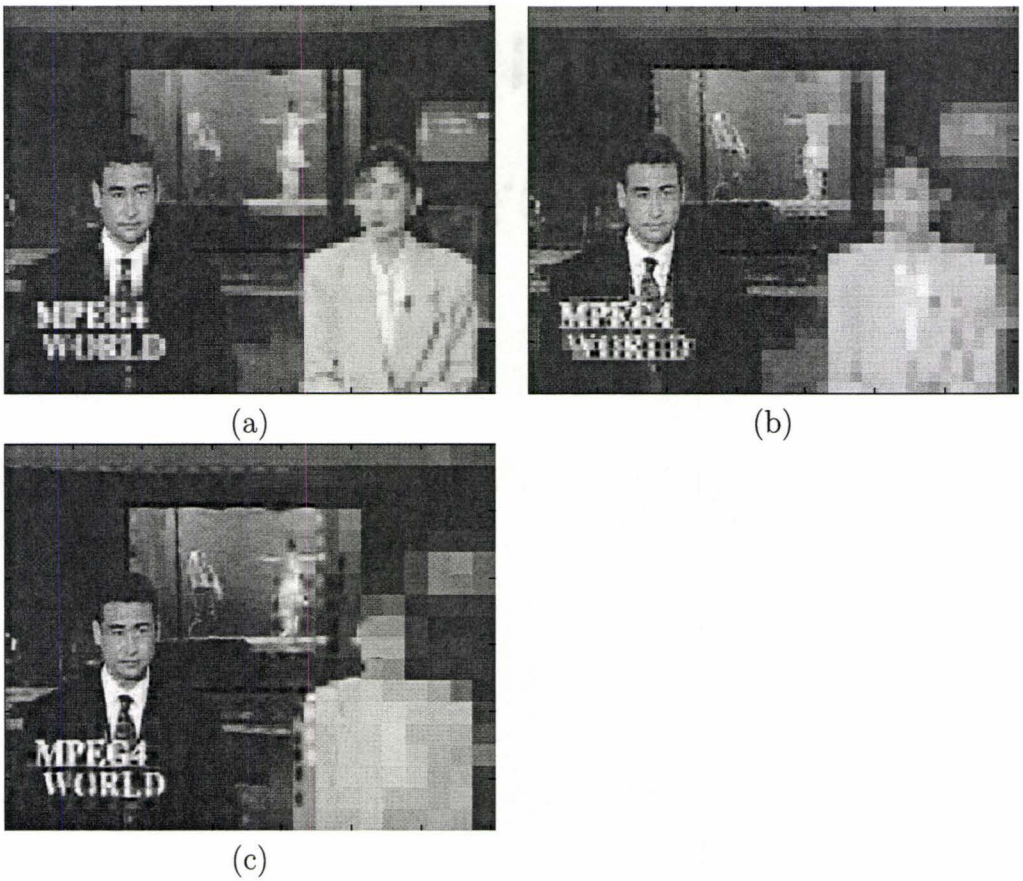


Figure 4.8: MP-based ROI image coding for different sizes of image blocks (the ROI is the one shown in fig. 4.5(a)). The bit-streams are truncated such that the overall rate for each case is about 0.3 bpp. (a) 4×4 image blocks and dictionary with 1600 elements. PSNR is 23.8732 dB and exact overall rate is 0.2977 bpp (b) 8×8 image blocks and dictionary of 6400 elements. PSNR is 25.3791 dB and exact overall rate is 0.2925 bpp(c) 16×16 image blocks and dictionary of 25600 elements. PSNR is 26.5264 dB and exact overall rate is 0.2942 bpp.

In order to sort the initial 2-D Gabor dictionary, a set of 8 images, different from our test "news" image, is used to provide the usage frequency when the images are analyzed by matching pursuit algorithm. According to the popularity of the dictionary elements, they are sorted in descending order for each MP stage. The corresponding transition vectors for each MP stage is then recorded to drastically reduce the memory requirement of having different versions of sorted dictionaries and the corresponding dictionary inner product matrices (used in Mallat's fast algorithm). If a sub-dictionary with N_D elements is required for an MP stage, the sorted dictionary corresponding to that stage is truncated to first N_D elements. Figures 4.9(a)-(e) show the MP-based ROI image coded versions of "news" test image for dictionaries of 4, 64, 256, 1024, and 4096 elements respectively. MP-based ROI analysis parameters are: $\alpha = 1.4$, and $R_1 = 0.125$. Since the image blocks are 8×8, the first sub-dictionary is an under-complete set. Thus, even if matching pursuit algorithm continues for infinite number of stages, there still would be approximation error. Although the second sub-dictionary seems to be complete, it does not have enough elements to extract feature of the signals properly and MP analysis may not converge fast. As the number of dictionary elements increases, the computational complexity of MP algorithm increases proportionally. Thus, for a choice of dictionary size of MP algorithm, there is always a trade off between the rate distortion behavior and the computation cost.

4.2.5 Rate-quality-complexity trade-off

The proposed MP based ROI image coding is able to provide compromises among bit rate, quality of the reconstructed image, and the computational complexity of MP analysis, as it is noted in the previous section. This problem

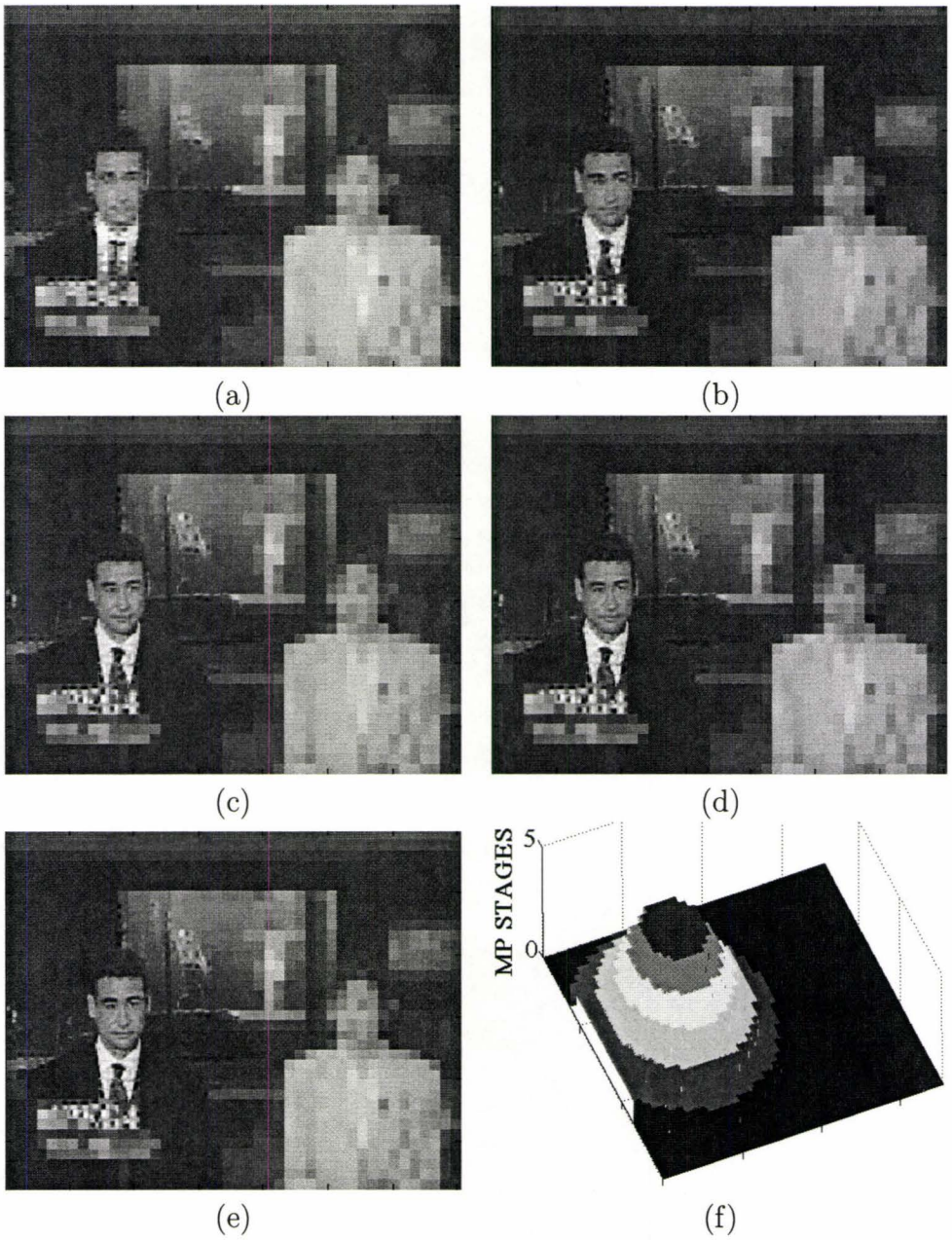


Figure 4.9: MP-based ROI image coding with resized dictionary. The original dictionary is 2-D Gabor of 6400 8×8 elements. Using transition vector for each MP stage, the dictionary is sorted and truncated to sub-dictionaries with N_D elements of 8×8 (a) $N_D = 4$, overall PSNR = 22.9069 dB, rate = 0.1189 bpp. (b) $N_D = 64$, overall PSNR = 23.5071 dB, rate = 0.1553 bpp. (c) $N_D = 256$, PSNR = 23.5781 dB, rate = 0.1739 bpp. (d) $N_D = 1024$, overall PSNR = 23.6011 dB, rate = 0.1925 bpp. (e) $N_D = 4096$, overall PSNR = 23.6092 dB, rate = 0.2110 bpp. (f) Spatial MP stage analysis status for all images

for a memoryless Gaussian source is studied in [66]. The knowledge of the inter-relation among rate, quality, and complexity is imperative to the selection of the best MP analysis set-up. The following results are based on average MP analysis and reconstruction results on a set of 8 different images. The analysis parameters, based on the proposed MP based ROI image coding, are $\alpha = 1.4$, and $R_1 = 0.125$. For different stages of MP algorithm, the original dictionary (i.e., 6400-element 2-D (8×8) Gabor dictionary) is sorted and truncated to a dictionary of size $N_D = 2^{B_D}$ elements of 8×8, 2-D Gabor functions.

Fig. 4.10 shows the quality variation of the reconstructed ROI coded images for ROI region and for whole image as function of number of quantization steps ($N_Q = 2^{B_Q}$) and MP dictionary size ($N_D = 2^{B_D}$) for a given fixed rate of 0.3 bpp. According to this figure quantization of inner product coefficients with 3 or 4 bits results in better quality performance when a fixed rate, i.e. 0.3 bpp, is targeted. Very fine quantization of inner product coefficients requires a large number of bits to be assigned to each coefficients and this results in small number of MP analysis stages for further refinement of residual vectors. On the other hand, very coarse quantization results in large quantization error. As it can be seen from the graphs, for a subset of original dictionary with just 4 elements, the quality performance is far worse than the other dictionary sizes. In this case, the dictionary is under-complete and not a good representative of all features of image blocks. According to this figure, the quality performance for $N_D = 64$, 256, 1024, and 4096 are comparable. Although bigger dictionary, i.e. more complexity, almost always yields better quality performance, the computational cost sometimes is too much. Taking into account the processing power of MP analyzer, the best dictionary size can be selected. Adding one more bit to the dictionary index bits means doubling the computational burden. The figure shows that for higher values of B_D (i.e.

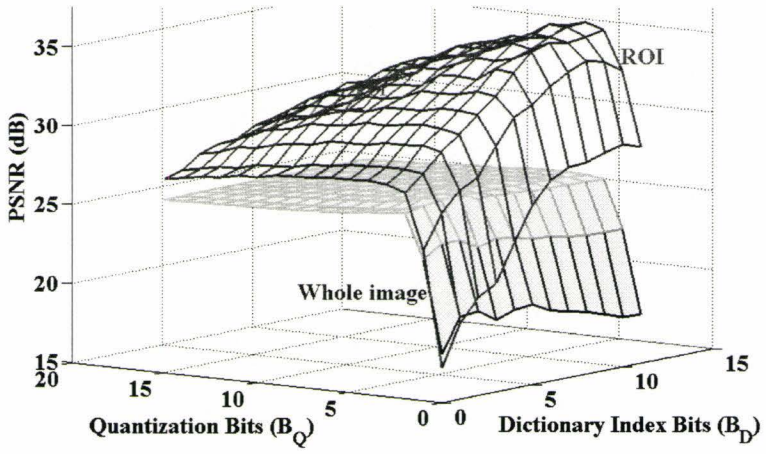


Figure 4.10: PSNR versus dictionary size and quantization step number for a fixed rate of 0.3 bpp

number of bits representing dictionary indices) little quality improvement is obtained considering the increase in the computational complexity.

Fig. 4.11(a) and (b) illustrate the quality of the output image of the proposed scheme as a function of bit-rate and the inner product coefficients quantization step number (B_Q). Here the dictionary size is set to $N_D = 2^9 = 512$. According to Fig. 4.11(a), the worst quality is obtained when the coarse version of image is sent with minimum number of quantization steps, i.e. $B_Q = 1$. Having only the coarse version of image at the receiver, the qualities at ROI and the whole image are almost the same. Refinement information separates the quality surfaces and improves ROI quality faster than the whole image. These two surfaces approach each other when the whole image is refined with maximum number of MP stages. Fig. 4.11(b) shows the effect of quantization step number on the rate-distortion performance of the method for ROI more distinctively. According to this figure, the best choice of quantization step number is $N_Q = 2^4 = 16$.

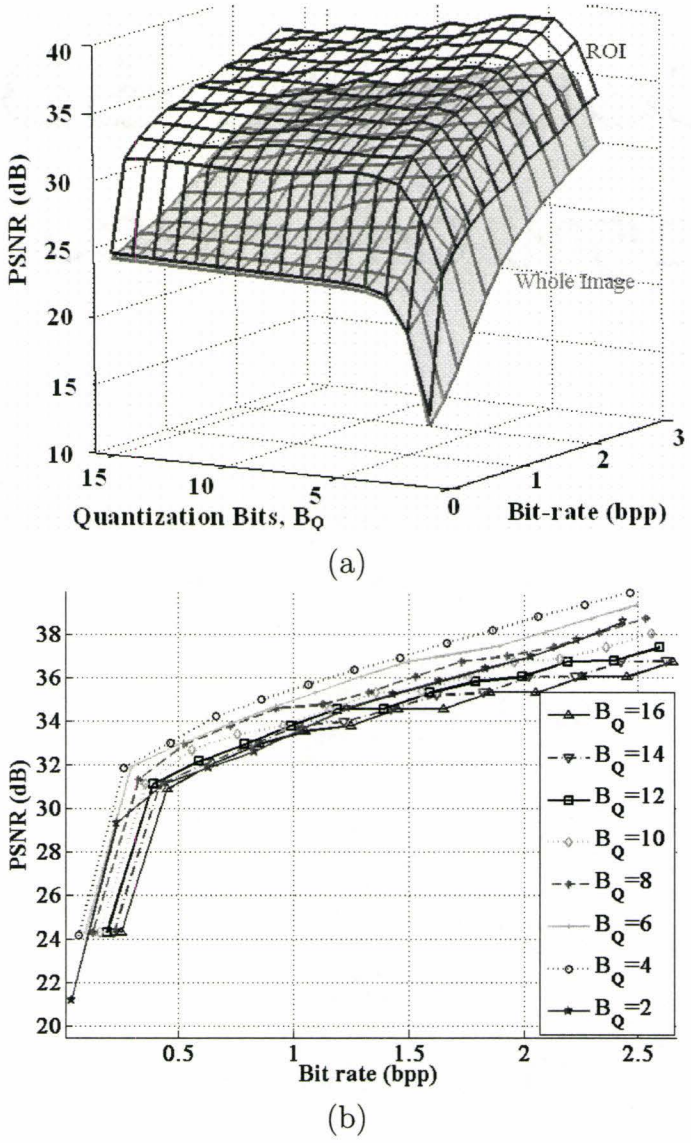
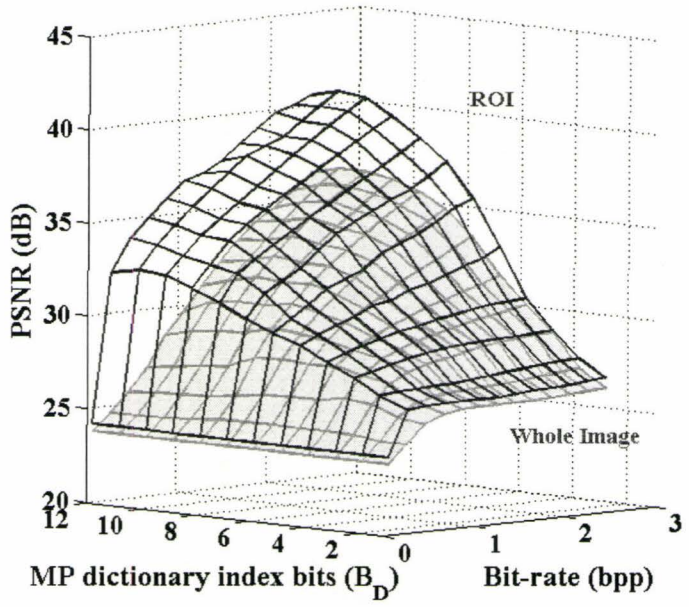


Figure 4.11: Rate-distortion performance of the MP based ROI image coding as a function of quantization step number of the MP inner product coefficients when the dictionary size is fixed to $N_D = 512$ (a) 3-D demonstration of quality-rate-quantization step number (b) 2-D demonstration of rate-distortion behavior of the proposed method in ROI for different amount of quantization step number.

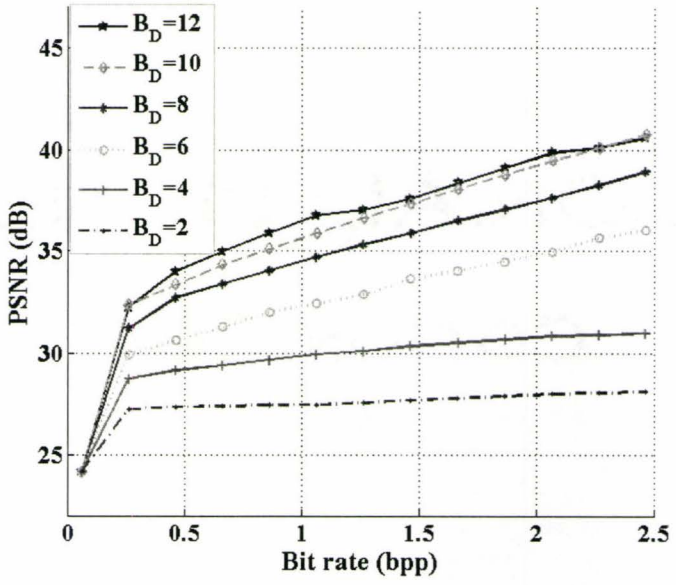
Fig. 4.12(a) and (b) demonstrate the rate-distortion behavior of the proposed ROI image coding scheme as a function of dictionary size, i.e. computational complexity. For these figures, the number of quantization steps is fixed to $N_Q = 2^4 = 16$. According to Fig. 4.12(a), MP dictionary of very small size yields little quality improvement with refinement bits. The quality difference for the ROI and the whole image is not very much. On the other hand, a large MP dictionary results in very rapid quality improvement by receiving refinement bits especially for the ROI. Fig. 4.12(b) gives a comparison of rate-distortion behavior for different sizes of MP dictionary. As it can be seen from this figure, the rate-distortion performances for large MP dictionaries, e.g., B_D =8 to 12, are close. When the computational burden is strictly limited and the bit-budget is fixed, decrementing B_D by one bit means reducing the computational cost to half.

As mentioned before, the proposed method provides a progressive bit-stream which can be truncated at a desirable bit-budget or quality target. The receiver can decide to change its choice of ROI, i.e., the coordinates of the point of interest and ROI parameters, while the previously received information is left intact and only the required refinement information is transmitted. There is a similar feature for Internet protocol JPEG2000 standard (JPIP), regarding to the interactive and progressive ROI image coding [85]. However, there are some differences between JPIP and the proposed MP based ROI image coding in terms of the functionality of methods. The proposed method is capable of providing distinctive ROI(s) and background. The method is also able to provide a gradual change from ROI(s) to background by properly choosing the ROI parameters. In JPIP, however, this gradual quality change is not regarded. In JPIP, the ROI change is not as easy as that in the proposed method. In JPEG2000, if the receiver changes its choice of ROI, the server

needs to reorder the transferable bit-stream according to its cache model of the client. However, in MP-based ROI image coding scheme this reordering of bit-stream is not necessary.



(a)



(b)

Figure 4.12: Rate-distortion behavior of the MP-based ROI image coding scheme as a function of MP dictionary size (a) 3-D illustration of rate-distortion-complexity (b) Rate-distortion performance for different sizes of MP dictionary

Chapter 5

Joint Source/Channel Decoding of MP Coded Images

The need to transmit large amount of data over a bandlimited channel has led to the development of various source and channel coding schemes many of which function by attempting to remove all redundancies from the source data stream. This is justified in some sense by an important result of Shannon's distinguished paper [15] which shows that for rates below channel capacity, the source and channel coding operations can be separated without any loss of optimality. However, in this source-channel separation theorem, there is no constraint on the complexity or the delay of the involved coders. In practical systems, where there are limits on complexity or delay, this source-channel separation may not be the best approach. Besides, an unwanted side-effect of this approach is to make the information more vulnerable to the channel noise. Efforts at protecting against errors involve the reinsertion of redundancy and consequently an increase in bandwidth requirements.

During source coder design, for the sake of simplicity or due to imperfect knowledge about the source model, assumptions have to be made about the

source that are often inaccurate. This results in remaining redundancy at the output of the source encoder known as "residual" redundancy [68]. According to another part of Shannon's work, this residual redundancy can be exploited to provide protection against channel noise [15]. This technique forms the foundation of the present chapter and falls into the joint source-channel (JSC) coding category. Unequal error protection [69], optimized rate allocation [70], and index assignment [71] can also be categorized in JSC coding framework. JSC decoding has been viewed as an interesting new error concealment and enhanced channel decoding method and many researchers have investigated it in different directions. Robust transmission of images using residual redundancy of the source output is one of those directions [72–75]. It has been shown that DPCM source coded stream has some symbol by symbol residual redundancy [68]. For image transmission using DPCM coding over an error-prone channel, different methods that exploit the correlation of neighboring pixels are presented, such as sequence based minimum mean squared error (MMSE) estimation [72], maximum *a-posteriori* detection (MAP) [73], and 2-D Viterbi algorithm solution [74]. In [75] for discrete cosine transform (DCT) based and sub-band coding (SBC) based image transmission systems a variant of Viterbi algorithm is presented, which takes advantage of bit-by-bit dependency of the source coder output stream. JSC decoding for 1-D signals has been more popularly investigated in literature. Most research directions address fixed length coding over memoryless channels e.g., [76–78], while some others treat variable length codes (VLC) [79–81] and channels with memory [82].

In this chapter, a JSC decoding scheme for (MP) based coded image signals for robust transmission over a memoryless noisy channel is proposed. The suggested method exploits residual redundancy in the output of the source coder using a suboptimal sequence based minimum mean squared estimation

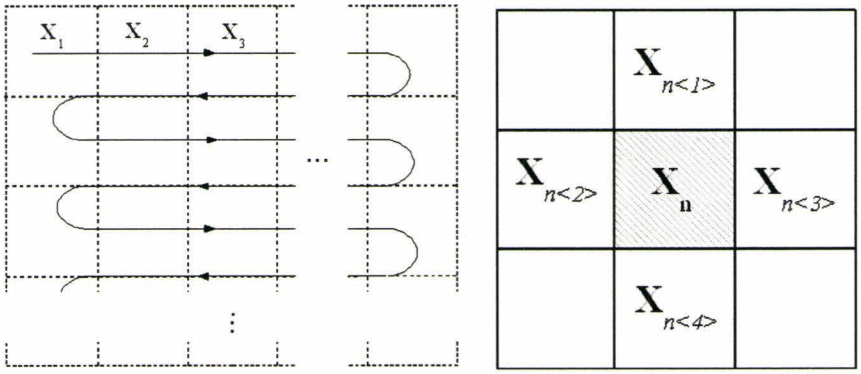


Figure 5.1: (a) Scan of image blocks in a raster manner (b) An image block and its closest neighbors

(MMSE) technique and equips the MP based image coding system with an error concealment solution without any additional bandwidth requirement. This chapter is organized as follows. Section 5.1 presents some preliminaries, such as notations and the MP based image transmission overview. In section 5.2, our suboptimal MMSE JSC decoding method for MP based image coded signals is formulated. Section 5.3 presents simulations and numerical results.

5.1 Preliminaries

5.1.1 Notations and System Overview

In this chapter, we employ the following notational rules. Random variables and their realizations are presented by capital letters, e.g. I, and lower case, e.g. i, respectively. For the sake of brevity we may write $P(I)$ instead of $P(I = i)$. Random vectors or matrixes are represented by bold-faced capital letters, e.g.: $\mathbf{I}$ or $\mathbf{X}$. Sequences of random variables are denoted by underlined letters, e.g.: $\underline{I}$. For the work in this thesis, images are divided into image blocks

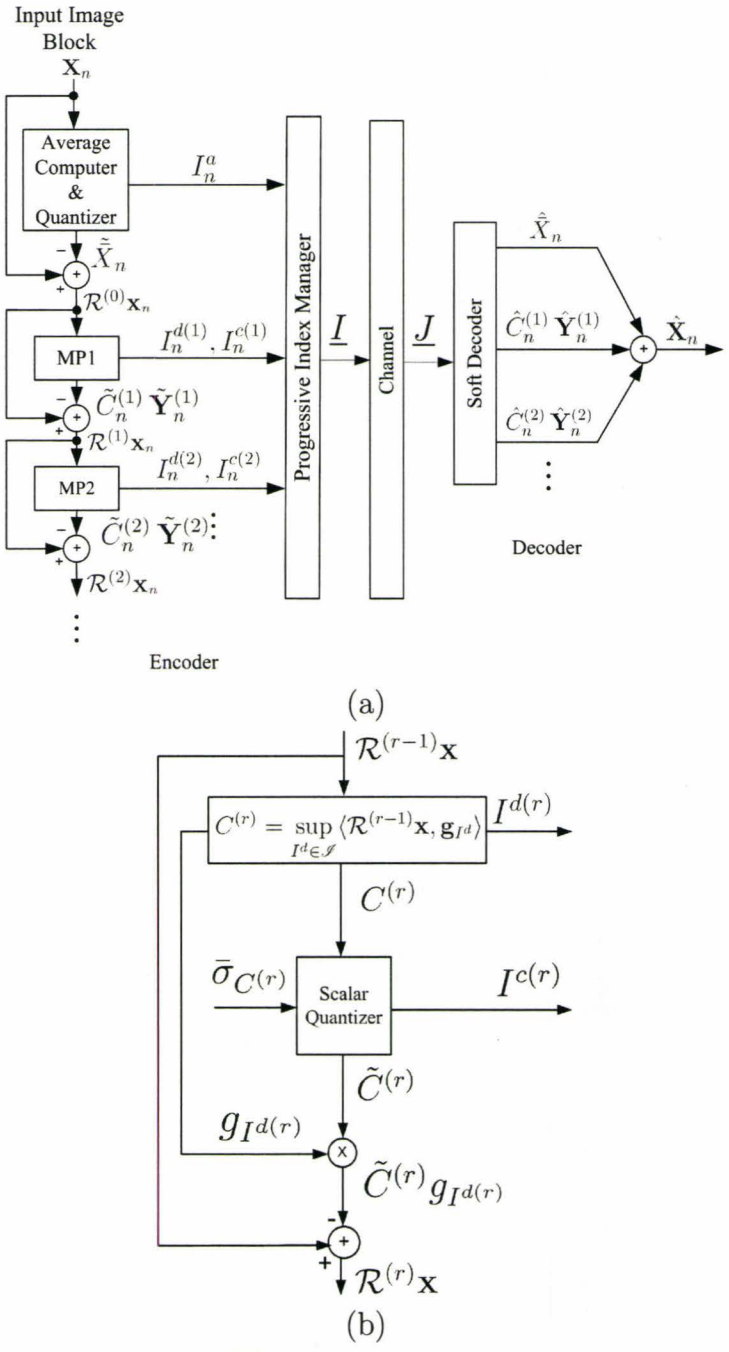


Figure 5.2: (a) Schematic block diagram of the MP based system under consideration (b) Detailed block diagram of r^{th} stage of MP analyzer

and coded in a raster manner [Fig. 5.1(a)]. The index associated with a one dimensional vector or the spatial location of a 2-D image block is labeled by a lower case subscript, like I_n. Figure 5.1(b) illustrates an image block $\mathbf{X}_n$ and its closest neighboring blocks. As shown in this figure, $\mathbf{X}_{n<r>} \quad r = 1, 2, 3, 4$ represent image blocks located on top, left, right, and at the bottom of $\mathbf{X}_n$ respectively. The lower case superscripts indicate the type of the indices, as in I_n^a [we will discuss different types of indices later in part (5.1.2)]. Superscript in parenthesis represent the MP stage number associated with an index. As an example, $I_n^{c(k)}$ represents the inner product coefficient index of k^{th} MP stage related to the vector $\mathbf{X}_n$.

The investigated transmission system is depicted in Fig. 5.2(a). The goal for this system is to deliver image signals using matching pursuit source coding scheme. An image is divided into image blocks of size $N_B \times N_B$ and a sequence of matrixes is formed $\underline{\mathbf{X}}_n = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$. Using an MP based source coder, image blocks are analyzed one by one and sequences of different indices $\underline{I}$ are generated. These indices are sent through a noisy channel and indices which are possibly corrupted, are delivered to the decoder ($\underline{J}$). The receiver reconstructs the image using the correlations of different received indices.

In this chapter, a memoryless binary symmetric channel (BSC) without inter-symbol interference is assumed. A binary phase-shift keying (BPSK) modulation over a channel with additive white Gaussian noise (AWGN) and hard decision is an example, which fits this model. This model is used in many previous works such as [77]. The channel model can be characterized by its index transition probability density function $P(J_n|I_n)$, where I_n is a transmitted index and J_n is the corresponding received index. This probability can be computed using the following equation:

$$P(J_n = j_n|I_n = i_n) = \epsilon^{H(j_n,i_n)}(1-\epsilon)^{l-H(j_n,i_n)} \tag{5.1}$$

where j_n, i_n are binary codewords, $H(j_n, i_n)$ is the Hamming distance between the transmitted and the received indices, ϵ is the bit-error probability (BEP), and l is the number of binary digits in I_n (or J_n). This probability function is assumed to be known by the receiver.

5.1.2 MP Based Image Coding and Transmission

The block diagram of the MP based communication system is shown in Fig. 5.2(a). The input signal is divided into a sequence $\underline{\mathbf{X}} = (\mathbf{X}_1, \mathbf{X}_1, \ldots, \mathbf{X}_n, \ldots)$ of $N_B \times N_B$ pixels image blocks. As the n^{th} input vector, $\mathbf{X}_n$ is applied to the first stage of the analysis part of the system. The mean of the input vector is initially removed by this stage in order to eliminate the necessity of a bigger dictionary for representing image blocks with different illumination and to improve the compression performance [1]. In order to reduce the effect of quantization error, the quantized mean value, $\tilde{\bar{X}}_n$, (and not $\bar{X}_n$) is employed to generate the mean removed entry vector to the first MP stage, i.e., $\mathcal{R}^{(0)}\mathbf{x}_n = \mathbf{X}_n - \tilde{\bar{X}}_n$. The index of this quantized average value (I_n^a) is sent through the channel. By applying the mean-removed vector $\mathcal{R}^{(0)}\mathbf{x}_n$ to the first MP analysis stage (MP1), the best match for this vector is selected using the following inner product operation:

$$C_n^{(1)} = \langle \mathcal{R}^{(0)}\mathbf{x}_n, \mathbf{g}_{I^{d(1)}} \rangle = \sup_{I^d \in \mathcal{J}^d} \langle \mathcal{R}^{(0)}\mathbf{x}_n, \mathbf{g}_{I^d} \rangle \tag{5.2}$$

This coefficient must then be quantized to $\tilde{C}_n^{(1)}$, since we can only transmit parameters with finite precision. The index of the selected dictionary element ($I_n^{d(1)}$) along with the index of the quantized inner product coefficient ($I_n^{c(1)}$) is sent to the decoder end. As shown in Fig. 5.2(a), in order to reduce the impact of quantization noise on the performance of MP analysis, the quantized inner product coefficient $\tilde{C}_n^{(1)}$ (instead of $C_n^{(1)}$) is employed for computation

of the residual signal, i.e., $\mathcal{R}^{(1)}\mathbf{x}_n = \mathcal{R}^{(0)}\mathbf{x}_n - \tilde{C}_n^{(1)}\mathbf{g}_{I_n^{d(1)}}$ (Note that $\mathbf{g}$ maps an index $I_n^{d(i)}$ to a 2D dictionary element $\mathbf{g}_{I_n^{d(i)}} = \tilde{\mathbf{Y}}_n^{(i)}$). $\mathcal{R}^{(1)}\mathbf{x}_n$ is now the entry signal to the second MP analysis stage (MP2). The resulting analysis parameters of MP2 ($I_n^{c(2)}$ and $I_n^{d(2)}$) are transmitted and the residual signal $\mathcal{R}^{(2)}\mathbf{x}_n = \mathcal{R}^{(1)}\mathbf{x}_n - \tilde{C}_n^{(2)}\mathbf{g}_{I_n^{d(2)}}$ is created as the input to the third MP stage (MP3). This iterative analysis procedure continues up to a predefined MP stage η. The detailed process at the r^{th} MP analysis stage is illustrated in Fig. 5.2(b). In this figure, the scalar quantizer adjusts its step size based on the standard deviation of the inner product coefficient of that stage. For more detail on the quantization of MP inner product coefficient see section 4.2.

The encoder part of Fig. 5.2(a) generates three different sequences of analysis indices that are to be sent to the receiver through a noisy channel:

- indices of average values of image blocks i.e., $\underline{I}_n^a = (I_1^a, I_2^a, \ldots, I_n^a)$
- dictionary index vectors, i.e., $\underline{\mathbf{I}}_n^d = (\mathbf{I}_1^d, \mathbf{I}_2^d, \ldots, \mathbf{I}_n^d)$ where $\mathbf{I}_i^d = (I_i^{d(1)}, I_i^{d(2)}, \ldots, I_i^{d(\eta)})$
- inner product coefficient index vectors, i.e., $\underline{\mathbf{I}}_n^c = (\mathbf{I}_1^c, \mathbf{I}_2^c, \ldots, \mathbf{I}_n^c)$ where $\mathbf{I}_i^c = (I_i^{c(1)}, I_i^{c(2)}, \ldots, I_i^{c(\eta)})$

The progressive index manager block in Fig. 5.2(a) arranges the analysis indices in order to form a progressive index bit-stream $\underline{I}$. On the other side of the noisy channel, the decoder receives the channel affected index sequences $\underline{J}$. If there was no channel error, the best estimation for the vector $\mathbf{X}_n$ would be

$$\tilde{\mathbf{X}}_n = \tilde{\bar{X}}_n + \sum_{i=1}^{\eta} \tilde{C}_n^{(i)}\, \mathbf{g}_{I_n^{d(i)}} = \tilde{\bar{X}}_n + \sum_{i=1}^{\eta} \tilde{C}_n^{(i)}\, \tilde{\mathbf{Y}}_n^{(i)} \tag{5.3}$$

In a more practical case and when the channel affects the transmitted indices, the joint source channel decoder estimates $\bar{X}_n$ as $\hat{\bar{X}}_n$, $C_n^{(i)}$ as $\hat{C}_n^{(i)}$,

and $\tilde{\mathbf{Y}}_n^{(i)}$ as $\hat{\mathbf{Y}}_n^{(i)}$ where $i = 1, 2, \ldots, \eta$. In this estimation, $\hat{\mathbf{Y}}_n^{(i)}$ may no longer belong to $\mathscr{D}$. The estimation for $\mathbf{X}_n$ can then be calculated using the following summations :

$$\hat{\mathbf{X}}_n = \hat{\hat{X}}_n + \sum_{i=1}^{\eta} \hat{C}_n^{(i)} \hat{\mathbf{Y}}_n^{(i)} \tag{5.4}$$

Details of how this JSC decoding works are explained in the next section.

5.2 Proposed Source-Channel Decoding

5.2.1 Optimal MMSE Based Decoder

We start this section with this premise that there are residual redundancy in the source output bit stream due to imperfect knowledge of source model and restrictions on the source coder complexity. The source-channel decoder presented in this section exploits the residual redundancy to improve the quality of a reconstructed image from a noise-corrupted bit-stream. We assume the decoder is allowed to perform with a delay of τ. The goal of MMSE estimator is to minimize $E[|(\mathbf{X}_n - \hat{\mathbf{X}}_n)^2|\underline{\mathbf{J}}_{n+\tau}]$, the estimation error, using all the received information up to time $n+\tau$. Note that $\underline{\mathbf{J}}_{n+\tau}$ comprises all received indices $[\underline{\mathbf{J}}^d_{n+\tau}, \underline{\mathbf{J}}^c_{n+\tau}, \underline{J}^a_{n+\tau}]$ up to time $n+\tau$. According to the fundamental theorem of estimation [83], the optimum MMSE solution to our problem is

$$\hat{\mathbf{X}}_n = E[\mathbf{X}_n|\underline{\mathbf{J}}_{n+\tau}]. \tag{5.5}$$

5.2.2 Sub-optimal MMSE Based Decoder

Optimal MMSE estimation of image blocks presented in (5.5) requires using all the past received indices and up to τ future image block indices. This

estimation approach is mathematically untractable and computationally unmanageable. In this part we consider some simplifying assumptions which yields practical sub-optimal soft decoder solution.

Here we assume the decoder has access to the same dictionary and code book as in the encoder, therefore $\hat{\mathbf{X}}_n \approx E[\tilde{\mathbf{X}}_n|\underline{\mathbf{J}}_{n+\tau}]$, where $\tilde{\mathbf{X}}_n$ is defined in (5.3), offers an approximate evaluation of (5.5) which limits the decoder complexity and the required memory to a practical level. Using (5.3) we have:

$$\hat{\mathbf{X}}_n = E[\left\{ \tilde{\bar{X}}_n + \sum_{i=1}^{\eta} \tilde{C}_n^{(i)} \times \tilde{\mathbf{Y}}_n^{(i)} \right\} |\underline{\mathbf{J}}_{n+\tau}] \tag{5.6}$$

Assuming the mutual independence of $\tilde{C}_n^{(i)}$, and $\tilde{\mathbf{Y}}_n^{(i)}$ we have:

$$\begin{aligned} \hat{\mathbf{X}}_n &= E[\tilde{\bar{X}}_n|\underline{\mathbf{J}}_{n+\tau}] + \sum_{i=1}^{\eta} E[\tilde{C}_n^{(i)} \times \tilde{\mathbf{Y}}_n^{(i)}|\underline{\mathbf{J}}_{n+\tau}] \\ &= E[\tilde{\bar{X}}_n|\underline{\mathbf{J}}_{n+\tau}] + \sum_{i=1}^{\eta} \left\{ E[\tilde{C}_n^{(i)}|\underline{\mathbf{J}}_{n+\tau}] \times E[\tilde{\mathbf{Y}}_n^{(i)}|\underline{\mathbf{J}}_{n+\tau}] \right\} \end{aligned} \tag{5.7}$$

Lets define the MMSE estimation associated to the analysis items $\bar{X}_n$, $C_n^{(i)}$, and $\mathbf{Y}_n^{(i)}$ as following:

$$\begin{aligned} \hat{\bar{X}}_n &= E[\tilde{\bar{X}}_n|\underline{\mathbf{J}}_{n+\tau}] = E[\tilde{\bar{X}}_n|\underline{\mathbf{J}}_{n+\tau}^d, \underline{\mathbf{J}}_{n+\tau}^c, \underline{J}_{n+\tau}^a] \\ \hat{C}_n^{(i)} &= E[\tilde{C}_n^{(i)}|\underline{\mathbf{J}}_{n+\tau}] = E[\tilde{C}_n^{(i)}|\underline{\mathbf{J}}_{n+\tau}^d, \underline{\mathbf{J}}_{n+\tau}^c, \underline{J}_{n+\tau}^a] \\ \hat{\mathbf{Y}}_n^{(i)} &= E[\tilde{\mathbf{Y}}_n^{(i)}|\underline{\mathbf{J}}_{n+\tau}] = E[\tilde{\mathbf{Y}}_n^{(i)}|\underline{\mathbf{J}}_{n+\tau}^d, \underline{\mathbf{J}}_{n+\tau}^c, \underline{J}_{n+\tau}^a] \end{aligned} \tag{5.8}$$

Using (5.7) we have:

$$\hat{\mathbf{X}}_n = \hat{\bar{X}}_n + \sum_{i=1}^{\eta} \hat{C}_n^{(i)} \times \hat{\mathbf{Y}}_n^{(i)} \tag{5.9}$$

The MMSE estimation problem has now been broken into three separate problems whose solution can be placed in (5.9) for ultimate image block reconstruction. Again, by using the mutual independence of $\tilde{\bar{X}}_n$, $\tilde{C}_n^{(i)}$ and $\tilde{\mathbf{Y}}_n^{(i)}$ and memoryless channel assumption, it is clear that each of these quantized

items is also mutually independent of the received indices of the other two quantized items. For example $\hat{\tilde{X}}_n$ is independent of $\underline{\mathbf{J}}^d_{n+\tau}$ and $\underline{\mathbf{J}}^c_{n+\tau}$. Therefore (5.8) yields:

$$\begin{aligned} \hat{\tilde{X}}_n &= E[\tilde{\tilde{X}}_n|\underline{J}^a_{n+\tau}] \\ \hat{C}^{(i)}_n &= E[\tilde{C}^{(i)}_n|\underline{\mathbf{J}}^c_{n+\tau}] \\ \hat{\mathbf{Y}}^{(i)}_n &= E[\tilde{\mathbf{Y}}^{(i)}_n|\underline{\mathbf{J}}^d_{n+\tau}] \end{aligned} \tag{5.10}$$

Using the law of total probability [84] we have

$$\begin{aligned} \hat{\tilde{X}}_n &= \sum_{I^a_n \in \mathscr{I}^a} E[\tilde{\tilde{X}}_n|I^a_n, \underline{J}^a_{n+\tau}]P(I^a_n|\underline{J}^a_{n+\tau}) \\ \hat{C}^{(i)}_n &= \sum_{I^{c(i)}_n \in \mathscr{I}^{c(i)}} E[\tilde{C}^{(i)}_n|I^{c(i)}_n, \underline{\mathbf{J}}^c_{n+\tau}]P(I^{c(i)}_n|\underline{\mathbf{J}}^c_{n+\tau}) \\ \hat{\mathbf{Y}}^{(i)}_n &= \sum_{I^{d(i)}_n \in \mathscr{I}^{d(i)}} E[\tilde{\mathbf{Y}}^{(i)}_n|I^{d(i)}_n, \underline{\mathbf{J}}^d_{n+\tau}]P(I^{d(i)}_n|\underline{\mathbf{J}}^d_{n+\tau}) \end{aligned} \tag{5.11}$$

As we mentioned previously, the decoder uses the same dictionary or codebook as the one in the encoder. Therefore by having the correct index of a quantized item we do not need any other information. This means for example $E[\tilde{\tilde{X}}_n|I^a_n, \underline{J}^a_{n+\tau}] \approx E[\tilde{\tilde{X}}_n|I^a_n]$. Consequently, (5.11) yields:

$$\hat{\tilde{X}}_n = \sum_{I^a_n \in \mathscr{I}^a} E[\tilde{\tilde{X}}_n|I^a_n]P(I^a_n|\underline{J}^a_{n+\tau}) \tag{5.12}$$

$$\hat{C}^{(i)}_n = \sum_{I^{c(i)}_n \in \mathscr{I}^{c(i)}} E[\tilde{C}^{(i)}_n|I^{c(i)}_n]P(I^{c(i)}_n|\underline{\mathbf{J}}^c_{n+\tau}) \tag{5.13}$$

$$\begin{aligned} \hat{\mathbf{Y}}^{(i)}_n &= \sum_{I^{d(i)}_n \in \mathscr{I}^{d(i)}} E[\tilde{\mathbf{Y}}^{(i)}_n|I^{d(i)}_n]P(I^{d(i)}_n|\underline{\mathbf{J}}^d_{n+\tau}) \\ &= \sum_{I^{d(i)}_n \in \mathscr{I}^{d(i)}} \mathbf{g}_{I^{d(i)}_n} P(I^{d(i)}_n|\underline{\mathbf{J}}^d_{n+\tau}) \end{aligned} \tag{5.14}$$

where the *a posteriori* probabilities act as weighting factors of a linear combination of quantized items. Here, the challenge is how to evaluate these probabilities.

5.2.3 *A posteriori* Probability Calculations

In this part the *a posteriori* probability computations, required in (5.12), (5.13), and (5.14), are explained. If there is no restriction on the number of exploited received indices for MMSE estimation of each analysis parameter, the problem will turn into a mathematically intractable and computationally very expensive task. Hence, in order to approximate $P(I_n^{(i)}|\underline{\mathbf{J}}_{n+\tau})$ in a practical fashion, the following indices from the larger collection of received indices ($\underline{\mathbf{J}}_{n+\tau}$) are selected:

- Instant information: $J_n^{(i)}$, the received (probably corrupted) version of $I_n^{(i)}$
- Stage-wise adjacent information: $J_n^{(i-1)}$ and $J_n^{(i+1)}$, the received analysis indices of the same image block but associated with previous and next MP analysis stage respectively (this information does not exist for I_n^a).
- Locally adjacent information: $J_{n<1>}^{(i)}, J_{n<2>}^{(i)}, J_{n<3>}^{(i)}, J_{n<4>}^{(i)}$, the received indices of the closest neighbors of $\mathbf{X}_n$ (see Fig. 5.1b)

The MP based image coding is assumed to follow a raster scan (fig. 5.1a) and to be progressive (layered). Therefore $J_n^{(i)}$ is always received before $J_n^{(i+1)}$ and after $J_n^{(i-1)}$ and the decoder must wait for next stage of MP analysis information to receive $J_n^{(i+1)}$. Using $J_n^{(i+1)}$ for MMSE estimation is more important for the first MP stage decoding where there is no prior stage-wise information to be exploited. It is also clear that $J_n^{(i)}$ is received before $J_{n<3>}^{(i)}$ and $J_{n<4>}^{(i)}$ and after $J_{n<1>}^{(i)}$ $J_{n<2>}^{(i)}$. Thus, $J_{n<3>}^{(i)}$ and $J_{n<4>}^{(i)}$ are to be received by enforcing delay to the decoder.

5.2.3.1 *A posteriori* probability associated with I_n^a

As explained before, the average value of each image block is scalar quantized and transmitted to the decoder prior to MP parameter transmission. Therefore, for the image block average value index, (I_n^a), useful information is the received indices of the average value of the block and its neighbors, and we have:

$$\begin{aligned} P(I_n^a|\underline{J}_{n+\tau}^a) &\approx P(I_n^a|J_n^a, J_{n<1>}^a, J_{n<2>}^a, J_{n<3>}^a, J_{n<4>}^a) \\ &= \frac{1}{P(J_n^a, J_{n<1>}^a, J_{n<2>}^a, J_{n<3>}^a, J_{n<4>}^a)} P(I_n^a)P(J_n^a|I_n^a) \\ &\quad \times P(J_{n<1>}^a|I_n^a, J_n^a)P(J_{n<2>}^a|I_n^a, J_n^a, J_{n<1>}^a) \\ &\quad \times P(J_{n<3>}^a|I_n^a, J_n^a, J_{n<1>}^a, J_{n<2>}^a) \\ &\quad \times P(J_{n<4>}^a|I_n^a, J_n^a, J_{n<1>}^a, J_{n<2>}^a, J_{n<3>}^a) \end{aligned} \tag{5.15}$$

where $P(I_n^a)$ is the index marginal PDF and $P(J_n^a|I_n^a)$ is the index transition probability formulated by (5.1). Using the fact that I_n^a is the original index and J_n^a is the corresponding received index over the noisy channel, probability $P(J_{n<1>}^a|I_n^a, J_n^a)$ can be reduced to $P(J_{n<1>}^a|I_n^a)$. We can also simplify $P(J_{n<2>}^a|I_n^a, J_n^a, J_{n<1>}^a)$ by the above reason and that $\mathbf{X}_{n<2>}$ is closer to $\mathbf{X}_n$ than to $\mathbf{X}_{n<1>}$ and consequently, knowing I_n^a does not leave any new information in $J_{n<1>}^a$ about $J_{n<2>}^a$. Hence, this conditional probability can be approximated by $P(J_{n<2>}^a|I_n^a)$. By the same token other conditional probabilities in (5.15) are simplified and we have

$$P(I_n^a|J_n^a, J_{n<1>}^a, \ldots, J_{n<4>}^a) \approx K_1 P(I_n^a)P(J_n^a|I_n^a) \prod_{r=1}^{4} P(J_{n<r>}^a|I_n^a) \tag{5.16}$$

where $K_1 = 1/P(J_n^a, J_{n<1>}^a, \ldots, J_{n<4>}^a)$ is a normalizing factor that makes the *a posteriori* probability hold the second axiom of probability [84]. Using the

law of total probability we have:

$$P(I_n^a|J_n^a, J_{n<1>}^a, \ldots, J_{n<4>}^a) \approx K_1 P(I_n^a)P(J_n^a|I_n^a) \prod_{r=1}^{4} \sum_{I_{n<r>}^a \in \mathcal{J}^a} P(I_{n<r>}^a|I_n^a)P(J_{n<r>}^a|I_{n<r>}^a) \tag{5.17}$$

where $P(I_{n<r>}^a|I_n^a)$ is a conditional probability that can be statistically calculated, using a set of test images, and stored in the decoder prior to the image communication. $P(J_{n<r>}^a|I_{n<r>}^a)$ is the index transition probability that can be calculated using (5.1).

5.2.3.2 *A posteriori* probability associated with $I_n^{c(i)}$

In order to evaluate the *a posteriori* probability used in (5.13), as explained before, we employ both stage-wise adjacent as well as locally adjacent information. Therefore we approximate the *a posteriori* probability of MP inner product coefficient index by the following expression.

$$P(I_n^{c(i)}|\underline{J}_{n+\tau}^c) \approx P(I_n^{c(i)}|J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, \ldots, J_{n<4>}^{c(i)}) \tag{5.18}$$

This conditional probability can be written in the form of the next joint probabilities:

$$P(I_n^{c(i)}|J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, \ldots, J_{n<4>}^{c(i)}) = \frac{P(I_n^{c(i)}, J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, \ldots, J_{n<4>}^{c(i)})}{P(J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, \ldots, J_{n<4>}^{c(i)})} \tag{5.19}$$

The probability chain rule :

$$P(A_1, A_2, A_3, \ldots) = P(A_1)P(A_2|A_1)P(A_3|A_1, A_2)\ldots \tag{5.20}$$

as well as (5.19) yield:

$$
\begin{aligned}
&P(I_n^{c(i)}|J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, \ldots, J_{n<4>}^{c(i)}) = \\
&\qquad \frac{1}{P(J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, J_{n<2>}^{c(i)}, J_{n<3>}^{c(i)}, J_{n<4>}^{c(i)})} \\
&\times P(I_n^{c(i)}) P(J_n^{c(i)}|I_n^{c(i)}) P(J_n^{c(i-1)}|I_n^{c(i)}, J_n^{c(i)}) P(J_n^{c(i+1)}|I_n^{c(i)}, J_n^{c(i)}, J_n^{c(i-1)}) \\
&\times P(J_{n<1>}^{c(i)}|I_n^{c(i)}, J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}) \\
&\times P(J_{n<2>}^{c(i)}|I_n^{c(i)}, J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}) \\
&\times P(J_{n<3>}^{c(i)}|I_n^{c(i)}, J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, J_{n<2>}^{c(i)}) \\
&\times P(J_{n<4>}^{c(i)}|I_n^{c(i)}, J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, J_{n<2>}^{c(i)}, J_{n<3>}^{c(i)})
\end{aligned}
\tag{5.21}
$$

Where each conditional probability must be calculated for the set of received indices and the transmitted index $I_n^{c(i)}$. The marginal probability $P(I_n^{c(i)})$, and index transition probability $P(J_n^{c(i)}|I_n^{c(i)})$ do not need to be touched. Starting with $P(J_n^{c(i-1)}|I_n^{c(i)}, J_n^{c(i)})$, since $J_n^{c(i)}$ is the received version of the original index $I_n^{c(i)}$ and the channel is memoryless, knowing the latter index, there is no information in $J_n^{c(i)}$ left about $J_n^{c(i-1)}$. Consequently, this conditional probability can be reduced to $P(J_n^{c(i-1)}|I_n^{c(i)})$. With the same reasoning and the fact that $I_n^{c(i)}$ has more information about $J_n^{c(i+1)}$ than $J_n^{c(i-1)}$ (since $J_n^{c(i)}$ is in an MP analysis layer between $J_n^{c(i+1)}$ and $J_n^{c(i-1)}$), we can shorten $P(J_n^{c(i+1)}|I_n^{c(i)}, J_n^{c(i)}, J_n^{c(i-1)})$ to $P(J_n^{c(i+1)}|I_n^{c(i)})$. Using the above logic and the fact that $J_{n<1>}^{c(i)}$ inherits more residual redundancy from $I_n^{c(i)}$ than the rest of given knowledge in $P(J_{n<1>}^{c(i)}|I_n^{c(i)}, J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)})$, this conditional probability can be reduced to $P(J_{n<1>}^{c(i)}|I_n^{c(i)})$. Due to spatial proximity of $\mathbf{X}_{n<2>}$ and $\mathbf{X}_n$ compared to $\mathbf{X}_{n<2>}$ and $\mathbf{X}_{n<1>}$ [see Fig. 5.1(b)], and all the above-mentioned reasons, $P(J_{n<2>}^{c(i)}|I_n^{c(i)}, J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)})$ can be reduced to $P(J_{n<2>}^{c(i)}|I_n^{c(i)})$. Two remaining conditional probabilities can also be simplified

with the same logic. Finally, (5.21) can be shortened to the following:

$$P(I_n^{c(i)}|J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, \ldots, J_{n<4>}^{c(i)}) \approx K_2 P(I_n^{c(i)}) \times$$
$$P(J_n^{c(i)}|I_n^{c(i)}) P(J_n^{c(i-1)}|I_n^{c(i)}) P(J_n^{c(i+1)}|I_n^{c(i)}) \prod_{r=1}^{4} P(J_{n<r>}^{c(i)}|I_n^{c(i)}) \quad (5.22)$$

where $K_2 = 1/P(J_n^{c(i)}, J_n^{c(i-1)}, J_n^{c(i+1)}, J_{n<1>}^{c(i)}, \ldots, J_{n<4>}^{c(i)})$ is a normalizing factor making the *a posteriori* probability on the left hand side of (5.22) conform the second axiom of probability [84]. $P(I_n^{c(i)})$ in (5.22) is the marginal PDF of $I_n^{c(i)}$ and can be statistically approximated using a set of test images. $P(J_n^{c(i)}|I_n^{c(i)})$ in this formula is the index transition probability which can be calculated using (5.1). Using the law of total probability [84] yields:

$$P(J_n^{c(i-1)}|I_n^{c(i)}) = \sum_{I_n^{c(i-1)} \in \mathscr{I}^{c(i-1)}} P(I_n^{c(i-1)}|I_n^{c(i)}) P(J_n^{c(i-1)}|I_n^{c(i-1)}) \quad (5.23)$$

$$P(J_n^{c(i+1)}|I_n^{c(i)}) = \sum_{I_n^{c(i+1)} \in \mathscr{I}^{c(i+1)}} P(I_n^{c(i+1)}|I_n^{c(i)}) P(J_n^{c(i+1)}|I_n^{c(i+1)}) \quad (5.24)$$

$$P(J_{n<r>}^{c(i)}|I_n^{c(i)}) \sum_{I_{n<r>}^{c(i)} \in \mathscr{I}^{c(i)}} P(I_{n<r>}^{c(i)}|I_n^{c(i)}) P(J_{n<r>}^{c(i)}|I_{n<r>}^{c(i)}) \quad (5.25)$$

$P(I_n^{c(i-1)}|I_n^{c(i)})$ in (5.23), $P(I_n^{c(i+1)}|I_n^{c(i)})$ in (5.24), and $P(I_{n<r>}^{c(i)}|I_n^{c(i)})$ in (5.25) are prior knowledge that have to be statistically collected and stored in the decoder.

5.2.3.3 *A posteriori* probability associated with $I_n^{d(i)}$

Although MMSE estimation of MP dictionary elements (5.14) deals with vectors, while (5.13) deals with scalars, formulas (5.12) to (5.14) have the same structure. The *a posteriori* probability associated with $I_n^{d(i)}$ like the one for $I_n^{c(i)}$ requires both stage-wise adjacent and locally adjacent knowledge.

Therefore to approximate the *a posteriori* probability for $I_n^{d(i)}$, we use similar approximation formulation to the ones for $I_n^{c(i)}$ and we have

$$P(I_n^{d(i)}|\underline{J}_{n+\tau}^{d}) \approx$$
$$P(I_n^{d(i)}|J_n^{d(i)}, J_n^{d(i-1)}, J_n^{d(i+1)}, J_{n<1>}^{d(i)}, \ldots, J_{n<4>}^{d(i)}) \approx K_3 P(I_n^{d(i)}) \times$$
$$P(J_n^{d(i)}|I_n^{d(i)}) P(J_n^{d(i-1)}|I_n^{d(i)}) P(J_n^{d(i+1)}|I_n^{d(i)}) \prod_{r=1}^{4} P(J_{n<r>}^{d(i)}|I_n^{d(i)}) \quad (5.26)$$

$$P(J_n^{d(i-1)}|I_n^{d(i)}) = \sum_{I_n^{d(i-1)} \in \mathcal{I}^{d(i-1)}} P(I_n^{d(i-1)}|I_n^{d(i)}) P(J_n^{d(i-1)}|I_n^{d(i-1)}) \quad (5.27)$$

$$P(J_n^{d(i+1)}|I_n^{d(i)}) = \sum_{I_n^{d(i+1)} \in \mathcal{I}^{d(i+1)}} P(I_n^{d(i+1)}|I_n^{d(i)}) P(J_n^{d(i+1)}|I_n^{d(i+1)}) \quad (5.28)$$

$$P(J_{n<r>}^{d(i)}|I_n^{d(i)}) \sum_{I_{n<r>}^{d(i)} \in \mathcal{I}^{d(i)}} P(I_{n<r>}^{d(i)}|I_n^{d(i)}) P(J_{n<r>}^{d(i)}|I_{n<r>}^{d(i)}) \quad (5.29)$$

where $K_3 = 1/P(J_n^{d(i)}, J_n^{d(i-1)}, J_n^{d(i+1)}, J_{n<1>}^{d(i)}, \ldots, J_{n<4>}^{d(i)})$ is a normalizing factor. In (5.26) to (5.29), $P(I_n^{d(i)})$, $P(I_n^{d(i-1)}|I_n^{d(i)})$, $P(I_n^{d(i+1)}|I_n^{d(i)})$, and $P(I_{n<r>}^{d(i)}|I_n^{d(i)})$ are approximated using a large set of training images and stored in the decoder. Also $P(J_n^{d(i)}|I_n^{d(i)})$, $P(J_n^{d(i-1)}|I_n^{d(i-1)})$, $P(J_n^{d(i+1)}|I_n^{d(i+1)})$, and $P(J_{n<r>}^{d(i)}|I_{n<r>}^{d(i)})$ are index transition probabilities and can be calculated using (5.1).

5.3 Experimental Results

The simulation results for the proposed joint source/channel decoding of the MP based coded image bit-stream are presented in this section. First, the MP based image coding set-up including the choice of dictionary and index bit assignment are discussed. Then different variants of the JSC decoding formulations are compared.

5.3.1 MP Image Coding Set-up

5.3.1.1 MP dictionary

MP image coding algorithm holds no restriction for the choice of dictionary. Here we choose the 2-D separable Gabor dictionary of dimension 8×8 similar to what is used in the previous chapter. This choice of dictionary is not claimed to be optimal, it is just to exhibit the functionality of the proposed JSC decoding scheme.

5.3.1.2 Index bit assignment

The size of the MP dictionary has a great impact on the computational complexity of the matching pursuit analysis. While an appropriately larger dictionary may better express signal structures, it requires more bits to represent each dictionary element when it is compared to smaller dictionaries. Therefore, we have to consider complexity, quality, and rate trade-offs when it comes to assign dictionary index bits and to choose the MP dictionary size. To reduce the size of MP dictionary, the method introduced in the previous chapter is employed[†] and subsets of the very large original dictionary of cardinality N_D are selected.

The energy of the MP inner product coefficients decreases exponentially with the MP stage number [42]. For the experimental results, a uniform adaptive mid-tread scalar quantizer, the granular region [68] of which is from $-3\sigma_{C^n}$

[†]The 6400-element Gabor dictionary, whose specifications are characterized in section 4.2.1, is employed for MP analysis of a set of test images. In each MP analysis stage, different subsets of the original MP dictionary are used more often. This is because of the residual patterns of image blocks in each MP stage, that are not necessarily similar to the structures in other stages, and in order to extract these patterns in different MP stages, different subsets of the original MP dictionary are used more frequently. To reduce the MP dictionary size to N_D, the usage frequency of each element of the original dictionary in each MP stage is recorded and the top N_D most popular elements of the dictionary are selected as the shrunk dictionary of size N_D in that stage.

to $3\sigma_{C^n}$ is employed, where σ_{C^n} is the standard deviation of inner product coefficients at n^{th} MP stage. in section 4.2 it was shown that the optimum number of quantization steps for the MP inner product coefficients is $N_Q = 16$. Therefore, we assign $B_Q = \log_2(N_Q) = 4$ bits to represent the coefficients. The mean value of each image block is also quantized by a 16-step scalar quantizer (or 4 bits).

5.3.2 Joint Source/Channel Decoding Results

For the experimental results regarding our proposed JSC decoding scheme no entropy coding is employed, however using entropy coding does not have any effect on functionality of the proposed method. For simulation results we compare different image reconstruction methods listed as follows:

- Case 1: without any soft decoding and by employing the indices as they are received,
- Case 2: using marginal probability of the indices, $P(I_n)$, to approximate *a posteriori* probability $P(I_n|\underline{\mathbf{J}}_{n+\tau})$ for all different types of indices,
- Case 3: using no residual information and by approximating the *a posteriori* probability with $KP(I_n)P(J_n|I_n)$ for the soft decoding (K is a normalizing factor for probabilities where it applies),
- Case 4: using soft decoding with no delay, i.e. by approximating the *a posteriori* probability with

$$P(I_n^a|J_n^a, J_{n<1>}^a, J_{n<2>}^a) \approx KP(I_n^a)P(J_n^a|I_n^a)P(J_{n<1>}^a|I_n^a)P(J_{n<2>}^a|I_n^a)$$

for image block average value estimation and

$$P(I_n^{(i)}|J_n^{(i)}, J_{n<1>}^{(i)}, J_{n<2>}^{(i)}, J_n^{(i-1)}) \approx K P(I_n^{(i)}) P(J_n^{(i)}|I_n^{(i)}) P(J_{n<1>}^{(i)}|I_n^{(i)}) \times P(J_{n<2>}^{(i)}|I_n^{(i)}) P(J_n^{(i-1)}|I_n^{(i)})$$

for dictionary indices or inner product coefficients estimations,

- Case 5: using *a posteriori* probability approximation introduced in (5.17), (5.22), and (5.26) for the first two MP stages and the initial mean reassembling stage. The further MP stages are reconstructed as in case 3.

- Case 6: using *a posteriori* probability presented in (5.17), (5.22), and (5.26).

The required probabilities in the *a posteriori* calculation of the proposed JSC decoding formulations (except index transition conditional PDF $P(J_n|I_n)$ which is calculated using (5.1)) are estimated statistically by MP analysis of a set of 100 different images. The image collection comprises images of different contexts such as scenery, portraits, and objects.

5.3.2.1 Visual comparisons of different reconstruction cases

Fig. 5.3 demonstrates the error concealment performance of the proposed JSC MMSE based soft decoding method. The test images used in this section are not part of the training image set. Fig. 5.3(a) shows the original monochrome 8 bpp (bits per pixel) "Zelda" test image with size 256×256. Fig. 5.3(b) exhibits the reconstructed image using $\eta = 5$ MP stages by MP dictionary of size $N_D = 256$ ($B_D = 8$ bits) without the presence of the channel noise. The image block average values and the inner product coefficients

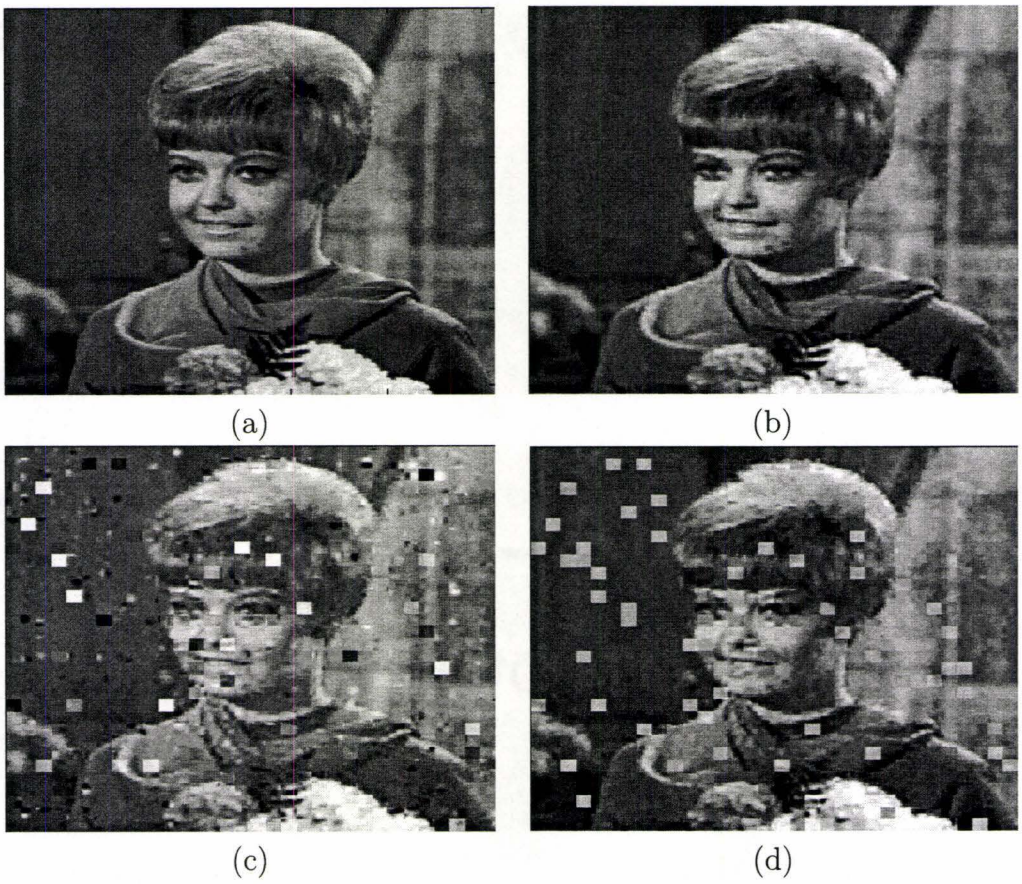
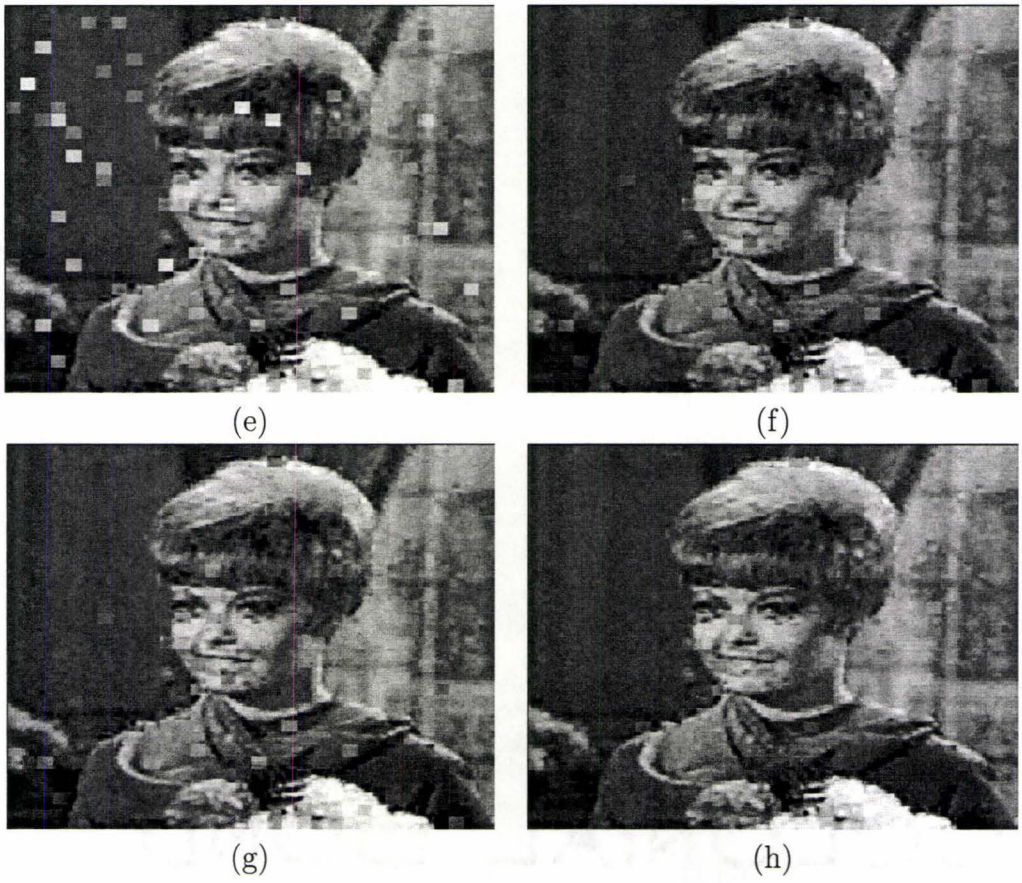


Figure 5.3: Visual comparison of different reconstruction cases at the presence of channel noise with bit error probability of $p_e = 0.02$. Image blocks are of size 8×8 and MP dictionary is the Gabor dictionary described in the previous sections with size $N_D = 256$ (dictionary indices are represented by 8 bits). Here the MP analysis is restricted to $\eta = 5$. Mean values as well as the inner product coefficients are represented by $B_Q = 4$ bits. (a) Original monochrome "Zelda" image of size 256×256 represented by 8 bpp. (b) Reconstructed image after 5 MP stages before being corrupted by the channel noise. (c) to (h) Reconstructed images using methods suggested in case 1 to case 6 respectively.



(e) (f)

(g) (h)

Figure 5.3 continue

are represented by $B_Q = 4$ bits. PSNR of the reconstructed image in this figure is 31.38 dB at 1 bpp. Fig. 5.3(c) shows the reconstructed image for case 1, i.e., when the MP analysis indices are exposed to channel noise and no error concealment scheme is employed. The channel bit error rate in this figure is $p_e = 0.02$ and the resulting PSNR in this case is 19.95 dB. In Fig. 5.3(d), the corrupted signal in 5.3(c) is a bit enhanced using case 2 of reconstruction, i.e., using the marginal probability of indices to approximate the required *a posteriori* probability for error concealment. PSNR in this case is a little bit better than the previous case and it is 21.9 dB. Fig. 5.3(e) displays the ameliorated version of Fig. 5.3(c) using the error concealment soft decoding without any use of residual information and by employing index marginal probability and index transition conditional probability (case 3). In this case, since the required *a posteriori* for soft decoding is approximated using more prior knowledge (compared to case 2), the resulting PSNR is 22.9 dB, a little higher than case 2. The reconstruction result for case 4 is shown in fig. 5.3(f). In this case, the soft decoding conforms the progressive nature of the layered MP analysis. In other words, the indices are estimated using the previously received information and with no delay. As it can be seen by this figure, the image quality has drastically enhanced compared to the previous cases. PSNR in this case is 26.37 dB. MP analysis removes most of the signal patterns in the early MP analysis stages and the residual vectors become more noise-like for higher MP stages. Therefore, the stage-wise and locally adjacent information seems to be not very useful for error concealment. In case 5, this fact has been taken into account and only first two MP stages are decoded using the full proposed a posteriori approximation formulas (5.17), (5.22), and (5.26). Fig. 5.3(g) displays the reconstructed image using this error concealment case. As it can be seen, image quality in this case is very close to case 4. PSNR in this

case is 26.35 dB. Fig. 5.3(h) shows the resulting reconstructed image using case 6 for error concealment method. In this case the instant information, the stage-wise adjacent information, and locally adjacent information for all 5 MP stages are exploited in order to ameliorate the corrupted received image using (5.17), (5.22), and (5.26). PSNR in this case is 27.1 dB, a little bit higher than the last two cases. Therefore, the proposed JSC decoding method has improved the image quality by 7 dB.

Fig. 5.4 exhibits another example of error concealment performance of the proposed JSC decoding scheme. Fig. 5.4(a) shows the original monochrome "House" image represented by 8 bpp. The test image is represented using the analysis parameters of 5 MP stages ($\eta = 5$) in Fig. 5.4(b) where PSNR=31.13 dB. In this figure, the MP setup is the same as Fig. 5.3. In Fig. 5.4(c), the analysis parameters are affected by a noisy channel with bit error rate of $p_e = 0.02$, and the image is reconstructed without using any error concealment method (case 1), which results in PSNR=21.31 dB. The proposed JSC decoding method (case 6) yields an improved version of the image, which is displayed in Fig. 5.4(d) with PSNR=28.29 dB. This figure also shows a PSNR improvement of 7 dB.

5.3.2.2 The effect of bit error rate, MP dictionary size, and number of stage

Fig. 5.5 illustrates the channel noise effect on the quality of the reconstructed image in different cases described previously. The effect is examined when "Zelda" image is analyzed by a 5 MP stages with Gabor dictionary of size $N_D = 256$. The experiment was done 10 times for each BER and the result reflects the mean value of PSNR for every bit error rate. As depicted in this figure, when bit error rate is very small, different reconstruction

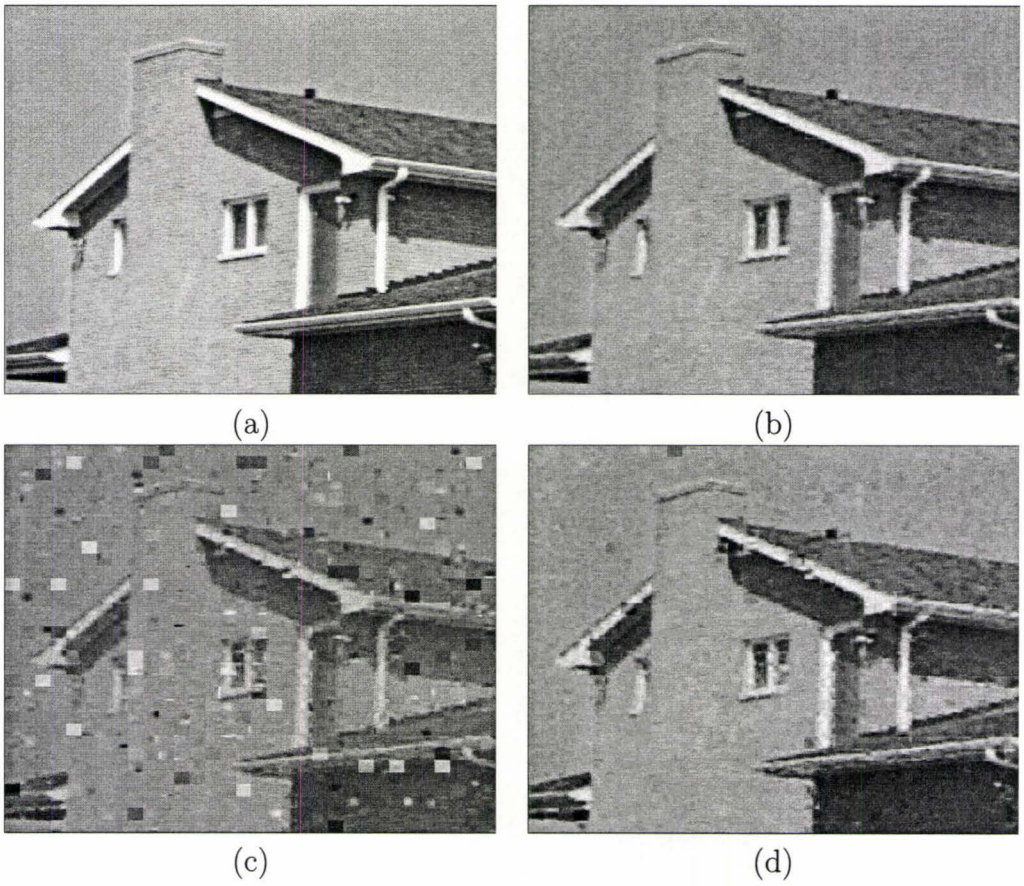


Figure 5.4: Error concealment performance of the proposed method for a different image. (a) Original monochrome "House" image represented by 8 bpp. (b) Image representation by $\eta = 5$ MP stages of dictionary size $N_D = 256$. Image blocks are of size 8×8 and each pixel is represented by 1 bpp and PSNR=31.13 dB. (c) The image reconstructed after being affected by channel noise of $p_e = 0.02$ using reconstruction case 1, i.e. using indices without any error concealment (PSNR=21.31 dB). (d) Reconstructed image using the proposed JSC decoding method with PSNR=28.29 dB.

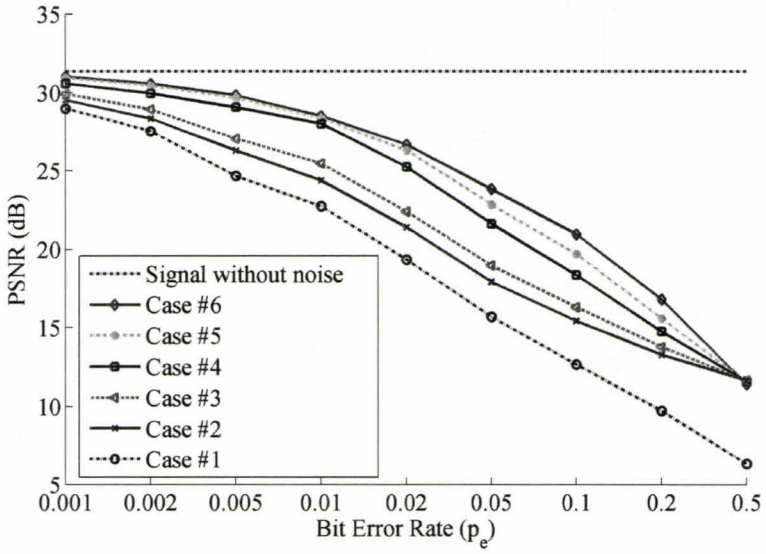
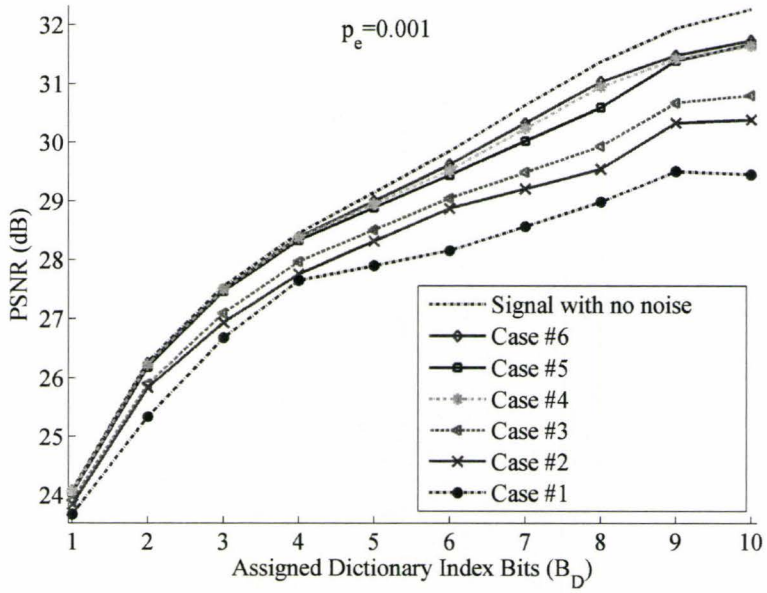


Figure 5.5: Image PSNR versus bit error rate when "Zelda" image is MP coded using 5 MP stages by a Gabor dictionary of size $N_D = 256$ for different reconstruction cases.

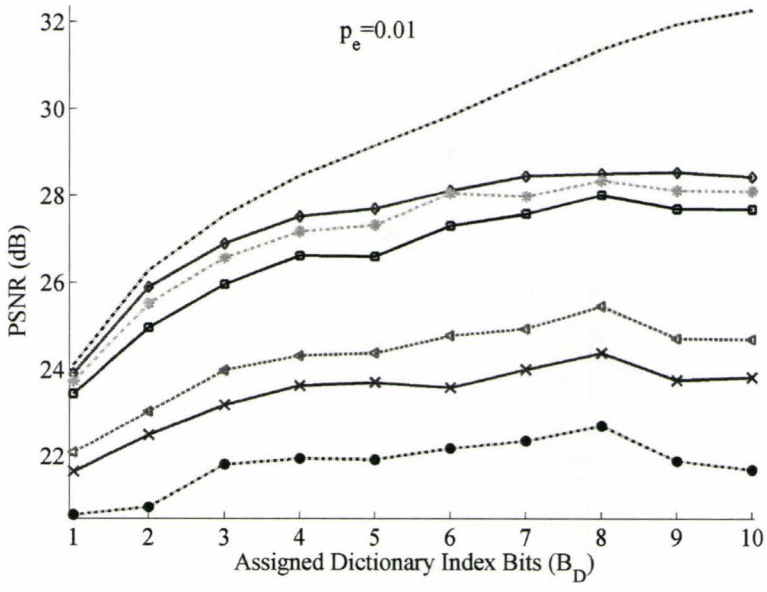
methods perform similarly and image quality approaches to the no-noise condition. This is because the proposed scheme is an error concealment method and works on indices which are corrupted by noise and leaves the correctly received indices unchanged. Also the figure shows that the full exploitation of instant, stage-wise adjacent, and locally adjacent information provides better reconstruction at the cost of higher computational complexity and delay for the soft decoder. Case 5 provides a little lower quality compared to case 6 but requires less computation. In this case there is delay for reconstruction of the first two MP stages. Case 4 yields no delay reconstruction with a lower image quality than case 5 and case 6. All the reconstruction cases perform better than case 1, with no soft decoding. The proposed method in this work is based on residual redundancy, i.e., the remaining correlation between the adjacent image blocks or adjacent MP stages. Therefore, the suggested error concealment method shows lower performance quality when the adjacent

indices are more corrupted. When all the adjacent indices are distorted by a very noisy channel, the performance of the proposed method approaches to the case where only instantaneous information is used (case 2, and 3). In perfect channel condition, all the reconstruction cases perform the same [when there is no channel noise ($p_e = 0$)].

Fig. 5.6 exhibits the effect of MP dictionary size on the performance of the proposed JSC decoding and error concealment method. Here, "Zelda" image is coded using 5 MP stages. Experiments for each bit error rate are repeated 10 times. Fig. 5.6(a) shows image quality for different reconstruction cases when $p_e = 0.001$. For small bit error rate, e.g. $p_e = 0.001$, the knowledge-base reconstruction methods (case 2 to 6) show more PSNR advantage when the MP dictionary is larger. This is because the indices of larger dictionaries require more bits for representation and consequently they are more likely to be corrupted than those of smaller dictionaries. Error concealment of a higher number of indices for the suggested soft decoders yields more PSNR advantage over case 1. When the bit error rate is higher, but not too high, the advantage of the soft decoding, especially when decoder exploits residual redundancy, over case 1 is more significant. As Fig. 5.6(b) depicts, case 6 outperforms all other cases for all dictionary sizes. This figure also shows that the slope of PSNR curve for each reconstruction case diminishes. This is due to higher probability of dictionary index error for larger dictionaries, although larger dictionaries yield better PSNR (when there is no channel noise). Fig. 5.7 displays a 3-D PSNR curve of case 1 and case 6 for different bit error rates and MP dictionary sizes. The surface shows that the highest PSNR advantage of case 6 (the full exploitation of instantaneous and adjacent information) over no error concealment is around $0.005 < p_e < 0.05$. When bit error rate is extremely high, the performance of case 6 reduced to case 2, since the received



(a)



(b)

Figure 5.6: The effect of dictionary size on the performance of the proposed JSC decoding scheme when 5-stage MP coded "Zelda" image with different MP dictionary size is corrupted with channel noise. The experiment has been repeated 10 times for each BER. (a) and (b) Results when p_e is equal to 0.001 and 0.01 respectively.

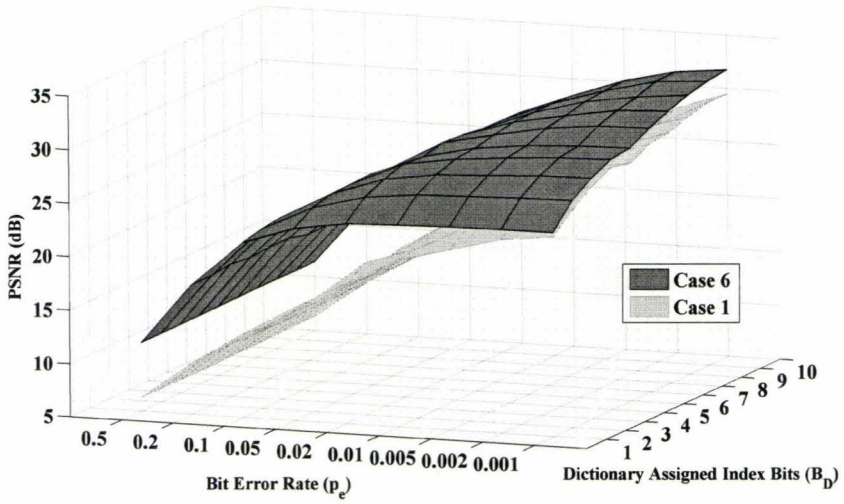


Figure 5.7: 3-D illustration of PSNR, MP dictionary size, and bit error rate relations in reconstruction case 1 and case 6.

indices do not carry any information (see tail of Fig. 5.5 for case 6 where $p_e = 0.5$).

Fig. 5.8 demonstrates the effect of MP employed stage numbers on performance of different image reconstruction cases. In order to make this figure, "Zelda" image was MP coded with different stage numbers by MP dictionary of size $N_D = 256$. The resulting bit-streams were corrupted by channel noise with bit error rate of $p_e = 0.02$ and different cases of image reconstructions were employed . We reproduced the results 10 times and the figure reflects the average values of PSNR for every number of employed MP stages. Stage 0 in this figure represents image blocks reconstructed merely using average values of the block. As this figure shows, the curve of image reconstruction cases have very slow slope for stages beyond stage 2. This is due to low energy of the residual vectors for those MP stages. By comparing Case 5 and case 6 in this figure, we can infer that, for higher MP stages (e.g., higher than 2) since

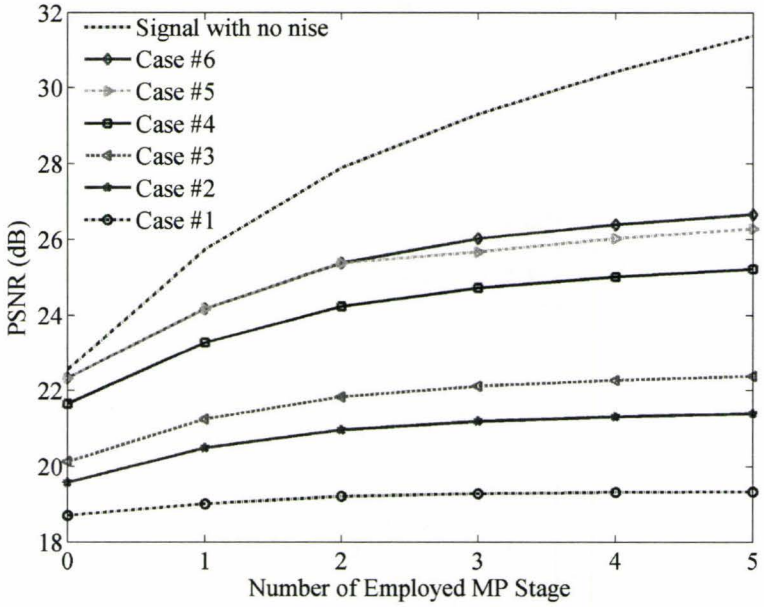


Figure 5.8: Image PSNR versus employed MP stages when "Zelda" image is MP coded using a Gabor dictionary of size $N_D = 256$ and bit error rate is $p_e = 0.02$.

the residual vectors are noise-like, there is not a significant amount of adjacent information while decoding the received indices. This manifests itself in little PSNR advantage of case 6 over case 5.

Chapter 6

Conclusions and Future work

6.1 Conclusions

Through the work presented in this thesis, the problem of image coding and transmission when the source coder is based on residual vector quantization or matching pursuit is addressed. The important property of quality scalability in image coding context can be manifested by progressive bit-stream and region of interest coding. Unequal error protection and joint source channel decoding are two other themes of this thesis for RVQ or MP based image communication.

In chapter 3, we introduced new ROI image coding approaches based on VQ. Although unstructured VQ using multiple codebook size provides a simple method of having variable spatial image quality, this approach suffers from low maximum local quality. ROI image coding based on residual vector quantization can address this drawback. Furthermore by its successive approximation, the RVQ approach provides an embedded progressive streaming scheme and bit-rate scalability. Moreover, this method allows the receiver client to change the ROI and to have multiple ROIs. The proposed algorithm is also able to

imitate the gradual quality degradation of the foveal model in the human visual system (see appendix A). Using the mean removed scenario for the RVQ foveation scheme not only has all the aforementioned advantages, but also improves the rate-distortion performance. Our jointly sub-optimal RVQ scheme improves the rate-distortion performance of the quantization scheme at the expense of more memory and computational cost. Therefore, a combination of mean-removed and jointly optimized RVQ can provide a rate-distortion-complexity trade off for our ROI image coding scheme. By considering the level of importance of different RVQ stages, we can protect the output bit-stream of the source encoder with an unequal error protection scheme.

In chapter 4, a new ROI image coding approach based on matching pursuits is introduced. The method interactively allows the transmitter to send MP analysis data with the emphasis on ROI(s). The ROI parameters include the radius of initial virtual circle (R_1) and the updating multiplier (α). Depending on the bit-budget, the degree of importance of the ROI(s), and the computational capability of the transmitter, a proper ROI parameter set and MP dictionary size can be selected. Our investigations in this chapter also show how reducing the size of MP dictionary can affect rate-distortion performance of the proposed image coding method while decreases the MP analysis computation burden. The effect of scalar quantization of MP inner product coefficients on transmission bit-rate and the reconstructed image quality were also investigated in this part of the thesis. The progressive nature of MP based coded bit-stream, like in RVQ based case, is potentially suitable for unequal error protection.

The proposed VQ and MP based methods are unbalanced and for the case where receiver end-user is not well equipped, these methods concentrate the complexity inside the transmitter and the receiver is very simple. Since

the methods are interactive, they are suitable for fast browsing from image archives. They progressively improve the quality of the selected image by emphasis on refining the visual information close to ROI.

Internet protocol standard for JPEG-2000 (JPIP) has very similar capabilities for interactive and progressive ROI image coding [85, 86]. Since there is no unique criterion to evaluate the performance of an ROI image coder, it is not easy to compare the RVQ and MP based ROI image coders and JPIP. However, in terms of functionality of methods the followings are noticeable: While the RVQ and MP based ROI image coding can provide a completely distinctive ROI(s) and background, by properly assigning values to α and R_1, the methods can also provide a gradual quality change from the ROI to the background. On the other hand, for JPEG-2000 this feature has not been considered. In JPEG-2000, if the client changes its ROI, the server may need to reorder the bit-stream and send the data which has not yet been sent according to its cache model of the client . In the RVQ and MP based ROI image coding schemes this reordering is not necessary in the case of ROI change.

In chapter 5, we have addressed MP based coded images that are transmitted over a memoryless noisy channel, and present a novel JSC decoding and error concealment scheme, which exploits the residual redundancy that exists in the received indices. Based on MMSE estimation, a proposed sub-optimal estimation retrieves the corrupted indices using the correlation with other received neighboring indices. According to the tolerable decoding delay and the computational complexity, we can choose from different variants of the suggested JSC decoding method. The proposed error concealment method exhibits significant visual and quantitative enhancement on the degraded received image signals with no increase in bandwidth requirement. Numerical

results show that, while exploiting all the offered residual redundancy outperforms other suggested reconstruction methods, progressive JSC decoding performance stands close to it with no reconstruction delay. The results also indicate that JSC decoding for the earlier MP stages are more beneficial than that for the later ones.

Matching pursuit signal analysis has lots of similarities to residual vector quantization (RVQ). The proposed JSC decoding method in chapter 5 can be easily modified to be used for RVQ image reconstruction of transmission over a memoryless noisy channel. In this case, the inner product coefficients do not exist and we only deal with image block average values and RVQ code-book indices.

6.2 Future Directions

The following relevant extensions of our work can be considered as future research directions.

- In chapter 5, the channel is assumed memoryless. An extension of the proposed JSC decoding method would be to consider channels with memory and fading channels. Under such assumption, many of the required *a posteriori* probabilities presented in this work may need to be modified.

- Throughout this thesis, we never used entropy codes to further compress the source coder bit-stream. The effect of entropy coding on the rate distortion performance of RVQ or MP based image coding and transmission system can also be a future work. In this direction, joint source-channel decoding of variable length codes (VLC) can be investigated.

- For matching pursuit image coding, the dictionary optimization is an

interesting future work. Dictionaries can be adopted according to image context. Dictionaries with variable element dimensions, from a tiny element of 1×1 pixel to a large element of size close to the whole image dimension, are difficult to be stored, but, they can represent an image better than elements with fixed dimensions. In addition to dictionary storage and rate-distortion performance issues, there is also a question of computational complexity, which needs further investigation, when these alternative MP dictionaries are employed.

- The presented index correlation of RVQ and MP based method can also be exploited for further signal compression. By knowing the required *a posteriori* probabilities of indices, entropy coding schemes such as Huffman or arithmetic lossless coding methods can be used to remove the redundancy of MP or RVQ coded bit-stream. This can also be an interesting future direction.

- In chapter 5, the a posteriori probabilities were calculated statistically using a large set of test images. The adopted method has the advantage of off-line probability calculation and lower real-time computation cost. However, there exist some other methods in the literature, such as forward/backward recursion [77], for estimating those probabilities. A research direction can be investigation of the JSC decoding presented in this thesis for image coding based on matching pursuit when the *a posteriori* probabilities are estimated by those alternative methods.

Appendix A

Foveated Video and Fixation Point Tracking

A.1 Foveation

The density of the cone receptors and ganglion cells on the human retina is not uniform and exponentially decreases as a function of eccentricity (Fig. A.1). Therefore, vision is a process of nonuniform sampling of an image such that the highest resolution information belongs to the area of image whose projection on the retina falls onto the center of the retina (the fovea) and the perceptible resolution drops rapidly away from this point [58].

There have been increasing research works on different approaches toward profiting from the above fact for perceptual image and video compression. These approaches are called *foveation*. One foveation approach is to use space variant (log-map) artificial image sensors and nonuniform sampling of the image source [87]. Very similar idea can be applied on uniform sampled images by applying a space variant re-sampling scheme which resembles human retina. Using some nonlinear logarithmic mapping (fish-eye transform) in order to

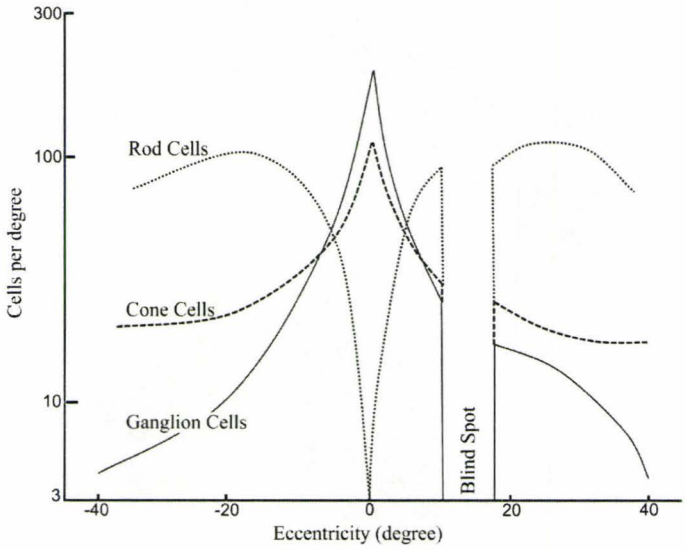


Figure A.1: Visual cells density on human retinal

pre-emphasis on the region of interest (ROI) and then an inverse mapping (de-emphasis) can also mimic the nonlinear spatial sampling of human eyes [59]. A high resolution perception of an image area can be considered as a filtered version of the original image in that area with a high cut-off frequency. This is the basic idea for another approach to foveation in which foveated images are obtained by applying a bank of low-pass filters with different cutoff frequencies. This method is very useful when the foveated images or video frames are to be compressed by DCT based compression schemes such as JPEG, MPEG, and H.263 [60, 61, 88, 89]. Using wavelet image coding algorithms, such as the embedded zero-tree wavelet (EZW) and the set partitioning in hierarchical trees (SPIHT) algorithms is another approach to image foveation which provides a progressive image coding [62].

In order to implement a foveation scheme, the first step is to find the fixation point. In this regard, there are different methods such as object class detection techniques [90], view point eye tracking methods, and using any

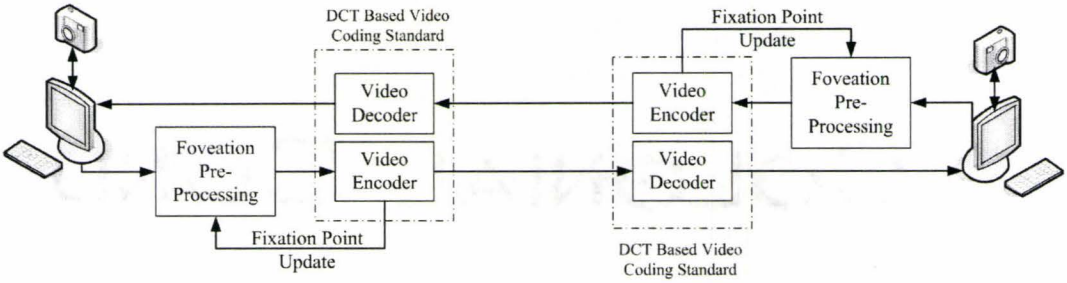


Figure A.2: Video foveation block diagram for DCT based video coding standards

pointing devices such as a mouse. Face detection is regarded as a special case of object class detection and can be employed in video conferencing foveation application. View point eye tracking methods require equipments such as helmet which carries gaze tracker devices. Using pointing devices such as mouse is very inexpensive and can be used to locate any kind of object in an image or a video frame.

In a video application (such as video conferencing using H.263 standard), the fixation point may be moved in any video frame, due to, for example, the movements of the conversing person or camera movements. These changes must be applied to the foveation system by updating the fixation point. Automatic tracking of fixation point eases the task of fixation point update, specially when a pointing device is used. In this appendix, a new automatic fixation point update mechanism for foveated H.263 or MPEG video standards is presented. The foveation method leaves the video compression standard intact and acts as a frame pre-processing unit. The update information regarding fixation point is extracted from the motion compensation mechanism existing in the video standard.

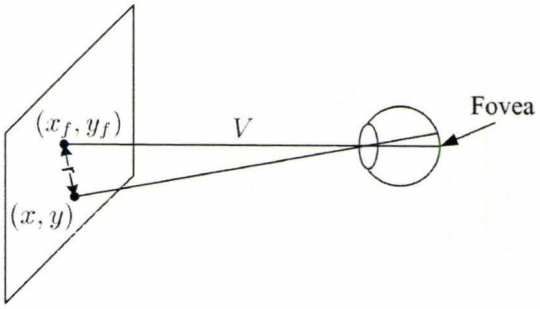


Figure A.3: Pixel projection on human eye

A.2 DCT Based Video Foveation

Fig. A.2 displays a schematic diagram of foveated DCT based video conferencing. Using the initial fixation point indicated by the user from other side of a conversation, the pre-processing unit performs foveation on the first incoming video frame. The selected foveation approach is to employ a bank of low-pass filters with different cut-off frequencies based on an empirical model presented in [89]. The normalized maximum detectable frequency for different location of an image is modeled by the following equation:

$$f_c(x, y, x_f, y_f, V) = 1/\left\{1 + K \tan^{-1}\left(\frac{\sqrt{(x - x_f)^2 + (y - y_f)^2}}{V}\right)\right\} \quad \text{(A.1)}$$

where (x, y) are coordinates of the location in image, while (x_f, y_f) is fixation point, the coordinates of the image pixel whose projection falls on the viewers fovea (Fig. A.3). In (A.1), $K = 13.75$ and V is viewing distance from the image. All distances and coordinates are measured by physical dimension of each pixel. This empirical cut-off frequency can then be quantized, in order to have a limited number of necessary low-pass filters (Fig. A.4). Since most of the high frequency content of each frame has been discarded, these frames can be compressed more efficiently.

The foveated frame is then delivered to the video encoder. The video

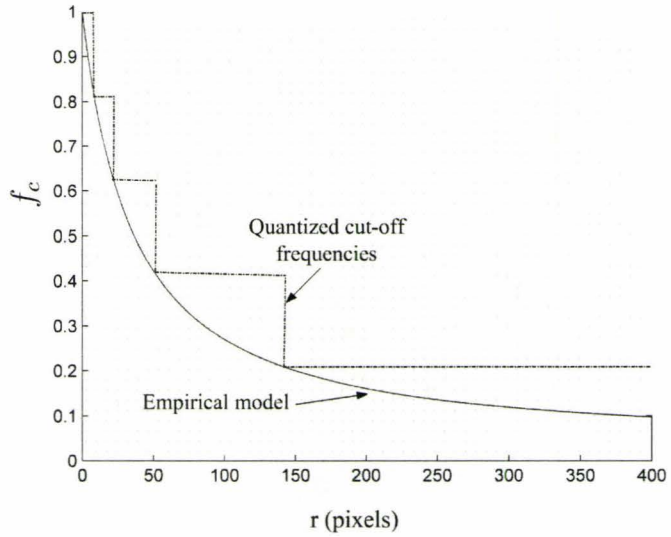
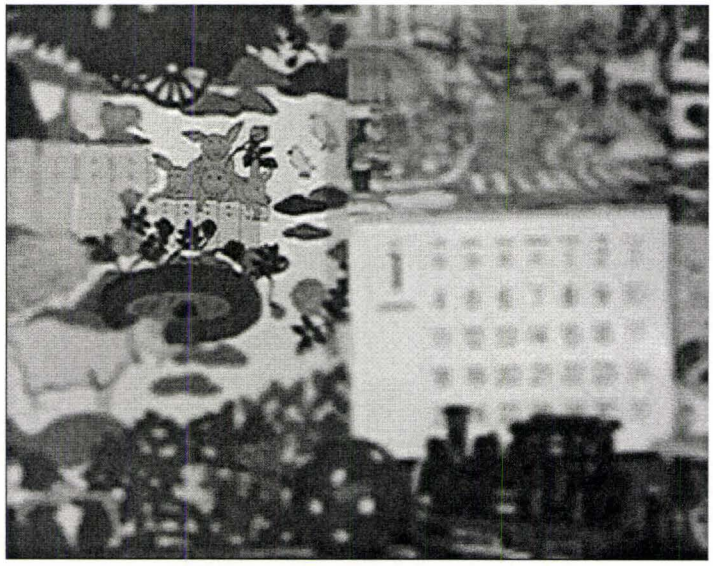


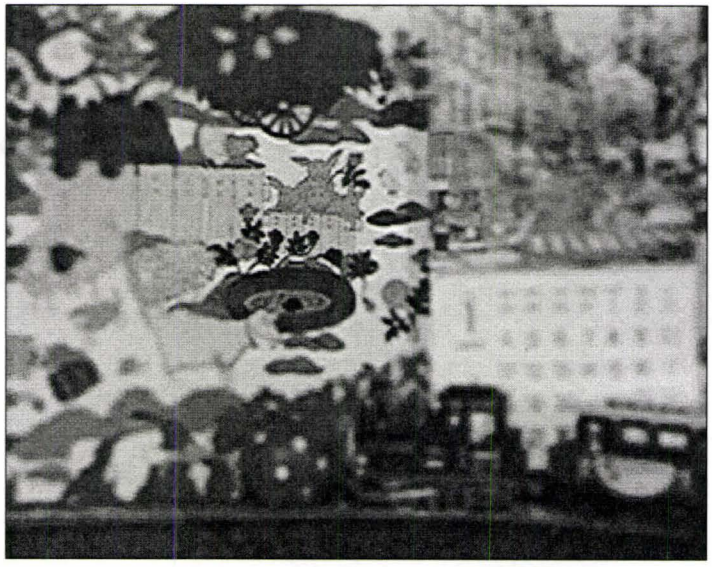
Figure A.4: Cut-off frequency as a function of distance from fixation point when $V = 500$ pixels

encoder calculates the motion vectors. The motion vector associated to the location of fixation point has the information of fixation point movement. This information helps foveation pre-processing unit operate based on new location of fixation point. Figures A.5(a) and A.5(b) show the foveated frames in two different frames using filter banks foveation approach and H.263 video coding standard. In these two figures, the fixation point is around piglets where it has been moved frame by frame in "mobile" test video.

The fixation point tracking method can be used in region of interest video coding as well. In case of multiple regions of interest, the tracking method can update the location of ROIs, frame by frame. Fig. A.6 shows a frame of "news" test video with two regions of interest.



(a)



(b)

Figure A.5: Automatic fixation point movement (a) Frame 100 of mobile test sequence (b) Frame 200 of mobile test sequence

Figure A.6: Multiple ROI video compression implementation using filter banks foveation pre-processing

Appendix B

Reed-Solomon Error Correcting Codes

The topics of error control coding and more specifically Reed-Solomon codes (also known as RS or rs codes) are extensive and require finite field (also called Galois field) mathematics, which is beyond the scope of this thesis. This appendix contains an overview of Reed-Solomon codes. For a profound study of these codes, interested readers may refer to [4, 91].

Introduced by Irving Reed and Gustave Solomon in 1960 [92], Reed-Solomon codes are a subset of a larger class of nonbinary, cyclic, linear block error correcting codes called BCH[†] codes [4]. RS codes have been widely used in a variety of digital storage applications, such as compact discs (CDs), digital versatile discs (DVDs), and barcodes. These codes are also used in wireless communications, high speed modems (such as ADSL), digital television, and satellite communications. An interesting historic application of Reed-Solomon codes has been deep space data communication (across many millions of miles with incredibly low power) with Voyager II space probe which explored Jupiter

[†]BCH codes are named after their inventors: Bose and Chaudhuri, and Hocquenghem.

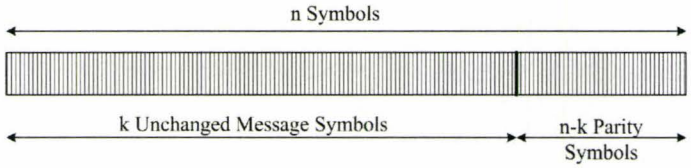


Figure B.1: RS(n, k) code

in late 1960's. By the expense of adding extra information (parity bits) to the original message and tolerating more encoding/decoding complexity, RS codes are capable of correcting errors caused, for example, by channel noise in digital communication application and disc scratches in data storage applications. Reed-Solomon codes can detect and correct a number of digital data errors. This number depends on the amount of redundancy added to the original data.

A Reed-Solomon code RS(n, k) is made up of a block of n m-bit symbol sequence ($m > 2$), where k symbols of them are the original message symbols ($k < n$) and $n - k$ of them are parity check symbols appended to the message symbols (Fig. B.1). $t = \lfloor \frac{n-k}{2} \rfloor$ is the symbol error capability of the code, where $\lfloor x \rfloor$ is the largest integer not exceeding x. Since the original message with no modification is included in RS codewords, these codes are systematic. Reed-Solomon codes exist for all n, k, and m as long as

$$0 < k < n < 2^m + 2 \tag{B.1}$$

For most conventional RS codes $n = 2^m - 1$ and $n - k = 2t$, where $3 \leq m \leq 8$. Out of 2^{mn} possible bit arrangement for an n symbol block, only 2^{mk} are valid codewords and Reed-Solomon coding makes sure that these codewords distributed with largest minimum distance[‡]. [93]. A Reed-Solomon decoder is able to correct up to t received m-bit symbols no matter one or all bits of the

[‡]Minimum distance measure determines the correction ability of a code. For nonbinary codes, such as Reed-Solomon codes, the distance between two codewords is similar to Hamming distance and measured by the number of symbols in which the codewords differ. For RS codes minimum distance is $d_{min} = n - k + 1$.

symbols are corrupted. This makes RS codes an excellent choice for controlling burst noise.

Reed-Solomon codes work based on a theorem in linear algebra which says any distinct k points uniquely determine a polynomial of degree less than k. Using finite field $GF(2^m)$ mathematics, RS encoder finds the $k-1$ polynomial related to the k message symbols. This polynomial is then evaluated at $n-k$ more points and the message symbols along with these $n-k$ parity symbols are sent. The decoder can recover the original message as long as sufficient code symbols are received correctly. Like correcting a curve by interpolating the gaps, RS decoders can bridge the errors and recover the message polynomial and the message.

Reed-Solomon codes, like any linear code, are able to correct up to $2t$ *erasure*. Erasures are symbol errors whose locations in the codeword are know in advance (for example by using side information). These codes can recover the correct message as long as $2E + S < n - k$, where E is the number of erasures and S is the number of errors.

Since addition and multiplication in finite field $GF(2^m)$ is well suited for hardware implementation, Reed-Solomon coding can be easily implemented by combinational logic. Digital signal processors (such as TMS320C6400) can also be used for RS coding realization. There are also routines in MATLAB and C programming designated for Reed-Solomon encoding/decoding.

Bibliography

[1] K. Sayood, *Introduction to Data Compression*, third edition, Morgan Kaufmann publishers, USA, 2006.

[2] W.B. Pennebaker and J.L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, 1993.

[3] D.S Taubman, M.W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, Kluwere Academic Publisher, 2002.

[4] S. Lin, D.J. Costello, *Error Control Coding: Fundumentals and Applications*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983.

[5] A. Ebrahimi-Moghadam and S. Shirani, "Progressive scalable interactive region-of-interest image coding using vector quantization", *IEEE Trans. on Multimedia*, vol. 7, no. 4, pp. 680-687, Aug. 2005.

[6] —,"Image foveation based on vector quantization", *Proc. IEEE Data Compression Conference* , pp. 426, 25-27 March 2003.

[7] —, "Unequal error protected ROI image transmission based on vector quantization", *Proc. IEEE CCECE*, vol. 1, pp. 591-594, May. 2004.

[8] —, "Matching pursuit based region-of-interest image coding", *IEEE Trans. On Image Processing*, vol. 16, no. 2, pp. 406-415, Feb. 2007.

[9] —, "Novel progressive region of interest image coding based on matching pursuits", *IEEE International Conference on Multimedia and Expo ICME-2006*, pp. 737-740, July 2006.

[10] —, "Error Concealment of Matching Pursuit Coded Images Using Joint Source-Channel Decoding", *IEEE Trans. on Image Processing*, Submitted.

[11] —, "Sub-optimal MMSE based joint source/channel decoding of a matching pursuit coded image bit-stream over a memoryless noisy channel", *Proceeding IEEE International Conference on Multimedia and Expo ICME-2008*, pp. 1385-1388, June 2008.

[12] N.S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984.

[13] ITU-T, *Video coding for low bit rate communication*, ITU-T Rec. H.263, version 1, Nov. 1995, version 2, Jan 1998, version 3, Nov. 2000.

[14] F. Pereira and T. Ebrahimi, *The MPEG-4 Book*, IMSC Press Multimedia Series, 1993.

[15] C. E. Shannon,"A mathematical Theory of communication," *Bell Syst. Tech. J.*, vol. 27, nos. 3 and 4, pp. 379-423 and 623-656, July and Oct. 1948.

[16] C. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE National Convention Record, Part4*, pp. 142-163, 1959.

[17] W.H.Equitz, "A new vector quantization clustering algorithm," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 37, no. 10, pp. 1568-1575, Oct. 1989.

[18] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantization design", *IEEE Trans. on Communications*, vol. COM-28, no. 1, pp. 84-95, Jan. 1980.

[19] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, third edition, Elsevier, USA, 2006.

[20] S.P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. on Information Theory*, vol. IT 28, no. 2, pp. 129-137, March 1982.

[21] A. Gersho and R. Gray, *Vector Quantization and Signal compression*, Kluwer academic publishers, Boston, 1992.

[22] E.E. Hilbert, "Cluster compression algorithm - a joint clustering data compression", *Technical Report JPL Publication*, pp. 43-77, NASA, 1977.

[23] Two-dimensional vector quantization animation using splitting technique. http://www.data-compression.com/vqanim.shtml

[24] N.M. Nasrabadi and Y. Feng, "Vector quantization of images based upon the Kohonen self-organizing feature map," *IEEE International Conference on Neural Networks*, vol. 1, pp. 101-105, USA, 1988.

[25] J. Vaisey and A. Gersho, "Simulated annealing and codebook design", *IEEE ICASSP*, vol. 2, pp. 1176 - 1179, USA, 1988.

[26] K. Zeger, J. Vaisey, and A. Gersho, "Globally optimal vector quantizer design by stochastic relaxation", *IEEE Trans. Signal Processing*, vol. 40, no. 2, pp. 310 - 322, Feb. 1992.

[27] K. Krishna, K.R. Ramakrishna, and M.A.L. Thathachar "Vector quantization using genetic k-means algorithm for image compression", *IEEE*

International Conference on Information, Communications and Signal Processing (ICICS), vol. 3, pp. 1585-1587, Sep. 1997.

[28] M.J. Sabin and R.M. Gray, "Product code vector quantizers for waveform and voice coding", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 3, pp. 474-488, June 1984.

[29] N.M. Nasrabadi, R.A.King, " Image coding using vector quantization: a review", *IEEE Trans. on Communications*, vol. 36, no. 8, pp. 951-971, Aug. 1988.

[30] C.F. Barnes and R.L. Frost, "Vector quantizers with direct sum codebook", *IEEE Trans. on Information Theory*, vol. 39, no. 2, pp. 565-580, March 1993.

[31] B.H. Juang and A.H. Gray, "Multiple stage vector quantization for speech coding", *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 82)*, pp. 597-600, April 1982.

[32] C.F. Barnes, *Residual Quantizers*, Ph.D. thesis, Brigham Young University, Provo, Utah, 1989.

[33] C.F. Barnes, S.A. Rizvi, N.M. Nasrabadi, "Advances in residual vector quantization: a review", *IEEE Trans. Image Processing*, vol. 5, no. 2, pp. 226-262, Feb. 1996.

[34] C.F. Barnes, "Adaptive successive approximation quantization of image waveforms with efficient codebook updates", *Proc. ICIP 94, IEEE International Conference on Image Processing*, vol. 3, pp. 876-880, Nov. 1994.

[35] S.A. Rizvi and N.M. Nasrabadi, "Residual vector quantization using a multi-layer competitive neural networks", *IEEE Jornal on Selected Areas in Communications*, vol. 12, no. 9, pp. 1452-1459, Dec. 1994.

[36] S. S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit", *SIAM Journal on Scientific computing*, vol. 20, no. 1, pp. 33-61, 1998.

[37] B.K. Natarajan, "Sparse approximate solutions to linear system", *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227-234, April 1995.

[38] NP-hard, http://en.wikipedia.org/wiki/NP-hard.

[39] J.H. Friedman, W. Stuetzle, "Projection pursuit regression", *Journal of the American Statistical Association*, vol. 76, pp. 817-823, 1981.

[40] P. J. Huber, "Projection Pursuit", *The Annals of Statistics*, vol. 13, no. 2, pp. 435-475, 1985.

[41] G. Davis, *Adaptive Nonlinear Approximations*, PhD thesis, New York University, September 1994.

[42] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397-3415, Dec. 1993.

[43] R. Rezaiifar and H. Jafarkhani, " Wavelet based speech coding using orthogonal matching pursuit", *Proceedings of the Twenty Ninth Annual Conference on Information Sciences and Systems* pp. 88-92, March 1995.

[44] F. Bergeaud, S. Mallat, "Matching pursuit of images", Proceeding IEEE International Conference on Image Processing ICIP95, vol. 1, pp. 53-56, Oct. 1995.

[45] M. Vetterli and T. Kalker, " Matching pursuit for compression and application to motion compensated video coding",Proceeding IEEE International Conference on Image Processing ICIP94, pp. 725-729, Nov. 1994.

[46] R. Neff A. Zakhor, and M. Vetterli, "Very low bit rate video coding using matching pursuits", *Proceedings of SPIE Conference on Visual Communication and Image Processing*, vol. 2308, pp. 47-60, Sept. 1994.

[47] P.J. Phillips, " Matching pursuit filters applied to face identification", Proceedings of SPIE, vol. 2277, pp. 2-11, July 1994.

[48] Y. Ro, R. Neff, and A. Zakhor, "Matching pursuit data acquisition in magnetic resonance imaging", *Proceedings of SPIE Conference on Medical Imaging*, vol. 3032, pp. 530-540, Feb. 1997.

[49] M. Akay and E. Mulder,"Matching pursuit analysis of fetal heart rate variability during maturation", *Proceedings of the 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1624-1625, Nov. 1996.

[50] J.C. Fu, C.A. Troy, and P.J. Phillips, "A matching pursuit approach to small drill bitbreakage prediction", *International Journal of Production Research*, pp. 3247-3261, Sept. 1999.

[51] J.L. Lin, W.L. Hwang, and S.C. Pei, "Video compression based on orthonormal matching pursuits", *Proceedings, IEEE International Symposium on Circuits and Systems ISCAS 2006*, pp. 5423-5426, May 2006.

[52] R. Neff and A. Zakhor, "Matching-pursuit video coding, part I: Dictionary approximation", *IEEE Trans. on Circuits System for Video Technology*, vol. 12 no. 1, pp. 13-26, Jan. 2002.

[53] R. Neff and A. Zakhor, "Matching-pursuit video coding, part II: Operational models for rate and distortion", *IEEE Trans. on Circuits System for Video Technology*, vol. 12 no. 1, pp. 27-39, Jan. 2002.

[54] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*, Second edition, MIT Press, 2001.

[55] S. Mallat, *A Wavelet Tour of Signal Processing*, Second edition, Academic Press 1999.

[56] A. Akodras, C. Christopoulos, and T. Ebrahimi,"The JPEG 2000 still image compression standards,", *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36-58, Sep. 2001.

[57] J. In, S. Shirani, F. Kossentini,"On RD optimized progressive image coding using JPEG,",*IEEE Trans. on Image Processing*, vol. 0, no. 11, pp. 1630-1638, Nov. 1999.

[58] L.K. Cormack,"Computational models of early human vision,"in *Handbook of Image and Video Processiong*, A. Bovik, Ed. New York, Academic Press, 2000.

[59] A. Basu and S. Licardie,"Modeling fish-eye lenses,", *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1822-1828, July 1993.

[60] H.R. Sheikh, S. Liu, B.L. Evans, and A.C. Bovik, " Reat-time foveation techniques for H.263 video encoding in software", *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1781-1784, May 2001.

[61] S. Lee, *Foveated video compression and visual communication over wireless and wireline networks, Ph.D Dissertation*, The Univ. of Texas at Austin, May 2000.

[62] Z. Wang and A. Bovik, "Embedded foveation image coding", *IEEE Trans. Image Processing*, vol.10, no. 10, pp. 1397 -1410, Oct. 2001.

[63] W.A.H. Mousa, M.A.U. Khan,"Design and analysis of entropy-constrained reflected residual vector quantization", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP 02) , vol. 3, pp. 2529-2532, May 2002.

[64] M.A. Khan, M.J. Smith, S.W. McLaughlin, " Jointly optimized trellis-coded residual vector quantization", *IEEE Trans. Communications*, vol. 49 no. 6 , pp. 937-942, June 2001.

[65] T.P. Chen, T. Chen , "Adaptive joint source-channel coding using rate shaping", *IEEE Proceeding ICASSP* , Vol.2, pp 1985-1988, 2002.

[66] A. Shoa, S. Shirani, "Progressive Coding of a Gaussian Source Using Matching Pursuit", *IEEE Trans. on Signal Processing*, vol. 56, no. 2, pp. 636-649, Feb. 2008.

[67] R. Neff, A. Zakhor, "Matching pursuit video coding at very low bit rates," *IEEE Trans. Circuits Syst. Video Technology*, pp 158-171, Feb. 1997.

[68] K. Sayood and J. C. Brokenhagen ,"Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. Communications* , vol. 39, no. 6 pp. 838-845, June 1991.

[69] A. Alavi, R. Link, and S. Kallel" Adaptive unequal error protection for subband image coding," *IEEE Trans. on Broadcasting*, vol. 46, no. 3 pp. 197-205, Sep. 2000.

[70] L. Bin, H. Feng, S. Lifeng, and Y. Shiqiang," Optimized Rate Allocation for Unbalanced Multiple Description Video Coding Over Unreliable Packet Network," in *Proc. IEEE ICME*, vol. 1, pp. 105-108, July 2006.

[71] N. Farvardin, " A study of vector quantization for noisy channels," *IEEE Trans. on Information Theory*, vol. 36, no. 4, pp. 799-809, July 1990.

[72] M. Park and D. J. Miller, "Improved image decoding over noisy channels using minimum mean-squared estimation and Markov mesh," *IEEE Trans. on Image Processing*, vol. 8, no. 6, pp. 863-867, June 1999.

[73] S. Emami and S. Miller, "DPCM picture transmission over noisy channel with the aid of a Markov model," *IEEE Trans. on Image Processing*, vol. 4, no. 11, pp. 1473-1481, Nov. 1995.

[74] R. Link and S. Kallel, "Optimal use of Markov models for DPCM picture transmission over noisy channels," *IEEE Trans. on Communications*, vol. 48, no. 10,pp. 1702-1711, Oct. 2000.

[75] W. Xu, J. Hagenauer, and J. Hollmann, "Joint source-channel decoding using the residual redundancy in compressed images," *in Proc. International Conference on Communications ICC96*, pp. 142-148, June 1996.

[76] N. Phamdo and N. Farvardin, "Optimal detection of discrete Markov sources over discrete memoryless channels–Aplications to combined source-channel coding," *IEEE Trans. on Information Theory*, vol. 40, no. 1, pp. 103-186, Jan. 1994.

[77] F. Lahouti and A. K. Khandani," Reconstruction of predictively encoded signals over noisy channels using a sequence MMSE decoder," *IEEE Trans. on Communications*, vol. 52, pp. 1292-1301, Aug. 2004.

[78] D.J. Miller and M. Park, "A sequence based approximate MMSE decoder for source coding over noisy channels using discrete hidden Markov models," *IEEE Trans. on Communications*, vol. 46, no. 2, pp. 222-231, Feb. 1998.

[79] K. Sayood, H. Otu, and N. Demir, "Joint source/channel coding for variable length codes," *IEEE Trans. on Communications*, vol. 48, no. 5, pp. 787-794, May 2000.

[80] K.P. Subbalakshmi and J. Vaisey, "On the joint source-channel decoding of variable-length encoded sources: The BSC case," *IEEE Trans. on Communications*, vol. 49, no. 12, pp. 2052-2055.

[81] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Trans. on Communications*, vol. 48, no. 1, pp. 1-6, Jan. 2000.

[82] M. Skoglund, "Soft decoding for vector quantization over noisy channels with memory," *IEEE Trans. on Information Theory*, vol. 45, no. 4, pp. 1293-1307, May 1999.

[83] S.M. Key, *Fundamentals of Statistical Signal Processing: Estimation Theory*, first edition, Prentice Hall, 1993.

[84] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, third edition, McGraw Hill, 1991.

[85] D. Taubman, R. Prandolini,"Architecture, philosophy and performance of JPIP: internet protocol standard for JPEG2000 ", Proc. International Symposium on Visual Communications and Image Processing (VCIP2003), Switzerland, July 2003

[86] A. Akodras, C. Christopoulos, T. Ebrahimi, "The JPEG 2000 still image compression standards," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36-58, Sep. 2001.

[87] R.S. Wallace, P.W. Ong, B.B. Bederson, and E.L.Schwartz,"Space variant image processing,", *International Journal of Computer Vision*, vol. 13, no. 1, pp. 71-90, 1994.

[88] S. Lee and A.C. Bovik, "Very low bit rate foveated video coding for H.263", *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3113-3116, March 1999.

[89] W.S. Geisler and J.S. Perry,"A real-time foveated multiresolution system for low-bandwidth video communication", *Proc. SPIE*, vol. 3299, pp. 294-305, July 1998.

[90] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", *Proceedings IEEE Computer Vision and Pattern Recognition CVPR 2001*, vol. 1, pp. I511-I518, Dec. 2001.

[91] S.B. Wicker and V.K. Bhargava, *Reed-Solomon Codes and Their Applications*, IEEE Press, USA, 1983.

[92] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields", *SIAM Journal of Applied Math.*, vol. 8, pp. 300-304, 1960.

[93] B. Sklar, *Digital Communications: Fundamentals and Applications*, second edition, Prentice Hall, USA, 2001.