11-1-1996

ADAPTIVE MULTIRESOLUTION IMAGE AND VIDEO COMPRESSION AND PRE/POST-PROCESSING OF IMAGE AND VIDEO STREAMS

Hamid R. Rabiee
Purdue University School of Electrical and Computer Engineering

R. L. Kashyap
Purdue University School of Electrical and Computer Engineering

Rabiee, Hamid R. and Kashyap, R. L., "ADAPTIVE MULTIRESOLUTION IMAGE AND VIDEO COMPRESSION AND PRE/POST-PROCESSING OF IMAGE AND VIDEO STREAMS " (1996). *ECE Technical Reports.* Paper 91.
http://docs.lib.purdue.edu/ecetr/91

ADAPTIVE MULTIRESOLUTION IMAGE AND VIDEO COMPRESSION AND PRE/POST-PROCESSING OF IMAGE AND VIDEO STREAMS

HAMID R. RABIEE
R. L. KASHYAP

TR-ECE 96-17
NOVEMBER 1996

SCHOOL OF ELECTRICAL
AND COMPUTER ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

.ADAPTIVE MULTIRESOLUTION IMAGE AND VIDEO COMPRESSION AND PRE/POST-PROCESSING OF IMAGE AND VIDEO STREAMS*

Hamid R. Rabiee and R. L. Kashyap

School of Electrical and Computer Engineering
1285 Electrical Engineering Building
Purdue University
West Lafayette, IN 47907-1285

ABSTRACT

This report is divided into two sections. In the first section, the focus is on adaptive transform-based image compression and motion compensation at low bit rates. A new adaptive algorithm for image representation and coding is introduced. This algorithm is based on the concept of segmented orthogonal matching pursuits (SOMP), and adaptively selects the best representation from an overcomplete dictionary of wavelet functions. In the second section, the pre-processing and post-processing of images and video streams are focused on. A new robust nonlinear filter based on the theory of generalized maximum likelihood and order statistics (GMLOS) is introduced. It is shown that this filter is an l_2-optimal order statistics filter and some of its properties are proved. A novel algorithm based on wavelet decomposition, variable size kernel GMLOS filters, and soft thresholding for removing the blocking effects in block-based transform coding techniques is introduced. Finally, a simple algorithm for cell packing in ATM networks is introduced, and a novel algorithm for error concealment of images and video streams, based on Multi-directional Recursive Nonlinear Filtering (MRNF) with GMLOS filters is introduced.

---

* Financial support provided by the Innovative Science and Technology (IST) program of the BMDO, monitored by the Office of Naval Research (ONR) under contract ONR N00014-91-J-4126 is hereby thankfully acknowledged.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

Page

THIS PAGE WAS INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE WAS INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure Page

Figure Page

1. INTRODUCTION

1.1 Introduction

In modern interactive communication systems two-way connections are used to provide voice, data, image, and video to the users. There are four fundamental operations associated with these systems: pre-processing, *coding/decoding* (CODEC), network *interfacing* (NIU), and post-processing. The main building blocks of a modern interactive communication system as depicted in Fig. 1.1.

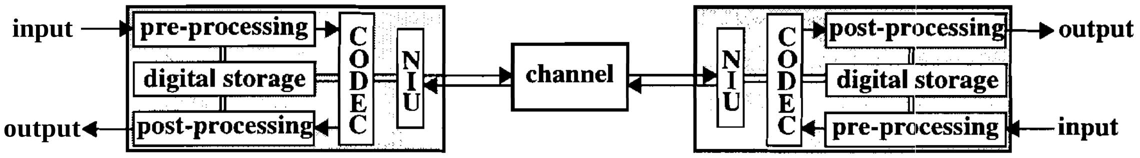


Fig. 1.1. The main building blocks of a modern interactive digital communication system. CODEC stands for coder/decoder, and NIU for Network Interface Unit.

Most of the required bandwidth in these systems is occupied by visual information. Since the amount of available bandwidth is limited by the channel, the CODEC should be able to encode the visual information at very low bit rates. The coding algorithms for visual information in these systems belong to the class of *lossy* image and video compression techniques[1], which usually introduce visible distortions to the original image. The amount of distortion is a function of the data rate used, and the particular characteristics of the source image. In addition, the compressed data is vulnerable to transmission impairments. In this case, the transmission errors may result in various types of degradations or partial loss of the data, depending on the network interface algorithm and channel characteristics. Some of these coding and transmission impairments can be reduced by using a set of pre-processing and post-processing algorithms on the data. The

---

[1] *In some applications such as medical imaging systems lossless compression may be required.*

following topics are studied in this thesis.

1.2 Multiresolution Image and Video Compression with Adapted Bases

A large class of image compression algorithms exploit the fact that applying a linear transformation to the input signal can result in a coding gain. The traditional approach in this class of compression techniques is to use a fixed transformation matrix. For example, the emerging image and video compression standards such as JPEG, H.261, and MPEG use. Discrete Cosine Transform (DCT), which is a fixed transform with sinusoidal bases [1]. Another fixed transform that has attained popularity in image coding applications is the Discrete Wavelet Transform (DWT) [2, 3]. DWT offers good frequency selectivity at lower frequencies and good time selectivity at higher frequencies, and when coupled with a quantization strategy that exploits this property, it has achieved siignificantly better performance over other transforms such as DCT [2, 3].

While coding algorithms with fixed transforms can be useful for a specific class of signals, they may not be adequate for the characterization of a more general class of signals with unknown or time varying characteristics [3]. In this case, it is advantageous to use: a transform that is signal dependent. The idea of adapting the transformation to the signal in image compression applications is related to the field of universal coding [4] and has gained popularity in recent years [3, 5].

In transform based video compression techniques, usually a hybrid approach based on the concept of *intra-frame* and *inter-frame* coding is used [6, 7]. In the intra-frame mode of operation, spatial redundancy is exploited by treating the video frames as still images. In the inter-frame mode of operation, temporal redundancy is often exploited by means of block matching Motion Estimation and Motion Compensation (MEMC) techniques. In the inter-frame mode, the current frame is coded by motion vector and residual error information that has been predicted from previous or possibly future frames. The intra-frame and inter-frame information is then combined to create the encoded video stream.

In the first part of this thesis, a new adaptive signal expansion technique called Segmented Orthogonal Matching Pursuit (SOMP) is introduced. This algorithm is then used for multiresolution image compression, and it is shown that at lower bit rates it performs better than traditional fixed transform coding techniques [8]. Moreover, it is shown that the process of separate prediction and residual coding in hybrid MEMC video compression techniques can be solved in a unified framework by using a modified version of the SOMP algorithm. Furthermore, it is shown that at lower bit rates, this new algorithm performs better than the traditional hybrid MEMC techniques [9].

1.3 Pre-Processing and Post-Processing of Images and Video Streams

Pre-processing refers to spatial and temporal image processing algorithms, such as noise smoothing and decimation, that are applied to the source image prior to data compression. Similarly, post-processing refers to image processing algorithms that are applied to the decompressed data to enhance its quality and appearance or to restore the data lost during the transmission. Various image degradations, coding artifacts, and transmission impairments can be corrected by using these techniques [10, 11]. In the second part of this thesis, a new robust nonlinear filter based on the theory Generalized Maximum Likelihood and Order Statistics (GMLOS) is introduced [12, 13]. It is an l_2-optimal order statistic filter that is well suited for the pre-processing and post-processing of images and image sequences. Moreover, this filter is used in two novel algorithms for the de-blocking of blocking artifacts in block-based transform coding [14], and the error concealment of encoded image and video streams over ATM networks [15].

1.4 Measures of Performance

In this thesis the quality of the filtered or decoded images and image sequences is often referred to. In image processing applications, there are several different criteria to measure the performance of visual information. These measures include the Mean Opinion Score (MOS), Picture Quality Scale (PQS), Mean Absolute Error (MAE), Mean Square Error (MSE), and Peak Signal-to-Noise Ratio (PSNR) [16, 17]. The most widely used quantitative measures of quality are MSE and PSNR. Although these measures are not always correlated with the perceived subjective quality of the Human Visual System (H'VS), their widespread use is justified by their simplicity. In this thesis MSE and MAE are used as quantitative measures of quality for filtered images, and PSNR for encoded images and image sequences. Visual subjective quality may also be used as a measure of performance. Aside from visual quality, other measures of performance such as computational complexity are also considered in this thesis.

1.5 Contributions of this Thesis

The contributions of this thesis can be summarized as follows.

- A new algorithm called Segmented Orthogonal Matching Pursuit (SOMP) is introduced for adaptive signal expansion.
- It is shown that SOMP performs better than the original matching pursuit algorithm in terms of sparsity of representation and convergence.
- A new multiresolution image compression algorithm based on the SOMP algorithm and an overcomplete dictionary of wavelet atoms are introduced.
- A new algorithm for MEMC is introduced that unifies the process of separate prediction and residual coding into a unified framework by using a modified SOMP algorithm. Moreover, it is shown that at lower bit rates this algorithm performs better than the traditional hybrid block-matching MEMC algorithms.
- A new robust nonlinear filter for pre-processing and post-processing of images and image sequences is introduced. This filter has been designed based on the theory of Generalized Maximum Likelihood and Order Statistics (GMLOS).
- It is shown that this filter is an 1,-optimal order statistics filter and some of its properties are proved.
- A novel algorithm based on wavelet decomposition, variable size kernel GMLOS filters, and soft thresholding for removing the blocking effects in block-based transform coding techniques is introduced.
- A simple algorithm for ATM cell packing is introduced.
- A novel algorithm for error concealment of images and video streams, based on Multi-directional Recursive Nonlinear Filtering (MRNF) with GMLOS filters is introduced.

1.6 Organization of this Thesis

The material presented in the remainder of this thesis is organized as follows. Chapter 2 presents a brief overview of the linear expansion of signals with wavelet bases. In chapter 3, signal expansion with adapted bases is studied. The Segmented Orthogonal Matching Pursuit (SOMP) algorithm is also presented in this chapter, and the design of optimal dictionaries for overcomplete expansions is addressed. Chapter 4 is devoted to multiresolution image compression with the SOMP algorithm and wavelet bases. Chapter 5 provides an algorithm for motion estimation using a modified version of the SOMP algorithm. Chapter 6 presents the GMLOS filter and some of its properties. It also presents a novel algorithm for removing the blockiness in block-based transform coding. Chapter 7 is devoted to the problem of error concealment over ATM networks. Finally, chapter 8 provides the concluding remarks and discusses future research.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

2. SIGNAL EXPANSION AND ANALYSIS WITH WAVELETS

2.1 Introduction

Signal expansion is one of the major components of many algorithms in digital signal and image processing. The idea is to represent a signal by a linear combination of elementary building blocks or atoms that exhibit certain desirable properties. In signal and image processing applications the goal of these decompositions is to achieve sparsity, high resolution, robustness, and speed. In particular, these properties are central to image compression applications at low bit rates. Sparsity leads to the representation of signals with a smaller number of significant coefficients, high resolution results in better subjective quality by capturing the most prominent details of the signal, robustness guarantees that small perturbations will not seriously degrade the quality of the representation, and speed is required in real time applications over the interactive multimedia communication systems.

Until recently, the Discrete Cosine Transform (DCT) has been the most popular signal expansion[1] algorithm for signal and image compression, and the block-based DCT has been adopted by many emerging image and video compression standards such as JPEG, H.261, and MPEG [1]. DCT is a good approximation to the Karhunen-Loève Transform (KLT) which is an optimal transform for stationary first order Markov signals based on the l_2 error measure [18]. While KLT is a signal dependent transform and produces an adapted basis dictionary for signal representation, DCT uses a fixed dictionary of sinusoids as basis functions. This property has led to the design of fast and efficient algorithms for the computation of DCT coefficients, and hence its popularity. Although under stationarity and Markovian assumptions DCT is a fairly good model for signal or image blocks, it fails to exploit the global structure of data. Moreover, in block-based DCT compression

---

[1] ***Discrete-time series expansions are often called discrete-time transforms.***

algorithms the correlation across the block boundaries is not removed. At lower bit rates, this usually results in loss of compression, and annoying blocking *effects* in decoded images [2].

More recently, there has been a growing interest in the representation and compression of signals by using dictionaries of basis functions other than the traditional dictionary of sinusoids. These new sets of dictionaries include Gabor functions [19], chirplets [20], warplets [21], wavelets [22], multi-wavelets [23], and wavelet packets [24]. Among these transformations, the Discrete Wavelet Transform (DWT) has attained more popularity in image compression due to its good performance and the existence of efficient algorithms for computing its coefficients [2]. In contrast to DCT, the wavelet transform gives good frequency selectivity at lower frequencies and good time selectivity at higher frequencies. This trade-off in the Time-Frequency (TF) plane is well suited to the representation of many natural signals and images that exhibit short-duration high-frequency and long-duration low-frequency events. The TF-localization property of the wavelet transform is also very attractive in compression applications, and when exploited with the proper quantization strategy in a coding algorithm, can produce significantly better results than other transforms such as DCT [2].

This chapter is designed to provide a brief overview of the wavelet transform as relevant to signal and image processing applications. Although a number of excellent tutorial papers and books on wavelet theory can be found in the literature [2, 22, 25, 26, 27, 28], this chapter presents a perspective that is particularly important for some of the applications considered in this thesis. It establishes the notational conventions for the algorithms that are adopted in subsequent chapters for the optimal adaptive expansion and compression of signals with wavelet and wavelet packet dictionaries.

2.2 Linear Expansion of Signals with Wavelet Bases

The linear expansion of signals with elementary building blocks or atoms is central to signal and image processing applications. Given any signal x from a Hilbert space $\mathcal{H}$, a set of atoms $\{\varphi_\gamma\}_{\gamma \in \Gamma}$, is desired such that x can be written as

$$x = \sum_{\gamma \in \Gamma} a_\gamma \varphi_\gamma \qquad (2.1)$$

If the space $\mathcal{H}$ is the space of square integrable functions $L_2(\boldsymbol{R})$ or square summable sequences $l_2(\boldsymbol{Z})$, it is infinite dimensional, and it is finite dimensional if $\mathcal{H} = \mathrm{R}^N$, or $\mathcal{H} = \mathrm{C}^N$, where R^N and C^N represent the space of real and complex N-tuples, respectively. If all signals $x \in \mathcal{H}$ can be expanded as in (2.1), the set $\{\varphi_\gamma\}$ is complete for the space $\mathcal{H}$, and there exist a dual set $\{\tilde{\varphi}_\gamma\}$ such that the coefficients a, can be computed as

$$a_\gamma = \int \tilde{\varphi}_\gamma(t)x(t)dt \tag{2.2}$$

when they are real continuous-time functions, or as

$$a_\gamma = \sum_n \tilde{\varphi}_\gamma[n]x[n] \tag{2.3}$$

when they are discrete-time sequences. The above equations represent the inner products of the signal x with φ_γ's, and are often denoted by $\langle \varphi_\gamma, x \rangle$. If the set $\{\varphi_\gamma\}$ is complete, and the φ_γ's are linearly independent ($\langle \varphi_i, \varphi_j \rangle = \delta_{i,j}$, where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise), they constitute an orthonormal basis for $\mathcal{H}$. In this case the basis and its dual are the same, and (2.1) can be written as

$$x = \sum_\gamma \langle \varphi_\gamma, x \rangle \varphi_\gamma \tag{2.4}$$

If the set $\{\varphi_\gamma\}$ is complete and the vectors φ_γ are linearly independent but not orthonormal, then they constitute a biorthogonal basis for $\mathcal{H}$. In this case the basis and its dual satisfy $\langle \tilde{\varphi}_i, \varphi_j \rangle = \delta_{i,j}$, and (2.1) can be written as

$$x = \sum_\gamma \langle \tilde{\varphi}_\gamma, x \rangle \varphi_\gamma = \sum_\gamma \langle \varphi_\gamma, x \rangle \tilde{\varphi}_\gamma \tag{2.5}$$

Finally, if the set $\{\varphi_\gamma\}$ is complete, but the φ_γ's are linearly dependent, they do not constitute a basis for $\mathcal{H}$, and the resulting redundant or overcomplete representation is called a frame [2, 22].

For analytical and practical reasons, the expansion of signals with structured bases is of great interest in signal and image processing applications. In these type of expansions, the basis vectors are related by some elementary operations such as scaling, shifting in time, and shifting in frequency (modulation). The Fourier series expansion, which uses harmonic sinusoids as basis functions, is a classical example of linear signal expansion in signal processing. Although this fast transform is well suited for many applications in signal and image processing, it has a major drawback; the Fourier base:; are not localized in time. Therefore, Fourier analysis works well if the signal is composed of a few stationary components. However, any abrupt change in time in a non-stationary signal is spread out over the whole frequency axis, as shown in Fig. 2.1.

One may attempt to localize the Fourier bases by windowing the data at fixed intervals with a smooth and compactly supported window prior to transformation. This transform is known as the Gabor or Short-Time Fourier Transform (STFT) [2]. The basis functions used in the STFT expansion are well structured (they are related to each other by shift in time and modulation), and produce a linear frequency analysis that partilions the TF-plane into fixed size rectangles called logons [19]. Another alternative to Fourier analysis is the Wavelet Transform (WT). In this transform, the signal is expanded by using the ***shifts*** and scales of a prototype function called the mother wavelet. Although, STFT is useful for the analysis of many types of signals, WT offers a number of advantages over STFT. While it is possible to construct a variety of compactly supported orthonormal bases for WT, there are no suitable orthonormal bases based on STFT. The second advantage comes from the fact that the scales used in the construction of wavelet bases are powers of an elementary scale factor that produces a logarithmic frequency analysis for WT. This property results in a more efficient tiling of the TF-plane by WT analysis, because at higher frequencies (smaller scales) the logons become more localized in time, while at lower frequencies they are more localized in frequency. It is important to note that the localization in time and frequency can not be arbitrarily small and their product is bounded by the Heisenberg inequality [28]. Some of these facts are illustrated in Fig. 2.1.

2.2.1 Continuous Wavelet Transform

In WT the notion of scale is introduced as an alternative to frequency that leads to a time-scale representation. For a continuous signal, the time and scale parameters of WT can be continuous, leading to the Continuous Wavelet Transform (CWT). Consider the family of functions obtained by the scaling and shifting of a mother wavelet $\psi(t) \in L_2(\boldsymbol{R})$ as

$$\psi_{a,b}(t) = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right) \tag{2.6}$$

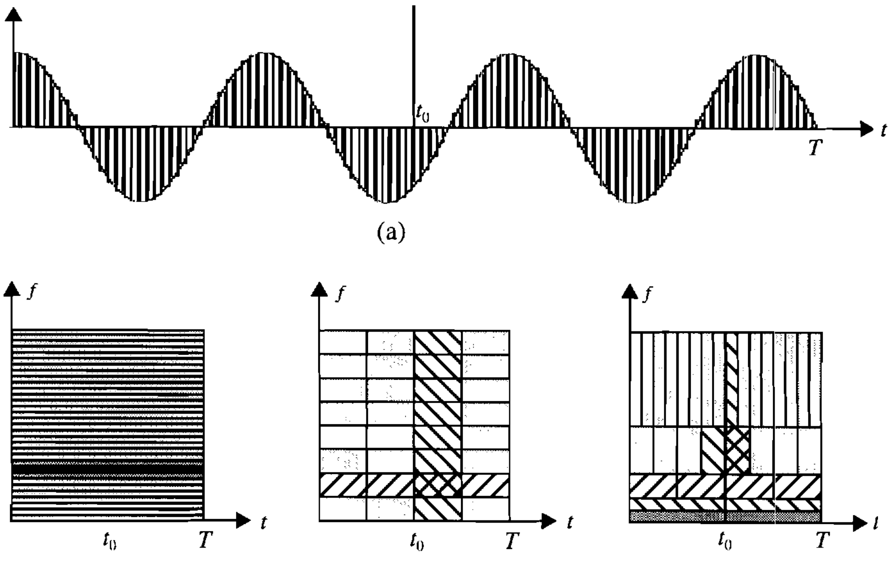


Fig. 2.1. Time-frequency tiling of a signal [2]. (a) Sine wave plus impulse. (b) Discrete-time Fourier series. (c) Short-time Fourier series. (d) Discrete-time wavelet series. The vertical axis represents either increasing frequency or decreasing scale, the horizontal axis represents time, artd darker and stripped regions represent larger coefficients. Clearly, the TF-localization of the WT analysis is superior to STFT or Fourier analysis for this example.

where a, b $\in \mathbf{R}$ and a $\neq 0$. The mother wavelet should satisfy the ***admissibility*** condition

$$C_{\psi} = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty \tag{2.7}$$

where $\Psi(\omega)$ is the Fourier transform of $\psi(t)$. Whenever $\Psi(t)$ decays sufficiently fast, the admissibility condition reduces to

$$\Psi(0) = \int_{-\infty}^{\infty} \psi(t) dt = 0 \tag{2.8}$$

It is important to note that since $\Psi(\omega)$ is zero at the origin and decays at high frequencies, the wavelet spectrum has a band-pass behavior.

If the above requirement is satisfied, then the CWT of a conttinuous function $x(t) \in L_2(\mathbf{R})$ is defined as

$$CWT_x(a, b) = \int_{-\infty}^{\infty} x(t) \psi_{a,b}^{*}(t) dt = \langle x, \psi_{a,b} \rangle \tag{2.9}$$

It has been shown that the function $x(t)$ can be recovered (in the L_2 sense) from its CWT by the following reconstruction formula, known as resolution of *the* identity [22]

$$x(t) = \frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} CWT(a, b)\psi_{a, b}(t)\frac{dadb}{a^2} \tag{2.10}$$

It is possible to have variations in the above reconstruction formula [22]. An important case is when the analysis and synthesis wavelets are different. Let $\psi_1(t)$ and $\psi_2(t)$ denote the analysis and synthesis wavelets, respectively. If the following inequality satisfied

$$\int_{-\infty}^{\infty}\frac{|\Psi_1(\omega)||\Psi_2(\omega)|}{|\omega|}d\omega < \infty \tag{2.11}$$

and CUT,ψ_2 is defined as

$$C_{\psi_1, \psi_2} = \int_{-\infty}^{\infty}\frac{\Psi^*{}_1(\omega)\Psi_2(\omega)}{|\omega|}d\omega \neq 0 \tag{2.12}$$

then the following reconstruction formula holds [22]

$$x(t) = \frac{1}{C_{\psi_1, \psi_2}}\int_{-\infty}^{m}\int_{-\infty}^{\infty}\langle\psi_{1_{a, b}}, x\rangle\psi_{2_{a, b}}(t)\frac{\mathrm{d}a\mathrm{d}b}{\mathrm{a}^2} \tag{2.13}$$

In this framework, $\psi_1(t)$ and $\psi_2(t)$ can have significantly different behaviors, a property that is desirable in many signal and image processing applications. For example, it can be used for the design of biorthogonal systems in which both analysis and synthesis bases are symmetric and compactly supported.

2.2.2 Frames and the Discrete Wavelet Transform

From (2.9), it is clear that CWT produces a redundant representation for a signal, because it maps a continuously indexed function of one variable into a continuously indexed function of two variables. One way to overcome this problem is to sample the (a, b) plane. Let $\mathrm{a} = a_0^m$ where $m \in \boldsymbol{Z}$ and $\mathrm{a}, \neq 1$, then for $\psi_{a, b}(t)$ to cover the whole time axis at a scale a, the discretized parameters are [22]

$$\mathrm{a} = \mathrm{a}_,^m,\quad \mathrm{b} = nb_0a_0^m,\quad a \in \boldsymbol{R}_+, b \in \mathrm{R},\quad m, n \in \boldsymbol{Z},\quad a_0 > 1, b_0 > 0 \tag{2.14}$$

and the discretized family of wavelets can be written as

$$\psi_{m, n}(t) = a_0^{-m/2}\psi(a_0^{-m}t - nb_0) \tag{2.15}$$

Therefore, a frame maps $L_2(\boldsymbol{R})$ to $l_2(\boldsymbol{Z}^2)$ with non-critical sampling of the scale and shift parameters. A wavelet frame can be defined as follows [22]: a family of wavelet functions $\{\psi_{m,n}\}_{m,n \in J}$, in a Hilbert space $\mathcal{H} = L_2(\boldsymbol{R})$ is called a waveletframe, if for all x in $\mathcal{H}$, there exists $0 < A \leq B < \infty$, such that

$$A\|x\|^2 \leq \sum_{m,n} |\langle \psi_{m,n}, x \rangle|^2 \leq B\|x\|^2 \tag{2.16}$$

where A and B are called theframe *bounds.* The transformation that uses a wavelet frame for the linear expansion of signals is called Wavelet Frame Transform (FVFT).

When $A = B$, the frame is called a tightframe, and by virtue of equation (2.16) there exists a reconstruction formula (at least in the weak sense [22]) given by

$$x(t) = A^{-1} \sum_{m,n \in J} \langle \psi_{m,n}, x \rangle \psi_{m,n} \tag{2.17}$$

It is important to realize that a frame is not an orthonormal basis for $\mathcal{H}$. It is only a set of non-independent vectors that spans the space $\mathcal{H}$. A numerically stable reconstruction formula for general frames can be found in [22].

In general, frames represent a middle ground between the CWT and the discrete wavelet transform. In fact, if $\{\psi_{m,n}\}_{m,n \in J}$ is a tight frame, with frame bound $A = 1$, and $\|\psi_{m,n}\| = 1$ for all $m, n \in J$, then $\{\psi_{m,n}\}$constitute an orthonormal basis for 31[2]. A particular case of interest is when the scale and shift parameters are critically sampled on a dyadic grid[2]. In this case if $a, = 2^{-1}$, and $b_0 = 1$, then (2.15) becomes

$$\psi_{m,n}(t) = 2^{m/2}\psi(2^m t - n) \tag{2.18}$$

The linear expansion in (2.4), which employs the above orthonormal basis functions is called the Discrete Wavelet Transform (DWT). DWT represents an octave-band decomposition of the signals in the TF-plane and possesses the greatest sparsity of representation among CWT, WFT, and DWT expansions for continuously indexed functions in $L_2(\boldsymbol{R})$. The reconstruction formula for DWT is given by

$$x(t) = \sum_{m,n \in J} \langle \psi_{m,n}, x \rangle \psi_{m,n} \tag{2.19}$$

---

[2] *Orthonormal DWT employing non-dyadic scaling can also be constructed.*

2.3 Wavelets and Multiresolution Analysis

Multiresolution Analysis of images by means of the multiresolution pyramid algorithm was first introduced in computer vision and image processing [29]. In the pyramid algorithm an image is decomposed into a coarse approximation plus a prediction error that is the difference between the original signal and a prediction based on the coarse approximation. A coarse version of the image at each level of decomposition is obtained by successive filtering and subsampling of the original signal with a low-pass filter. Although in this method the expansion is redundant[3], the pyramid algorithm is intimately related to subband and wavelet decompositions [2]. The theory of Multiresolution Analysis (MRA) based on the wavelet bases of $L_2(\boldsymbol{R})$ was developed in [30, ***31,*** 32], and the Discrete-time Multiresolution Analysis (DMRA) based on the wavelet bases of $l_2(\boldsymbol{Z})$ was studied in [33]. They are summarized in the following subsections.

2.3.1 Multiresolution Analysis and Wavelet Bases of $L_2(\boldsymbol{R})$

The MRA based on wavelet bases for continuous-time signals in $L_2(\boldsymbol{R})$ is defined as a sequence of closed and nested subspaces V_j of $L_2(\boldsymbol{R})$ [31, ***22,*** 26], with the following properties

1. $V_j \subset V_{j+1}, \forall j \in \boldsymbol{Z}$,
2. $x(t) \in V_j \Leftrightarrow x(2t) \in V_{j+1}$,
3. $x(t) \in V_0 \Leftrightarrow x(t+1) \in V_0$,
4. $\bigcup_{j \in Z} V_j$ is dense in $L_2(\boldsymbol{R})$, and $\bigcap_{j \in Z} V_j = \{0\}$,
5. there exists a ***scaling function*** $\phi \in V_0$, with a non-vanishing integral, such that the collection $\{\phi(t-k) | k \in Z\}$ is a Riesz basis of V_0.

Since $\phi \subset V_0 \subset V_1$, and from *(2)* and *(5)* one may conclude that there exists a sequence $\{h[n]\} \in l_2(\boldsymbol{Z})$, such that the scaling function satisfies a ***two-scale*** difference equation given by

$$\phi(t) = \sqrt{2} \sum_{n \in Z} h[n] \phi(2t-k) \tag{2.20}$$

This equation is also called the *refinement* or ***dilation*** equation. Usually no explicit

---

[3] ***The number of samples in the expansion is greater than the number of samples in the original image.***

expression for the scaling function ϕ is available, but fast algorithms exist that use the dilation equation to evaluate ϕ at dyadic intervals [2, 22, 25]. Since the dilates and translates of the scaling function are often used, they will be denoted by $\phi_{j,k} = 2^{j/2}\phi(2^{j}t - \mathrm{k})$. From the above properties it is immediate that the collection of functions $\{\phi_{j,k}\}_{k\in ,}$ is a Riesz basis of V_j.

Now, let W_j denote a space complementing V_j in V_{j+1}, since V, $\subset V_{j+,}$, then

$$V_{j+1} = V_j \oplus W_j \tag{2.21}$$

where $\oplus$ denotes a direct sum. Equation (2.21) implies that the space W, contains the detail information needed to go from an approximation at resolution j to an approximation at resolution j+1, as shown in Fig. 2.2.

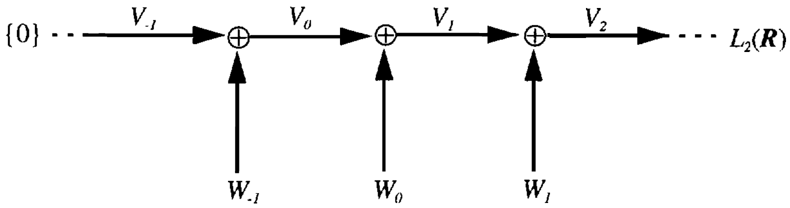


Fig. 2.2. Multiresolution analysis of $L_2(\boldsymbol{R})$.

Given the above properties, a function ψ is called a wavelet if the collection of functions $\{\psi(t-k)|k\in \boldsymbol{Z}\}$ is a Riesz basis of W_0. If the shifts and dilates of this; wavelet function are denoted as $\psi_{j,,} = 2^{j/2}\psi(2^{j}t - \mathrm{k})$, then the collection of wavelet functions $\{(\mathrm{I},,\}_{j,,\in ,}$ is a Riesz basis of $L_2(\boldsymbol{R})$, and there exists a sequence $\{g[n]\}\in l_2(\boldsymbol{Z})$ such that

$$\psi(t) = \sqrt{2}\sum_{n\in \boldsymbol{Z}} g[n]\phi(2t-k) \tag{2.22}$$

The above equation implies that the wavelet functions can be obtained from their corresponding scaling functions.

In general, there exists a dual multiresolution analysis associated with each MRA. For the dual MRA the approximation and detail spaces are denoted by $\tilde{V}_j$ and w,, respectively. The properties of the dual MRA are the same as the original MRA and its scaling and wavelet functions are given by $\tilde{\phi}_{j,k} = 2^{j/2}\tilde{\phi}(2^{j}t - \mathrm{k})$, and $\tilde{\psi}_{j,,} = 2^{j/2}\tilde{\psi}(2^{j}t - \mathrm{k})$, respectively. Then any function x in $L_2(\boldsymbol{R})$ can be decomposed into its projections in approximation and detail spaces as

$$x(t) = \sum_{k=-\infty}^{\infty} \langle x, \tilde{\phi}_{0,k} \rangle \phi_{0,k} + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} \langle x, \tilde{\psi}_{j,k} \rangle \psi_{j,k} \tag{2.23}$$

In the above equation, the projection of the function x onto W_j is referred to as "the details at level j", and generates the wavelet coefficients associated with $\psi_{j,k}$. It is important to note that the spaces W_j are not necessarily unique, because there are different ways to complement V_j in V_{j+1}. Therefore, it is possible to construct different classes of wavelet functions for multiresolution analysis of signals [2, 22, 25, 26].

The above analysis represents a biorthogonal MRA, and the wavelets obtained with this analysis are referred to as biorthogonal wavelets. It is possible to construct an orthonormal MRA by imposing some orthonormality constraint. A sufficient condition for an MRA to be orthogonal is $W_0 \perp V_0$ or equivalently $\langle \psi, \varphi(\,\cdot - k) \rangle = 0$, for all k in $\boldsymbol{Z}$. In this case, the wavelet spaces W_j are defined as the orthogonal complement of V_j in V_{j+1}. Moreover, the collection of functions $\{\phi(t-k) | k \in \boldsymbol{Z}\}$, and $\{\psi(t-k) | k \in Z)$, constitute orthonormal bases for spaces V_0 and W_0, respectively. Finally, in an orthonormal MRA the scaling and wavelet functions are the same as their duals.

2.3.2 Multiresolution Analysis and Wavelet Bases of $l_2(\boldsymbol{Z})$

An orthonormal MRA based on wavelet bases for discrete-time signals in $l_2(\boldsymbol{Z})$ is defined as a sequence of embedded closed spaces $V_{-J} \subset \ldots \subset V_{-2} \subset V_{-1} \subset V_0$, such that $V_0 = l_2(\boldsymbol{Z})$ [18]. In this case, we have

$$\bigcup_{j=-J}^{0} V_j = V_0 = l_2(\boldsymbol{Z}) \tag{2.24}$$

Let. W_j denote an orthogonal space complementing V_j in V_{j+1}, then $V_{j+1} = V_j \oplus W_j$. Assume that there exists a sequence $h[n]$ in V_0 such that $\{h[n-2k]\}_{k \in Z}$ is a basis for V_{-1}. Then, it can be shown that [33] there exists a sequence $g[n]$ in the approximation space V_0 such that $\{g[n-2k]\}_{k \in Z}$, is a basis for W_{-1}. This sequence is given by

$$g[n] = (-1)^n h[-n+1] \tag{2.25}$$

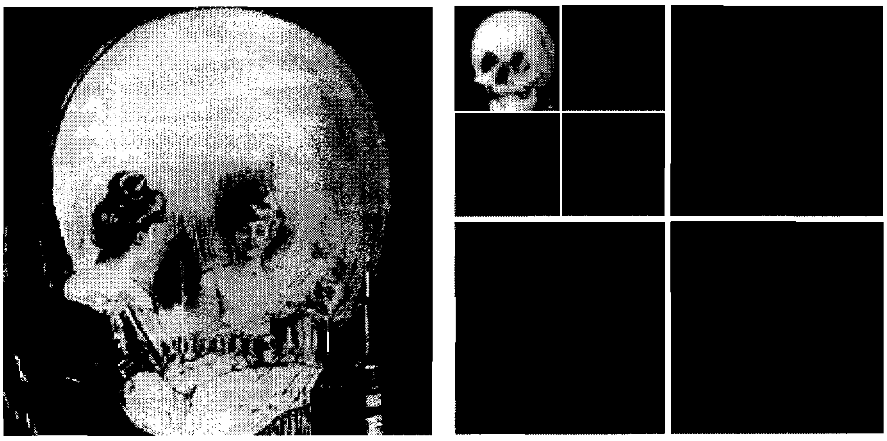
From the above discussion, one may conclude that $\{h[n-2k], \mathrm{g}[\mathrm{n}-2k]\}_{k}$,, is an orthonormal basis for V_0. Therefore, this decomposition can be iterated on V_{-1} for J times, and V_0 can be written as

$$V_0 = W_{-1} \oplus W_{-2} \oplus \ldots \oplus W_{-J} \oplus V_{-J} \tag{2.26}$$

This decomposition into approximation and detail spaces can be efficiently implemented by iterated perfect reconstruction filter banks [2,22, 25, 34].

In general, DMRA for discrete-time signals is very similar to MRA for continuous-time signals. However, there exist a few important distinctions. First, in DMRA only a finite number of decomposition levels are being considered. Therefore, there exists a coarsest resolution associated with V, and a *finest* resolution associated with V_0. Finally, in DMRA various iterated filters are used to perform the decomposition, whereas in MRA for continuous-time signals, a simple function and its scales and shifts are used to perform the reconstruction.

Although multiresolution techniques are of great theoretical value and have been successfully used in many signal and image processing applications, in some cases they may be suboptimal or even misleading. A counter-example to multiresolution analysis is shown in Fig. 2.3.



(a) (b)

Fig. 2.3. Counter-example to multiresolution analysis. (a) The original image (Comet Photo AG). (b) A two-level multiresolution decomposition of (a) with D4 orthogonal wavelets [22]. The coarse approximation shown in top-left comer of (b) is unrelated to the full-resolution image

2.4 Wavelets, Multiresolution Analysis, and Filter Banks

Recent work on the construction of structured wavelet bases for the linear expansion of signals has shown that this problem is in many ways analogous to the design of multirate filter banks in signal processing applications [2, 22, 25, 34]. In fact, given a discrete-time sequence of length N, the Discrete-Time Wavelet Transform (DTWT) of this sequence can be efficiently implemented with multirate filter banks in $O(N)$ [2]. This relation is evident from equations (2.20) and (2.22), which basically represent filtering operations followed by sub-sampling. In these equations, the sequences $h[n]$ and $g[n]$ can be considered as impulse responses of the high-pass and low-pass filters in the synthesis portion of a iterated filter bank as shown in Fig. 2.4.

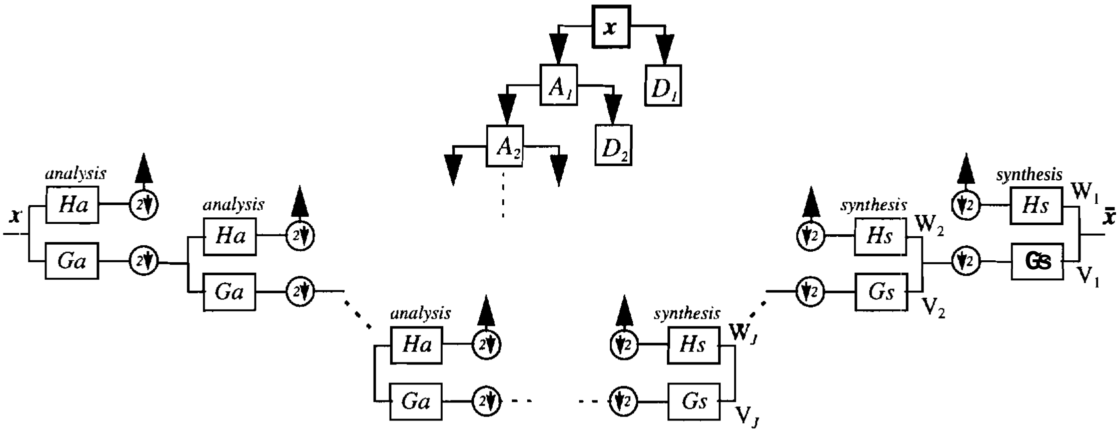
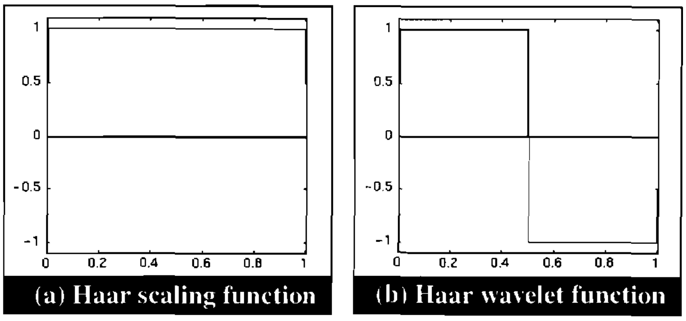


Fig. 2.4. Iterated two channel filter bank and decomposition tree. (bottom)The iterated filter bank, H and G represent the high-pass and low-pass filters, respectively. (top) The decomposition tree corresponding to the analysis banks. The symbols A_J and D_J represent the coarse approximation and detail decompositions at level J, respectively.

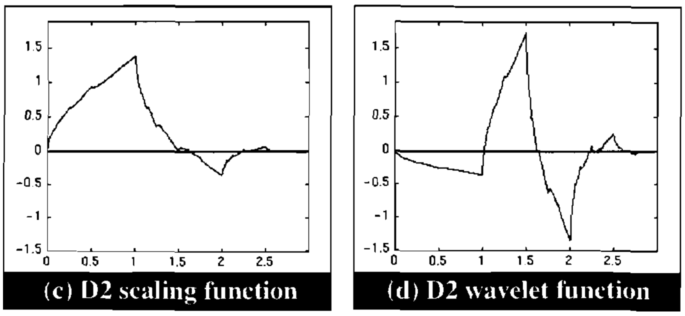
From Fig. (2.4) one can conclude that a frequency analysis can be obtained by iterating a two-channel filter bank on the previous low-pass channel. The implementation of multirate filter banks with iterated two-channel filter banks is often referred to as a constant-Q filter bank algorithm [2]. If the above filter banks have the perfect reconstruction property (i.e. if the transfer function of the entire system is unity), this system exactly resembles the DMRA of section 2.3.2. The corresponding approximation and detail spaces for DMRA are labeled in the synthesis filter banks in Fig. (2.4). The conditions for perfect reconstruction filter banks, Quadrature Mirror Filters (QMF), and their relation to wavelets is given in [2]. Another important link between the filter bank and wavelet theory is the fact that continuous wavelet bases can be obtained by iterated filter banks [2, 22].

In signal and image processing applications, the greatest interest is in construction of structured bases. The wavelet bases more frequently used for image compression include orthogonal wavelets, biorthogonal wavelets, and wavelet packets. The compactly supported orthonormal wavelets can be efficiently implemented with FIR filter banks [25, 34]. The oldest example of this type is the Haar wavelet. This wavelet has good time resolution, but its frequency resolution is poor. In addition, it has poor **regularity** properties. To overcome these problems, Daubachies [22] proposed an algorithm for designing orthonormal and compactly supported wavelets that are more regular, and can be efficiently implemented with filter banks. These filters are called Daubachies (Dn) filters. The Haar wavelet (or equivalently D1 wavelet), D2 wavelet, and their corresponding scaling functions are shown in Fig. 2.5.



(a) Haar scaling function

(b) Haar wavelet function



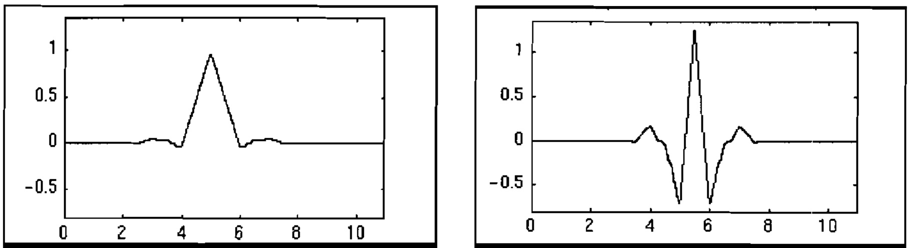
(c) D2 scaling function

(d) D2 wavelet function

Fig. 2.5. Examples of compactly supported orthogonal wavelet. (a) The Haar scaling function. (b) The Haar wavelet function. (c) The D2 scaling function, (d) The D2 wavelet function.

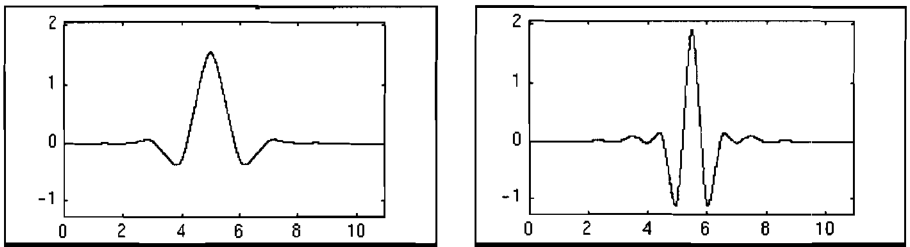
A major disadvantage of the compactly supported orthonormal wavelets is their asymmetry. In image processing applications it is more desirable to have symmetric wavelets bases. A lack of symmetry results in nonlinear phase FIR filter banks, and consequently causes reconstruction artifacts at the boundaries of the objects.

To obtain both compact support and symmetry, one may use biorthogonal wavelets [2]. An example of a biorthogonal wavelet based on spline functions is shown in Fig. 2.6.



(a) analysis scaling function

(b) analysis bi-orthogonal wavelet



(c) synthesis scaling function

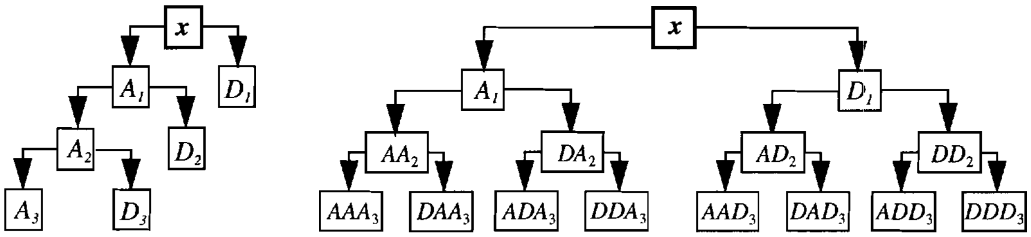
(d) synthesis bi-orthogonal wavelet

Fig. 2.6. Example for bi-orthogonal wavelets. (a) The analysis scaling function, (b) The analysis wavelet function, (c) The synthesis scaling function, (d) The synthesis wavelet function.

Shift-orthogonal wavelets constitute another class of symmetric and compactly supported wavelets that are suitable for image compression [35]. In this class the wavelets are orthogonal to their translates within the same scale but not across scales. They allow the design of shorter wavelet synthesis filters while preserving the orthogonality within the independent wavelet channels. These features can be used for efficient quantization and coding of individual channels [36]. Furthermore, because of orthogonality with respect to shifts this class of wavelets can be implemented with standard tree-structured perfect reconstruction filter banks.

2.5 Wavelet Packets

Wavelet Packets (WP) were introduced in [24] as a family of orthonormal bases for discrete functions of R^N, and include the wavelet basis and STFT basis as their members. In fact, wavelet packets represent a generalization of the multiresolution decomposition of signals into wavelet bases that encompass the entire family of subband tree decompositions. In contrast to wavelet analysis, in WP analysis the details as well as the approximations are decomposed, as illustrated in Fig. 2.7.



(a) (b)

Fig. 2.7. Decomposition tree for a signal. The symbols A, and Dj represent the coarse approximation and detail decompositions at level J, respectively. (a) Wavelet tree decomposition. (b) Wavelet packet tree decomposition.

While for a J-level wavelet decomposition there are J+1 possible ways to decompose a signal, for a J-level wavelet packet decomposition there are 2^J possible different ways to decompose a signal. This gives a rich menu of basis functions that can be used to represent independent segments of a non-stationary signal. Choosing one out of all possible decompositions presents an interesting problem that will be discussed further in subsequent chapters.

2.6 Concluding Remarks

Wavelet theory offers a set of new tools for signal analysis and expansion. The time-frequency characteristics of wavelets, and their relation to multirate filter banks is well suited to many image processing applications. An interesting case in wavelet based expansion is when an adapted basis is used for analysis. In this case the transform depends on the signal and it is possible to find ***best bases*** that will allow sparse and high resolution expansion of signals. These types of bases are most desirable in image and video compression applications, because they result in expansions that retain the most energy in the fewest possible number of coefficients. In addition, no prior model for the data is assumed in adaptive expansions. The adaptive expansion of signals based on best basis selection is the subject of the next chapter.

3. OPTIMAL ADAPTIVE SIGNAL EXPANSION WITH SEGMENTED ORTHOGONAL MATCHING PURSUITS

3.1 Introduction

In recent years, transform based techniques using a fixed set of Fourier or wavelet basis functions have been extensively used for image representation and lossy image compression [1, 2]. In these techniques, the signal is first expanded into a linear combination of a fixed number of basis functions called ***atoms.*** If the transform captures the underlaying sparse structure of the signal, then most of the energy of the signal is retained in a few terms of the expansion. This is often referred to as ***energy compaction*** of the transform in compression literature. This energy compaction property of the transform can then be exploited with various quantization and coding technique:; to achieve lossy compression at low bit rates [2].

In this work the main concern is with the representation and compression of natural images. Natural images are two dimensional signals with unknown or time-varying characteristics. For this type of signal, linear expansion with a fixed set of basis functions is not flexible enough to represent the data with the desired degree of sparseness. For example, the Fourier transform is not a good fit for regions with sharp discontinuities such as edges, and the wavelet transform is not a good fit for regions with periodic high-frequency components such as localized textures or stripes [3]. In general, the Fourier basis provides a poor representation of signals well localized in time: (space), and the wavelet bases are not well adapted to represent signals whose Fourier transforms have a narrow high frequency support. As stated in [37], linear decompositions in a fixed basis set are analogous to a text written with a small vocabulary. Although this vocabulary may be sufficient to express general ideas, it requires the use of circumvolution, or replacing available words by full sentences. Therefore, it is possible to improve the energy compaction of a transform by enlarging the number of atoms beyond a basis. This enlarged and redundant set of atoms is called an *overcomplete dictionary* [37]. In overcomplete dictionaries, some of the atoms can be represented by linear combinations of others. Consequently, the expansion of signals in these overcomplete dictionaries is not unique. However, this non-uniqueness can be exploited by using efficient adaptive algorithms to achieve signal expansions that are more sparse than the traditional fixed transforms.

In recent years, there has been growing interest in the representation of signals with overcomplete dictionaries of atoms [2]. In this type of signal expansion the following issues should be addressed:

- Which atoms should be included in a dictionary.
- Given a dictionary, how to select the *best* atoms to represent the signal.

Answers to the first question have resulted in the introduction of various dictionaries with different characteristics, and answers to the second question have produced a number of adaptive and non-adaptive algorithms for signal expansion.

In this chapter a new adaptive algorithm for signal expansion is introduced. This algorithm often results in expansions which are sparse and high resolution. An algorithm for the construction of optimal and near optimal hybrid dictionaries is also presented. The organization of this chapter is as follows. Section 3.2 provides an overview of the most widely used dictionaries and a brief description of their characteristics. Although this list is not exhaustive, the dictionaries relevant to this work have been included. Section 3.3 presents an overview of existing adaptive and non-adaptive signal expansion algorithms. Section 3.4 is devoted to the description of the new adaptive algorithm. Finally, experimental results and concluding remarks are provided in section 3.5.

3.2 Classification of Dictionaries

A dictionary $\mathcal{D}$, is **a** collection of parametrized atoms given by $\mathcal{D} = \{\phi_\gamma : \gamma \in \Gamma\}$. In this notation, the ϕ_γ's are discrete-time vectors (atoms) of length N, and the parameter γ is an index. If the number of atoms in $\mathcal{D}$ is greater than N, then the dictionary is overcomplete, and if this number is equal to N, the dictionary is complete. An overview of the most widely dictionaries used in signal and image processing are given the following subsections.

3.2.1 Dirac Dictionary

The Dirac dictionary is the collection of Dirac delta functions, which are zero except at one point. Given a discrete-time signal of length N, this dictionary can be indexed by $\gamma = \{0, 1, \ldots, N-1\}$, and the atoms are given by

$$\varphi_{\gamma}[n] = \mathrm{I}_{[n = \gamma]} \tag{3.1}$$

where $\mathrm{I}_{[n = \gamma]}$ is the usual Dirac delta function, which is equal to one if $n = \gamma$, and zero otherwise. From (3.1) it is clear that $\mathcal{D}$ is a collection of *standard basis* functions, and therefore it constitutes an orthonormal basis for R^{N}. The atoms in this dictionary are well localized in time, but their frequency localization is very poor.

3.2.2 Heaviside Dictionary

The Heaviside dictionary is a collection of step functions. Given a discrete-time signal of length N, the atoms in this dictionary are indexed by $\gamma = \{0, 1, \ldots, N-1\}$, and are given by

$$\varphi_{\gamma}[n] = \mathrm{I}_{[n \geq \gamma]} \tag{3.2}$$

The time localization of the atoms of this dictionary is not as good as the Dirac dictionary, but the atoms have better frequency localization properties. Although in this case $\mathcal{D}$ is complete, it is easy to show that the atoms in this dictionary are not orthogonal.

3.2.3 Frequency or Fourier Dictionaries

This type of dictionary is simply a collection of sinusoidal waveforms. The atoms in this set can be indexed by an angular frequency $\omega \in [0, 2\pi)$, and a phase type indicator $\rho \in \{0, 1\}$, where the indices 0 and 1 correspond to sine and cosine functions, respectively. Therefore, the atoms in a Fourier dictionary can be indexed by $\gamma = (\omega_k, \rho)$, and are given by

$$\varphi_{\omega_k, 0}[n] = \sin(\omega_k n) \tag{3.3}$$

and

$$\varphi_{\omega_k, 1}[n] = \cos(\omega_k n) \tag{3.4}$$

In the standard Fourier dictionary the cosines are sampled at $\omega_k = 2\pi k/N$, for $k = 0, 1, \ldots, N/2$, and sines are sampled at $\omega_k = 2\pi k/N$, for $k = 1, \ldots, N/2 - 1$. This dictionary consists of N waveforms that are mutually orthogonal, and therefore form a basis for $l_2(\mathbf{Z})$. It is also possible to obtain overcomplete Fourier dictionaries by oversampling the angular frequency. Other frequently used dictionaries in this class include the sine and cosine dictionaries. This class suffers from poor frequency localization properties.

3.2.4 Time-Scale Dictionaries

A time-scale dictionary is a collection of the dilations and translations of a function, called the mother wavelet, together with the dilations and translations of a function called the scaling function. In this case the atoms are indexed by $\gamma = (a, b, p)$, where $a \in (0, \infty)$, $b \in [0, N]$, and $p \in \{0, 1\}$. Here, the values 0 and 1 for p correspond to the scaling function and the mother wavelet, respectively. The atoms in this dictionary are given by

$$\varphi_{a, b, 0}[n] = a^{1/2}\phi(a(n-b)) \tag{3.5}$$

and

$$\varphi_{a, b, 1}[n] = a^{1/2}\psi(a(n-b)) \tag{3.6}$$

For the applications of this thesis, discrete-time atoms defined on a dyadic grid are of main interest. In this case the dyadic scales are given by $a_J = 2^J/N$, for $J = 1, \ldots, \log_2(N)-1$, and the translations are specified by integer multiples of the scale as $b_{J,k} = k \cdot a_J$, for $k = 0, \ldots, 2^J - 1$.

The oldest member of this family is the Haar dictionary, in which the mother wavelet is defined as $\psi = I_{[0, 1/2]} - I_{(1/2, 1]}$, and the scaling function is given by $\phi = I_{[0, 1]}$. The wavelet forms an orthonormal basis for $l_2(\mathbf{Z})$, and has good time localization properties, but poor frequency localization and regularity properties [22]. For this type of dictionary a wide variety of wavelets with various time localization, frequency localization, and regularity properties can be derived from the two-scale equation. The most important variations include the compactly supported and orthogonal Daubechies wavelets, and Daubechies near symmetric wavelets [22], biorthogonal wavelets [2], shift-orthogonal wavelets [35, 36], and spline wavelets [26], and multiwavelets [23]. For 2-D signals both separable and non-separable wavelets are possible [2, 22]. Separable 2-D wavelets are obtained from the tensor product of 1-D wavelets, and because of this directional preference a wider range of separable wavelet dictionaries is possible.

3.2.5 Time-Frequency Dictionaries

A time-frequency dictionary is a collection of modulated functions indexed by angular frequency, time location, phase, and duration. In this case the atoms in the dictionary are indexed by $\gamma = (\omega_k, \tau_k, \mathrm{An}, \theta)$, and the atoms are given by

$$\varphi_\gamma[n] = \exp\{-(n-\tau_k)^2/(\Delta n)^2\} \cdot \cos(\omega_k(n-\tau_k)+\theta) \qquad (3.7)$$

where $0 \in \{0, \pi/2\}$, and for a fixed An, complete and overcomplete sets of dictionaries are obtained by different choices of sampling frequencies for ω_k and τ_k [38]. The oldest members of this class of dictionaries are the STFT or Gabor dictionaries [19]. These dictionaries are complete, and have better time and frequency localization properties than the Fourier dictionaries. However, their time and frequency localization properties are inferior to those of time-scale dictionaries. More recently a number of overcomplete time-frequency dictionaries have been proposed that have better localization properties than Gabor dictionaries. These are wavelet packet (WP) and cosine packet (CP) dictionaries [2, 24]. These dictionaries offer a rich menu of atoms for signal expansion. In fact, the orthonormal wavelet dictionaries are special cases of the wavelet packet dictionary, and the standard Fourier and Gabor dictionaries are special cases of the cosine packet dictionary.

3.3 Optimal and Sub-optimal Signal Expansion

Let x $= \{x[n]; n = 0, 1, \ldots, N-1\}$, be a discrete-time signal in a finite dimensional Hilbert space $\mathcal{H} = R^N$, with the inner product of x, y $\in \mathcal{H}$ defined as

$$\langle x, y\rangle = \sum_n x[n] \, . \, y[n] \tag{3.8}$$

and its norm as $\|x\| = \langle x, x\rangle^{1/2}$. Given a dictionary of basis functions $\mathcal{D} = \{\phi_\gamma : \gamma \in \Gamma\}$ in $\mathcal{H}$, with $\|\phi_\gamma\| = 1$ and $span(\mathcal{D}) = \mathcal{H}$, the goal is to obtain an exact representation of **x** with linear combinations of a small number of atoms in $\mathcal{D}$ such that

$$x = \sum_{\gamma \in \Gamma} a_\gamma \varphi_\gamma \tag{3.9}$$

or an approximate decomposition as

$$x = \sum_{\gamma \in \Gamma} a_\gamma \varphi_\gamma + r \tag{3.10}$$

where $\{\varphi_\gamma\}_{\gamma \in \Gamma}$ represents the set of atoms in $\mathcal{D}$, the a_γ's are the coefficients of expansion ($a_\gamma \in C$, where C denotes the set of complex numbers), and r is the residual.

Alternatively, If all atoms in the dictionary are written out as columns of a matrix Φ, and all the coefficients as a column vector a, then the decomposition in (3.9) can be written as a linear equation given by

$$\Phi a = x \tag{3.11}$$

In this notation, the problem reduces to that of finding an exact or approximate solution for the above linear equation.

In signal representation and compression applications, the goal is to expand a signal over $\mathcal{D}$, with a small number of atoms retaining most of the energy of the signal. These expansions will be also referred as approximations. An optimal approximation with L atoms can be defined as follows.

Definition 3.1 Let $\mathcal{D} = \{\phi_\gamma : \gamma \in \Gamma\}$, be a dictionary of atoms in *an* N-dimensional *Hilbert* space $\mathcal{H}$. Let $\hat{x}$ denote the approximation of x $\in \mathcal{H}$, by L atoms in $\mathcal{D}$ (L $\leq$ N), *such* that

$$\hat{x} = \sum_{\gamma=0}^{L-1} a_\gamma \varphi_\gamma \tag{3.12}$$

Then, an L-optimal approximation is an expansion that minimizes $\varepsilon_L = \|\hat{x} - x\|$.

When $\mathcal{D}$ is complete and the atoms in $\mathcal{D}$ are orthogonal, it constitutes an orthonormal basis for $\mathcal{H}$, and the error ε_L is minimized by simply selecting L atoms that have the largest inner products with x. Therefore, an L-optimal approximation for x can be found in $O(N^2)$ operations. For certain bases and spaces, it is also possible to estimate the rate of decay for ε_L, as L increases. For example, when atoms in $\mathcal{D}$ are wavelet bases, the rate of the decay of ε_L for functions that belong to certain Besov spaces can be estimated [39]. Furthermore, it is shown that this wavelet representation is asymptotically near optimal in the sense that the rate of decay for ε_L is equal to the largest decay attainable by a general class of nonlinear transform-based approximation schemes [39].

When $\mathcal{D}$ is overcomplete, the error ε_L can be minimized by simply selecting the L atoms that have the largest inner products with x. In fact, the problem of finding an L-optimal approximation with general overcomplete dictionaries is *NP-hard* due to the following theorem [40].

Theorem 3.1 *Let* $\mathcal{D}$ *be the set of all dictionaries for an N-dimensional Hilbert space* $\mathcal{H}$ *that contains* $O(N^k)$ *atoms, where* $k \geq 1$. *Then finding the L-optimal approximation problem is NP-hard.*

The above theorem implies that the problem of finding an L-optimal approximation for general dictionaries is an NP-hard problem. However, for specific dictionaries such as the orthonormal wavelet dictionary mentioned above, this problem can be solved in polynomial time.

Because of the difficulty of obtaining an L-optimal approximation, one can resort to near optimal or suboptimal expansions over a specific dictionary of atoms. In recent years, a variety of algorithms have been proposed to find near optimal expansions in different applications. Some of these algorithms are explained in the following subsections.

3.3.1 The Method Of Frames

The Method of Frames (MOF) uses an optimization procedure that adaptively refines the signal approximation over a redundant dictionary of basis functions [22, 38]. Given $\Phi a = x$, and an overcomplete dictionary $\mathcal{D}$ whose elements belong to a finite dimensional Hilbert space $\mathcal{H} = R^N$, MOF tries to find a representation of the signal whose coefficients have minimal l_2 norm by solving the following optimization problem

$$min\|a\|_2 \text{ subject to } \Phi a = x \tag{3.13}$$

The solution to the above problem is often called a minimum-length solution, because the collection of all the solutions to (3.11) form an affine subspace of R^N, and MOF finds the element of this subspace closest to the origin [41]. The minimum-length solution to this problem is found by the generalized inverse of Φ as

$$a_{opt} = \Phi^t(\Phi\Phi^t)^{-1}x \tag{3.14}$$

The numerical value of a_{opt} is often obtained with conjugate gradient iterative algorithms, which are faster than the generalized solutions using singular value decomposition [41, 42]. For the wavelet packet dictionary, the frame is tight, and the solution to MOF is given in closed form by

$$a_{opt} = (\log_2 N)^{-1}\Phi^t x \tag{3.15}$$

In this case Φ^t is the analysis operator of the wavelet packet dictionary and can be computed with $N(\log_2 N)$ operations. The major disadvantage of MOF is that it often does not exploit the sparse structure of signals [43]. Moreover, the resolution of the MOF expansion is limited and can not resolve those features of the signal that are sharply localized in time [43].

3.3.2 The Method Of Best Orthogonal Basis

The method of Best Orthogonal Basis (BOB) is a fast algorithm that adaptively selects a single orthogonal basis that is the best basis in the wavelet packet dictionary, based on an entropy measure [44]. Let $x[\mathcal{B}]_l$ represent the vector of coefficients corresponding to the expansion of x over an orthogonal basis set $\mathcal{B}$, $e(\bullet)$ be a scalar function with a scalar argument, and let $\mathcal{E}(x[\mathcal{B}]) = \sum_l e(x[\mathcal{B}]_l)$ define the entropy of $x[\mathcal{B}]$. Then BOB finds the best orthogonal basis in order of $N(\log N)$ operations, by solving the following optimization problem

$$min\{\mathcal{E}(\boldsymbol{x}[\mathcal{B}]):\ (\mathcal{B} \subset \mathrm{D})\} \tag{3.16}$$

When a signal can be sparsely expanded over an orthogonal basis in $\mathcal{D}$, this method works well and produces near optimal expansions for a signal in terms of sparsity, but fails to deliver a sparse representation when the signal is a superposition of a moderate number of highly non-orthogonal components [43].

3.3.3 The Single Tree Algorithm

The Single Tree Algorithm (STA) searches a wavelet packet dictionary for the best bases to represent the signal, by using a cost function and a fast search algorithm [45]. The goal of the STA is to find a representation that is more suitable for signal compression in a Rate-Distortion (R-D) sense. STA uses the Lagrangian cost function $\mathrm{J} = \mathrm{D} + \lambda R$, which trades off rate for distortion at a quality factor given by the Lagrange multiplier $\lambda \geq 0$. In STA the parameter λ represents the absolute slope of the R-D curve, and the optimal slope for a specific coding goal should be matched to the target rate R. Due to convexity of the *R-D* curve, the optimal value of λ can be obtained with a fast search algorithm, as follows. Assume that the R-D cost metrics are additive over the entire WP tree. This assumption can be written as $R(tree) = \sum R(leaf\ \mathrm{nodes})$, and $D(tree) = \sum D(leaf\ \mathrm{nodes})$. Then grow a full tree for the entire signal up to a maximum fixed depth, and populate each WP tree node with the minimum Lagrangian cost over all quantization choices Q, for that node. Then the minimum cost at each node can be found from

$$\mathrm{J}(node) = \min_{Q}\ [D(node) + \lambda R(node)] \tag{3.17}$$

then starting from the leaf nodes, the full WP is recursively pruned subject to an optimal dynamic programing split-and-merge decision. In this algorithm the children nodes are pruned if $\mathrm{J}(parent\ \mathrm{node}) \leq [\mathrm{J}(child1) + \mathrm{J}(child2)]$. For a discrete-time signal of size N, the computational complexity of this algorithm is $N(\log N)$.

For 2-D images, it can be shown that the number of atoms $A(d)$, in a single tree of depth d, is given by the recursion $A(d) = [A(d-1)]^4 + 1$, with $A(1) = 2$. For example in a depth five 2-D WP decomposition, the STA algorithm should search a wavelet packet dictionary of 5.60×10^{78} atoms.

The STA algorithm produces an R-D optimal expansion for stationary signals. However if the signal is non-stationary or exhibit time-varying characteristics, this algorithm can not locally adapt to different segments of the signal.

3.3.4 The Double Tree Algorithm

The Double Tree Algorithm (DTA) is simply an extension of the STA from frequency decomposition to the joint time-frequency decomposition of signals, as shown in Fig. 3.1 [46].

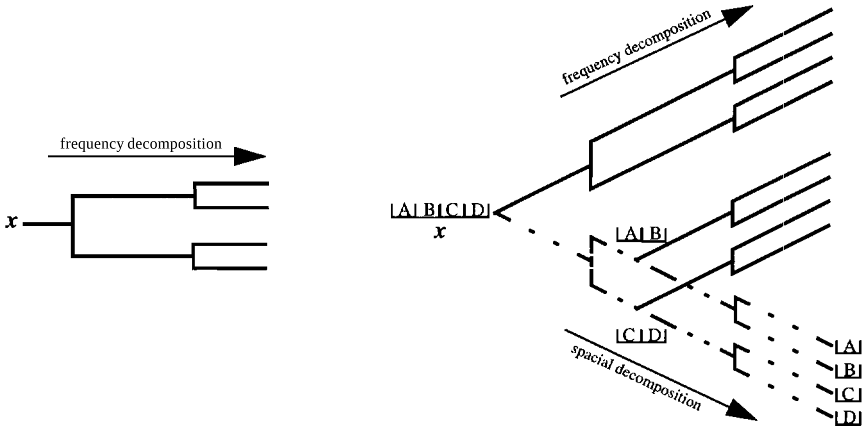


Fig. 3.1. Single and double tree of depth two for a 1-D signal. The single tree algorithm (left) uses a fixed dictionary of wavelet functions and performs a static frequency decomposition on the signal. The double tree algorithm (right) performs a spatial decomposition. in addition to frequency decomposition. Solid lines present the frequency tree and dotted lines present the spatial tree.

In DTA a single tree is first grown on the entire signal and stores the associated cost at each node by using the single tree algorithm. It then calculates the single tree for the first and second halves of the signal and stores the costs at each node and continues the decomposition by bisecting the signal in a similar manner until the desired depth is reached. The double tree is then pruned in a manner similar to STA.

For a discrete-time signal of size N, the computational complexity of this algorithm is $N(\log N)^2$. For 2-D images, it can be shown that the number of atoms $D(d)$, in a double tree of depth d, is given by the recursion $D(d) = [D(d-1)]^4 + A(d) - A(d-1) + 1$, with $D(1) = 2$, where $A(d)$ is the number of bases searched by a single tree of depth d. For example, in a depth five 2-D double tree WP decomposition, the algorithm should search a wavelet packet dictionary of 6.50 x 10^{96} atoms. In this algorithm, coarse segmentation of the signal in this algorithm may result in boundary artifacts in the reconstructed signal. This problem can be solved in a number of ways as explained in [47, 48].

3.3.5 The Method Of Basis Pursuit

The method of Basis Pursuit (BP) uses a convex optimization procedure that adaptively refines the signal approximation over a redundant dictionary of basis functions [43]. Given $\boldsymbol{\Phi a} = \boldsymbol{x}$, and an overcomplete dictionary $\mathcal{D}$, whose elements belong to a finite dimensional Hilbert space $\mathcal{H} = R^N$, the method of BP tries to find the representation of the signal whose coefficients have minimal l_1 norm. That is, one has to solve an optimization problem of the form

$$min\|\boldsymbol{a}\|_1 \text{ subject to } \boldsymbol{\Phi a} = \boldsymbol{x} \qquad (3.18)$$

From one point of view, the method of BP is very similar to the method of frames, because it simply replaces the l_2 norm in the method of frames with the l_1 norm. However, this minor change has a major impact on the outcome of this optimization problem [43]. While the method of frames solves a quadratic optimization problem, BP should solve a convex and non-quadratic optimization problem. Although the method of BP involves nonlinear optimization, it is possible to reformulate the equation (3.17) into a linear optimization problem with the method of slack variables [49]. Moreover, for a signal at a noise level of $\sigma > 0$, it is possible to obtain an approximate solution as in (3.10) by solving the following optimization problem

$$min\|\boldsymbol{\Phi a} - \boldsymbol{x}\|_2^2 + \lambda_n\|\boldsymbol{a}\|_1 \qquad (3.19)$$

where $\lambda_n = \sigma\sqrt{2log(card(\mathcal{D}))}$, and $card(\mathcal{D})$ denotes the number of distinct basis functions in the dictionary. Based on the theory of linear programing, it is possible for the linearized BP to converge to a global optimum. However if the signal is non-stationary, this algorithm will choose a set of atoms that is best on average for the whole signal, but can. not effectively exploit the local sparse structure of the signal.

All of the above optimization techniques start with a general model for the signal and perform a global optimization to select a subset of atoms from the dictionary which best represent the signal according to an optimization criterion. In other words, they adopt a bottom up approach by expanding the signal over all the atoms in the dictionary and proceed by pruning the representation into an approximation which is optimal for the entire signal. However if the signal is non-stationary or exhibit time-varying (space-varying) characteristics, most of these algorithms[1] can not locally adapt to different segments of the signal and produce expansions which are suboptimal for the entire signal in terms of sparsity.

In signal compression applications, it may be more advantageous to use a top-down greedy algorithm that starts with a single atom that ***best*** approximate the signal according to a given measure of optimality and proceeds by refining the approximation by selecting more atoms from the dictionary until the desired bit budget or PSNR is achieved. A top-down algorithm that is suitable for such a greedy approximation is the method of matching pursuit.

3.3.6 The Method of Matching Pursuit

The method of Matching Pursuit (MP) [37] uses a greedy algorithm that adaptively refines the signal approximation with an iterative procedure. This algorithm is basically a special case of the Projection 'Pursuit (PP) algorithm in statistical parameter estimation theory [50, 51]. Let $\mathcal{D} = \{\varphi_\gamma : \gamma \in \Gamma\}$ be a dictionary of unit vectors in $\mathcal{H} = R^N$, and $x \in \mathcal{H}$ be the input signal. The MP begins by searching $\mathcal{D}$ for some φ_{γ_0}, and projecting x onto this atom as

$$x = \langle x, \varphi_{\gamma_0} \rangle \varphi_{\gamma_0} + r^{(1)} x \tag{3.20}$$

---

[1] *The DTA algorithm is the only bottom-up algorithm that can adapt to local structure of the signal.*

where the superscript (•) denotes the number of iterations. Clearly, the residual $r^{(1)}x$ is orthogonal to φ_{γ_0}. This implies that

$$\|x\|^2 = |\langle x, \varphi_{\gamma_0}\rangle|^2 + \|r^{(1)}x\|^2 \tag{3.21}$$

In order to minimize the norm of the residual in (3.21), the atom $\varphi_{\gamma_0} \in \mathcal{D}$ has to be chosen such that $|(x, \varphi_{\gamma_0}\rangle|$ is maximized. Therefore φ_{γ_0} should be selected such that

$$|\langle x, \varphi_{\gamma_0}\rangle| \geq \alpha \sup_{\gamma \in \Gamma} |\langle x, \varphi_\gamma\rangle| \tag{3.22}$$

where the parameter $0 < \alpha \leq 1$, is an optimality factor. For finite dimensional spaces (3.22) reduces to $|\langle x, \varphi_{\gamma_0}\rangle| \geq \alpha|\langle x, \varphi_\gamma\rangle|$, for $\gamma_0 \neq y$, and α is typically close or equal to one. The algorithm then chooses the next basis in $\mathcal{D}$ to match $r^{(1)}x$, and proceeds iteratively on the residues until some measure of error or convergence criterion such as l_2 norm is met. For example, the algorithm could be terminated at iteration p if $\|r^{(p)}x\| < \varepsilon\|x\|$ for some $\varepsilon > 0$.

The MP algorithm can be summarized as follows. Let $r^{(0)}x = x$, assuming $r^{(k)}x$ have been already computed, then φ_{γ_k} can be chosen from $\mathcal{D}$, such that

$$|\langle r^{(k)}x, \varphi_{\gamma_k}\rangle| \geq \alpha \sup_{\gamma \in \Gamma} |\langle r^{(k)}x, \varphi_\gamma\rangle| \tag{3.23}$$

then proceeding by projecting $r^{(k)}x$ onto φ_{γ_k} as

$$r^{(k+1)}x = r^{(k)}x - \langle r^{(k)}x, \varphi_{\gamma_k}\rangle\varphi_{\gamma_k} \tag{3.24}$$

since $r^{(k+1)}x$ is orthogonal to φ_{γ_k}

$$\|r^{(k+1)}x\|^2 = \|r^{(k)}x\|^2 - |\langle r^{(k)}x, \varphi_{\gamma_k}\rangle|^2 \tag{3.25}$$

Then, after p iterations of the algorithm

$$x = \sum_{k=0}^{p-1} \langle r^{(k)}x, \varphi_{\gamma_k}\rangle\varphi_{\gamma_k} + r^{(p)}x \tag{3.26}$$

The residual $r^{(p)}x$ in (3.26) is the approximation error, and its energy is given by

$$\|r^{(p)}x\|^2 = \|x\|^2 - \sum_{k=0}^{p-1} |\langle r^{(k)}x, \varphi_{\gamma_k}\rangle|^2 \tag{3.27}$$

The convergence of the error to zero in infinite dimensions can be proven as a consequence of the proof for the convergence of projection pursuit regression [52], and is given in [37]. The convergence of error can be extremely slow in infinite dimensional spaces. However, the convergence in finite dimensional spaces is exponential. For finite dimensional spaces, the rate of decay of $\|r^{(p)}\boldsymbol{x}\|$ depends on the correlation between the residues and elements of $\mathcal{D}$. This correlation can be quantified by using the correlation ratio as defined below.

Definition 3.2 *Let $\mathcal{D} = \{\phi_\gamma : \gamma \in \Gamma\}$ be a dictionary in a finite dimensional Hilbert space $\mathcal{H}$. Then for any $\boldsymbol{x} \in \mathcal{H}$, and $\varphi_\gamma \in \mathcal{D}$, the correlation ratio of $\boldsymbol{x}$ with respect to $\mathcal{D}$ is given by*

$$\rho(\boldsymbol{x}) = \sup_{\gamma \in \Gamma} (\langle \boldsymbol{x}, \varphi_\gamma \rangle / \|\boldsymbol{x}\|) \qquad (3.28)$$

The correlation ratio can be used to prove that in finite dimensional spaces the rate of convergence for MP is exponential, as stated in the following theorem [37, 40].

Theorem 3.2 *Let $\mathcal{D} = \{\phi_\gamma : \gamma \in \Gamma\}$ be a dictionary in a finite dimensional Hilbert space $\mathcal{H}$. Then for any $\boldsymbol{x} \in \mathcal{H}$, the energy of the residual $\|r^{(p)}\boldsymbol{x}\|^2$ in the MP algorithm has an exponential decay.*

Proof: Let φ_{γ_p} be the atom selected by the MP algorithm at iteration $\boldsymbol{p}$, and let $\boldsymbol{a} = 1$. Then the correlation ratio for $r^{(p)}\boldsymbol{x}$ is given by

$$\rho(r^{(p)}\boldsymbol{x}) = \left|\langle r^{(p)}\boldsymbol{x}, \varphi_{\gamma_p} \rangle\right| / \|r^{(p)}\boldsymbol{x}\|$$

and by (3.25), the energy of the residual error in iteration $\boldsymbol{p+1}$ is given by

$$\|r^{(p+1)}\boldsymbol{x}\|^2 = \|r^{(p)}\boldsymbol{x}\|^2 - \rho^2(r^{(p)}\boldsymbol{x})\|r^{(p)}\boldsymbol{x}\|^2 = \|r^{(p)}\boldsymbol{x}\|^2(1 - \rho^2(r^{(p)}\boldsymbol{x}))$$

Since there exists at least one basis for $\mathcal{H}$ in $\mathcal{D}$, and the unit sphere of $\mathcal{H}$ is compact in finite dimensions, it can be concluded that there exists a $\lambda_{min} > 0$, such that $\rho(r^{(p)}\boldsymbol{x}) \geq \lambda_{min}$ [37]. Therefore, from the above equation it can be concluded that the energy of the residual decays exponentially with a minimum rate of $-\frac{1}{2}\log(1 - \lambda_{min}^2)$.

□

3.4 The Segmented Orthogonal Matching Pursuit

Although in (3.26), the residue $r^{(p)}x$ is orthogonal to $\varphi_{\gamma_{p-1}}$, it may not be orthogonal to the other bases in the dictionary. Therefore, even in finite dimensional spaces, the rate of convergence of the MP algorithm can be extremely slow. This fact is illustrated in the following example.

Example 3.1 Let $\mathcal{H} = \boldsymbol{R}^2$, and $D = \{\varphi_1, \varphi_2\}$, where $\varphi_1 = [1, 0]'$, and $\varphi_2 = [\cos(\theta), \sin(\theta)]'$, and let $x = [0.5, \sqrt{3}/2]$. Then it can be verified that for small values of 8, the MP algorithm converges very slowly. The convergence curve for the above example when 8 = 20°, is shown in Fig 3.2.

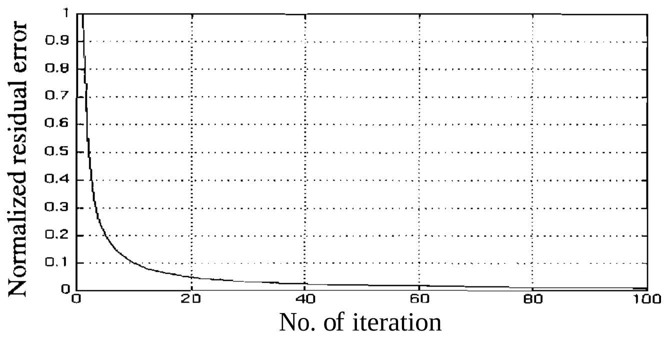


Fig. 3.2. Convergence rate of the matching pursuit algorithm for example 3.1.

The time varying characteristics of non-stationary signals can also slow down the convergence and sparsity of the matching pursuit expansions. In this case, because of the time varying behavior of signal, a single atom from the dictionary can not be a good matched to the local characteristics of the signal. Therefore, it is possible that the algorithm selects wrongly in the first few iterations, and then start correcting for these mistakes by iterating on the residue.

To overcome these problems a simple two-step algorithm is introduced in the following subsection. This algorithm greatly improves the performance of the original matching pursuit in terms of sparsity of expansion and speed of convergence. This algorithm is called the Segmented Orthogonal Matching Pursuit (SOMP').

3.4.1 The Segmented Orthogonalized Matching Pursuit Algorithm

In general, the atom φ_{γ_k} selected at each iteration of the MP algorithm is not orthogonal to the previously selected atoms φ_{γ_p}, for $0 \leq p < k$. When the algorithm subtracts the projection of $r^{(k)}\boldsymbol{x}$, it introduces new components in the directions of the previously selected atoms. This problem can be avoided by orthogonalizing the direction of projections at each step of the algorithm. The time varying characteristics of the signal can be exploited by a proper time (space) segmentation algorithm. If proper criterion for segmentation is selected, the algorithm will converge faster and consequently produces an sparse representation for the data. The SOMP algorithm for one-dimensional discrete signals of length N can be summarized as follows[2].

Let $\mathcal{D} = \{\phi_\gamma : \gamma \in \Gamma\}$ be a dictionary in a finite dimensional Hilbert space $\mathcal{H} = \mathrm{R}^N$, and let $\boldsymbol{x} = \{x[n]; n = 0, 1, \ldots, N-1\}$, represent a discrete sequence of length N, where N is an even integer. Assume in the first iteration of the MP algorithm the, atom φ_{γ_1} is selected from $\mathcal{D}$. The Segmentation and orthogonalization steps of the SOMP algorithm are performed as follows.

- Step 1: (segmentation)

Let $\boldsymbol{x}_{l_1} = \{x[n];\ n = 0, \ldots, N/2-1\}$, and $\boldsymbol{x}_{r_1} = \{x[n];\ n = N/2, \ldots, N-1\}$ be the left-child, and right-child of $\boldsymbol{x}$, at iteration one, respectively. Let $\mathbf{J}(\boldsymbol{x}, \varphi_{\gamma_1}) = \langle \boldsymbol{x}, \varphi_{\gamma_1}\rangle / \|\boldsymbol{x}\|$, be the measure of correlation between the selected atom φ_{γ_1} and the given signal x. Then segment $\boldsymbol{x}$ into its left-child and right-child if the following criteria are met

$$\mathbf{J}(\boldsymbol{x}, \varphi_{\gamma_1}) < min\{\mathbf{J}(\boldsymbol{x}_{l_1}, \varphi_{\gamma_1}), \mathbf{J}(\boldsymbol{x}_{r_1}, \varphi_{\gamma_1})\} \tag{3.29}$$

and

$$\|r^{(1)}\boldsymbol{x}\| > \|r^{(1)}\boldsymbol{x}_{l_1}\| + \|r^{(1)}\boldsymbol{x}_{r_1}\| \tag{3.30}$$

The objective function in (3.29) is basically the correlation ratio that was introduced in Definition 3.2. It is a strictly positive number [37] that can be used to estimate the closeness of the match between the selected atom and signal in every iteration of the MP

---

[2] *Extension to 2-D signals for image processing applications is straightforward.*

algorithm. The equation *(3.29)*states that if the atom is more correlated with any of the children, then segment the data, otherwise proceed with the next iteration. The condition in equation *(3.30)*ensures that the error at the boundary of the segmentation is finite. If needed, the segmentation procedure can be carried out on children of $\boldsymbol{x}$ in a similar manner. The segmentation step allows the MP algorithm to use different type of atoms at each iteration to represent different segments of the signal, and consequently can greatly improve the speed of convergence and quality of the expansion.

- Step *2: (Orthogonalization)*

The MP algorithm can be orthogonalized to ensure that at each step of iteration the residue is orthogonal to all the previous terms in the expansion. The procedure for orthogonalization of the MP is similar for $\boldsymbol{x}$ or any of its children. However to keep the notation simple we only discuss the orthogonalization when $\boldsymbol{x}$ is being expanded. Let $\vartheta_0 = \varphi_{\gamma_0}$, and assume that at iteration p, the MP algorithm selects φ_{γ_p}. This atom can be orthogonalized with respect to all the previously selected atoms by

$$\vartheta_p = \varphi_{\gamma_p} - \sum_{k=0}^{p-1} \frac{\langle \varphi_{\gamma_p}, \vartheta_k \rangle}{\|\vartheta_k\|^2} \vartheta_k \tag{3.31}$$

and the residual can be found by computing the orthogonal projection of $\boldsymbol{x}$ onto the orthonormal complement of the space that has been created by the previously selected atoms, as

$$r^{(p+1)}\boldsymbol{x} = r^{(p)}\boldsymbol{x} - \frac{\langle r^{(p)}\boldsymbol{x}, \vartheta_p \rangle}{\|\vartheta_p\|^2} \vartheta_p \tag{3.32}$$

using *(3.30)*and *(3.31)* we can write

$$\|r^{(p+1)}\boldsymbol{x}\|^2 = \|r^{(p)}\boldsymbol{x}\| - \frac{|\langle r^{(p)}\boldsymbol{x}, \varphi_{\gamma_p} \rangle|^2}{\|\vartheta_p\|^2} \tag{3.33}$$

The orthogonalization step ensures that the residue $r^{(p)}\boldsymbol{x}$ is orthogonal to all the previously selected atoms from the dictionary, and the signal $\boldsymbol{x}$ can be expanded as

$$\boldsymbol{x} = \sum_{k=0}^{p-1} \frac{\langle r^{(k)}\boldsymbol{x}, \varphi_{\gamma_k} \rangle}{\|\vartheta_k\|^2} \vartheta_k + r^{(p)}\boldsymbol{x} \tag{3.34}$$

The above orthogonalization procedure results in a Orthogonalized Matching Pursuit

(OMP). The orthogonalization process is an obvious step to improve the convergence of the matching pursuit. The need for orthogonal projections in an adaptive greedy algorithm was first noticed by Donoho in the context of projection pursuit regression in statistics and has also been studied independently in [40].

The SOMP algorithm uses steps 1 and 2 until a stopping criterion is met. For 1-D signals, after p iterations of the algorithm, the signal x can be represented by a binary tree of depth p, with x as its root, and the parameters of the expansion can be efficiently indexed at the nodes of the tree for further processing. For 2-D signals, the algorithm generates a quadtree in a similar manner.

3.4.2 Convergence of the SOMP Algorithm

Because of the orthogonality of the residues with previously selected atoms from the dictionary, in finite dimensional spaces the SOMP and OMP algorithms converge in a finite number of iterations.

Theorem 3.3 ***Let Hbe an N-dimensional vector space. Then for any $x \in \mathcal{H}$, orthogonal matching pursuit (OMP) algorithm converges in less than or equal to N iteration.***

Proof: The Proof is in the appendix.

The convergence of the SOMP algorithm can be proven as a direct consequence of Theorem 3.3. However, It has to be shown that the error introduced at the boundary of the segmentation is finite. But this is true by virtue of the condition in equation (3.30). Therefore, the convergence property of the SOMP algorithm can be summarized in the following theorem.

Theorem 3.4 ***Let Hbe an N-dimensional vector space. Then for any $x \in \mathcal{H}$, segmented orthogonal matching pursuit (SOMP) algorithm converges in less than or equal to N iteration.***

In general, for time-varying signals, the SOMP algorithm produces expansions that are more sparse than the OMP algorithm. This fact is illustrated in the following example.

Example 3.2 Consider the signal shown in Fig. 3.3(a). This signal consists of a slow sin wave followed by a medium sine wave. The dictionary is a WP dictionary formed with D4 atoms. The length of the signal is 512, and it is desired to expand the signal with 6 atoms.

The residue plot for OMP and SOMP algorithms is shown in Figs. 3.3(b) and 3.3(c), respectively. While an OMP expansion with 6 atoms results in a PSNR of 8.51 dB, the SOMP expansion with the same number of atoms has a PSNR of 9.39 dB.

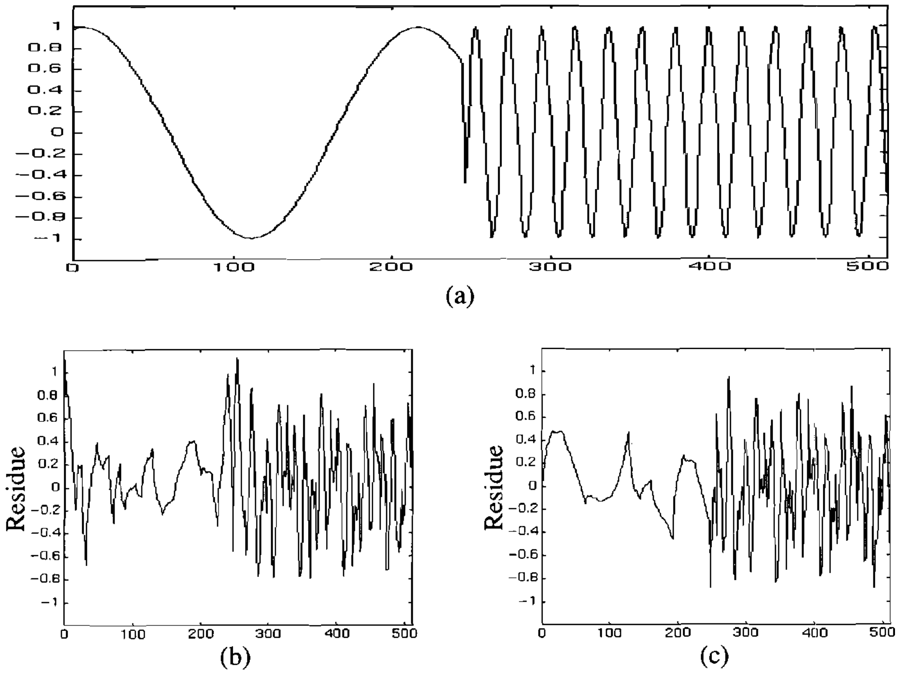


Fig. 3.3. The OMP and SOMP residue plots for example 3.2. (a) The test signal. (b) Residue plot for OMP, PSNR = 8.51 dB. (c) Residue plot for SOMP, PSNR = 9.39dB.

The computational complexity of the SOMP algorithm is discussed in the following subsection.

3.4.3 Computational Complexity of the SOMP Algorithm

We assume the space H is finite dimensional and $\mathcal{D}$ is a dictionary with finite number of atoms. This algorithm is first initialized by computing a set of inner products given by $\langle x, \varphi_\gamma \rangle$ for all $\gamma \in \Gamma$. These inner products can be stored in a table for further processing. The process of selecting the first best atom from the table requires $O(1)$ operations, on average. After the selection of the best atom from the dictionary, the inner product of the children and residuals must be found. The inner product for the residuals can be found in a recursive manner with the following updating formula

$$\langle r^{(p+1)}x, \varphi_\gamma \rangle = \langle r^{(p)}x, \varphi_\gamma \rangle - \langle r^{(p)}x, \varphi_{\gamma_p} \rangle \langle \varphi_{\gamma_p}, \varphi_\gamma \rangle \tag{3.35}$$

Since at iteration p, the first two terms on the right hand side of (3.35) are already computed, the inner product $\langle \varphi_{\gamma_p}, \varphi_\gamma \rangle$ is the only additional computation needed to obtain $\langle r^{(p+1)}x, \varphi_\gamma \rangle$. Assume the inner product of any two atoms in the dictionary requires $O(N)$ operations and there are $O(M)$ inner products that has to be computed at each iteration. Therefore, for p iteration of the algorithm the total complexity is $O(MNp)$. Now the number of operations for orthogonalization process has to be computed. The orthogonalization can also be performed in a recursive manner in $O(p^2)$ operations for p atoms that have been selected by the algorithm [41]. Therefore, for p iteration of the algorithm, the total number of computations for 1-D signals is given by $O(p^3+p^2MN)$.

3.4.4 Design of Optimal and Sub-optimal Dictionaries

The selection of a suitable dictionary is crucial to the quality of representation in any signal expansion algorithm. Clearly, a dictionary with a fixed set of atoms is not adequate for expansion of general class of signals. For example, the Fourier basis is not suitable for expansion of signals with discontinuities and Haar basis is not suitable: for expansion of smooth signals. On the other hand, a very large dictionary with a variety of atoms is not also desirable. Because an optimal expansion over general dictionaries is an NP-hard problem according to the Theorem 3.1.

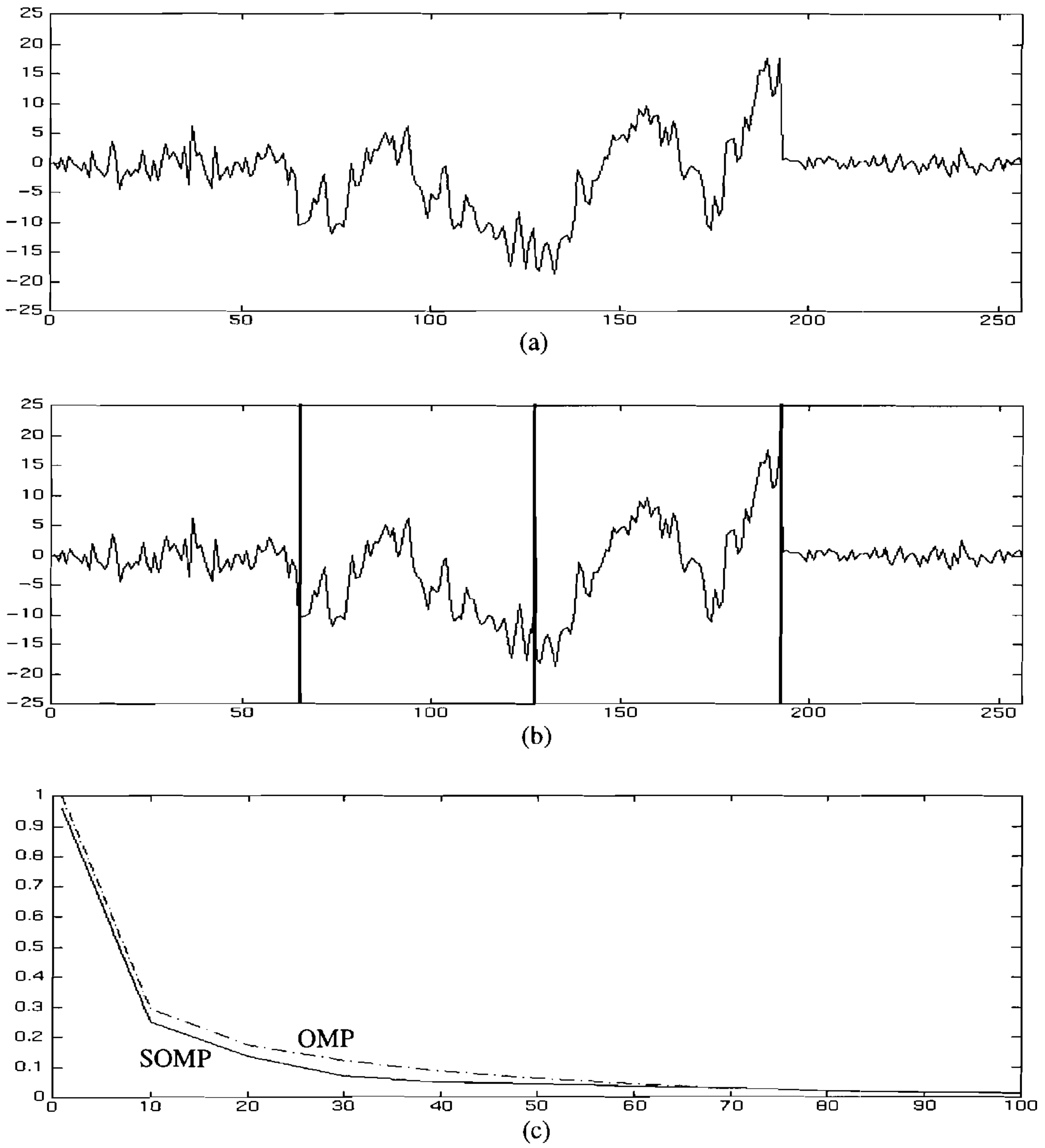
An alternative is to establish an analogy between the problem of Vector Quantization (VQ) and matching pursuit. In fact, a single iteration of the matching pursuit is very similar to the shape-gain VQ algorithm [18], and the MP algorithm can be seen as a cascade shape-gain VQ. Therefore, the generalized Lloyd algorithm can be used for design of optimal dictionaries for matching pursuit. However, given the complexity of the algorithm, and the need for training, suggests the need for a suboptimal solution.

In many applications, various heuristics rules can often be used to design a suboptimal dictionary of atoms. This heuristic rules can be deduced from the properties of various dictionaries, as discussed earlier in this chapter. According to this study, wavelet packets constitute a more general class of wavelets and offer a rich menu of atoms for image representation and image compression applications [3]. They are easy to implement and can. be efficiently indexed on a tree structure. This type of dictionary is also well suited for the SOMP algorithm. Because if a good match to the characteristics of the signal is not found, then the algorithm iteratively segments the signal until an efficient representation is obtained. Wavelet packet dictionaries are used in the following section, and will be used in the subsequent chapter for image compression.

3.5 Experimental Results And Conclusions

Simulations were carried out to demonstrate the effectiveness of the proposed algorithm. The first test signal consists of 256 samples from three different AR first order Markov processes with zero mean [78]. The first quarter has a variance of 50, and p = 0.1, the second and third quarter have a variance of 100, and r = 0.9, and the fourth quarter has a variance of 1, and p = 0.1, as shown in Fig. 3.5(a). The test signal 2 consists 512 samples from line 256 of the test image Peppers, and is shown in Fig. 3.6(a). Finally, the test signal 3 consists of 512 samples from line 256 of the test image Lena, and is shown in Fig. 3.7(a). The corresponding segmented signals with the SOMP algorithm are shown in Figs. 3.5(b), 3.6(b), and 3.7(b), respectively. The rates of convergence for the OMP and SOMP algorithms, and D4 wavelet packets are shown in Figs. 3.5(c), 3.6(c), and 3.7(c), respectively. In all cases the SOMP algorithm converges faster than the OMP algorithm. The rates of convergence for the test signals and OMP algorithm with Haar and D4 wavelet packets are shown in Fig. 3.8. All the vertical axes in this section represent the normalized mean squared error.

The experimental results in this section confirmed that the SOMP expansions are more sparse than the OMP algorithm. It also indicates that the rate of convergence can be reduced by selecting a suitable dictionary for signal expansion.



(a)

(b)

(c)

Fig. 3.4. Convergence rates of OPM and SOMP algorithms for the test signal 1. (a) The test signal 1, mixed AR(1) processes. (b) Segmented signal with the SOMP algorithm. (c) Convergence curves for the OMP and SOMP algorithms.

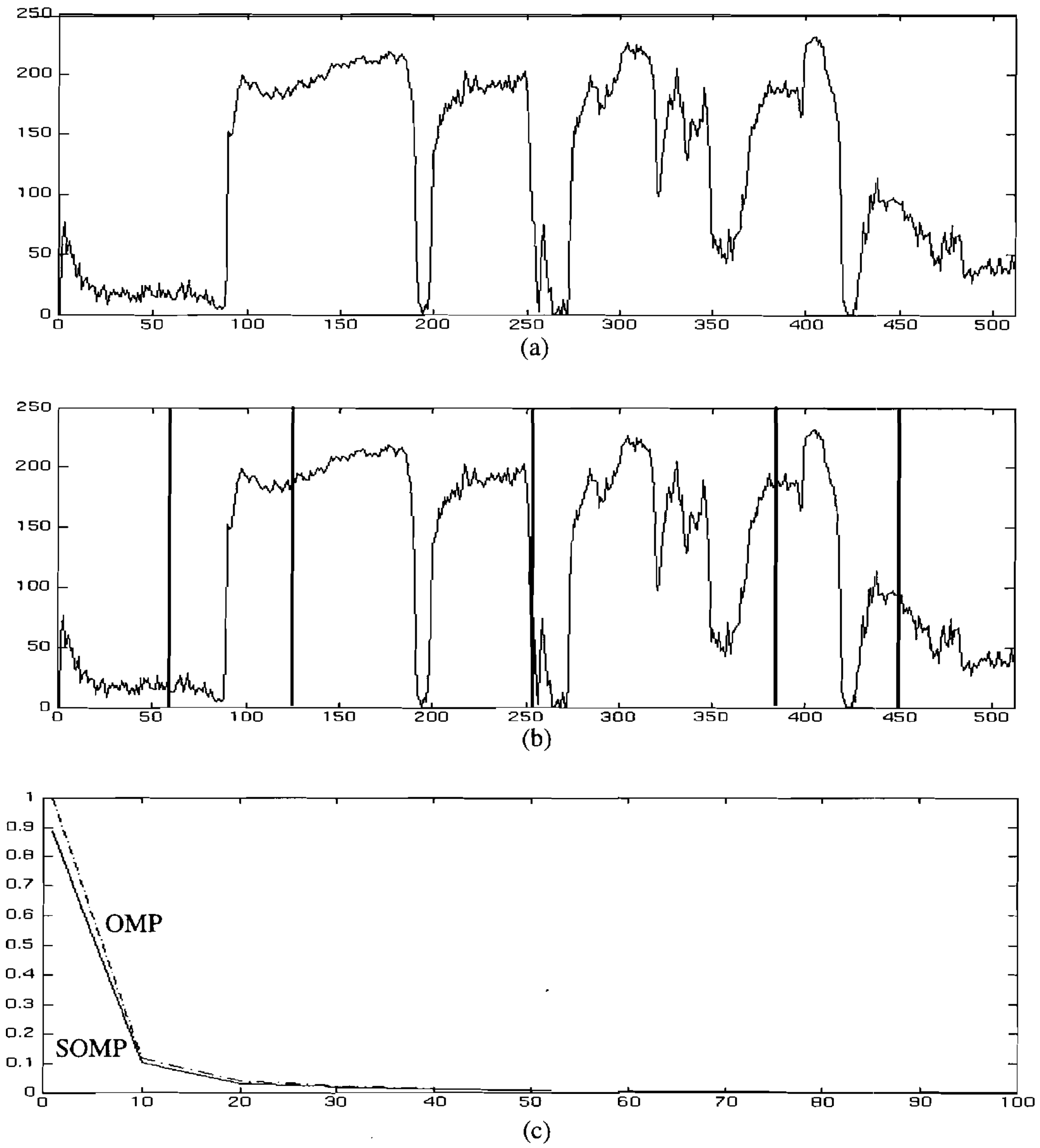


Fig. *3.5.* Convergence rates of OPM and SOMP algorithms for the test signal *2.* (a) The test signal *2,* Line *256* of test image Peppers. (b) Segmented signal with the SOMP algorithm. (c) Convergence curves for the OMP and SOMP algorithms.

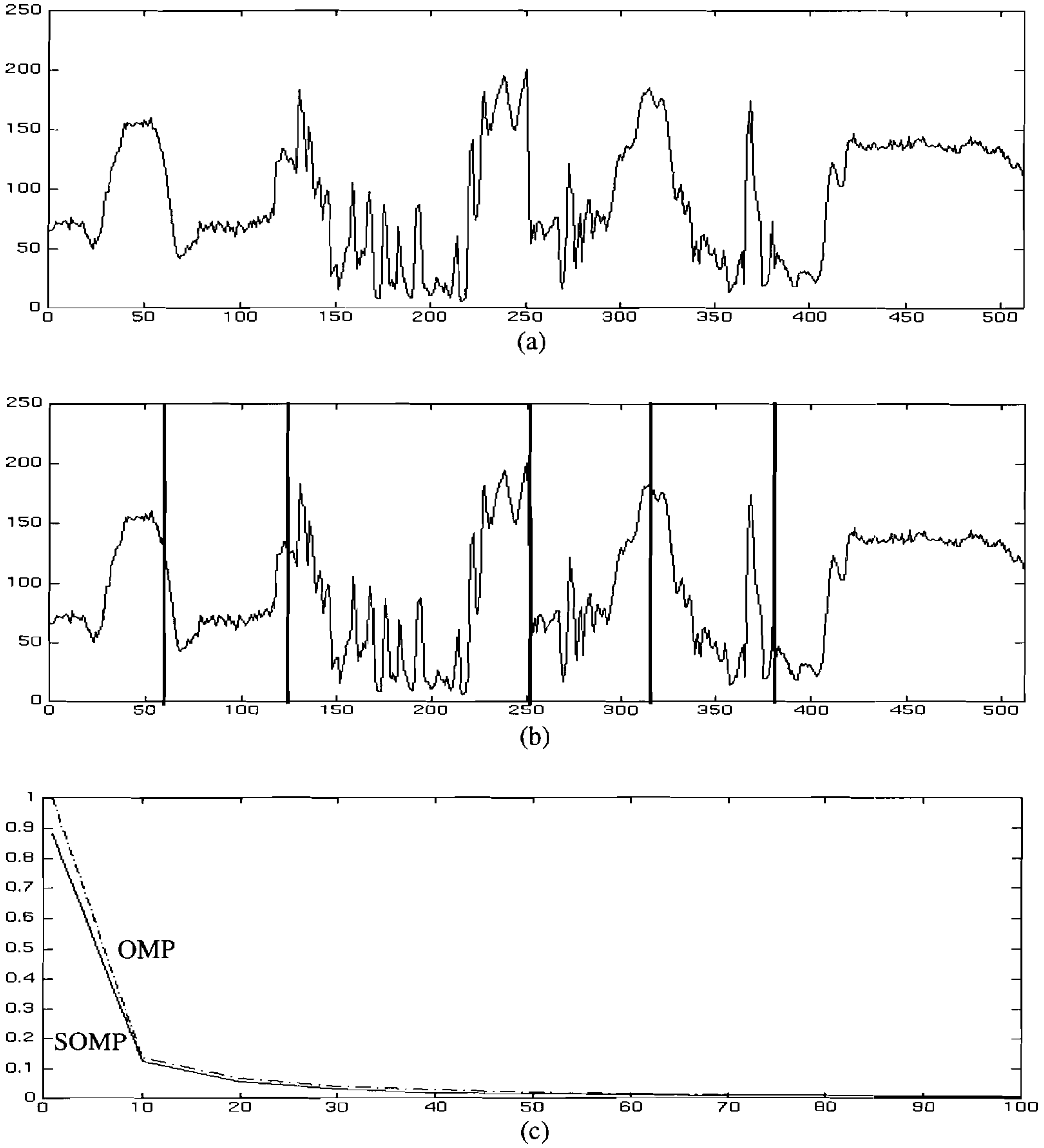


Fig. *3.6.* Convergence rates of OPM and SOMP algorithms for the test signal *3.* (a) The test signal 3, from line *256* of Lena. (b) Segmented signal with the SOMP algorithm. (c) Convergence curves for the OMP and SOMP algorithms.

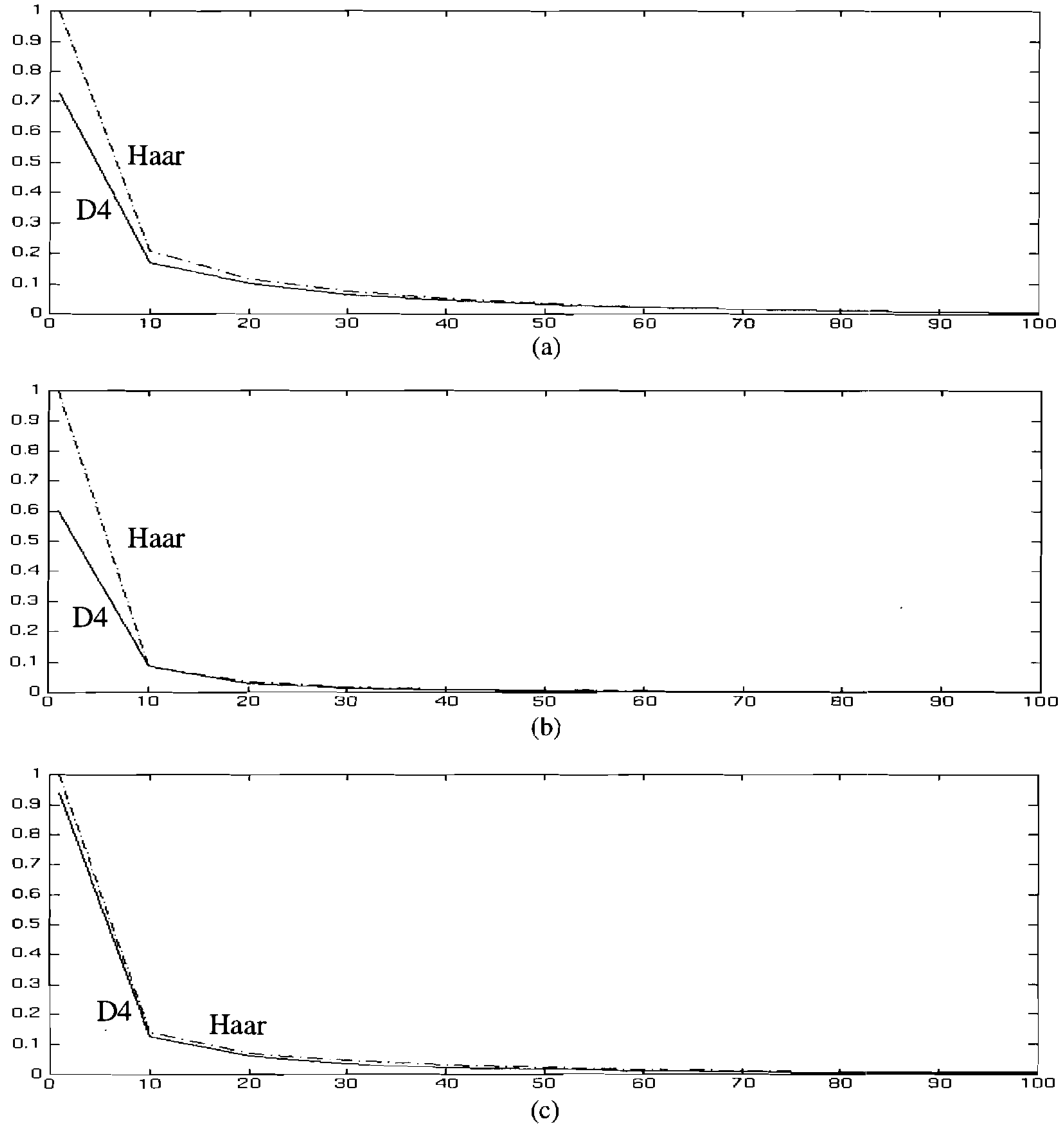


Fig. 3.7. Convergence of the OMP algorithm with Haar and D4 WP dictionary. (a) Test signal 1 (b) Test signal 2. (3) Test signal 3.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

4. ADAPTIVE MULTIRESOLUTION IMAGE CODING WITH QSOMP

4.1 Introduction

The storage and transmission of digital images in their original or raw form is usually very expensive or impractical. In order to make the widespread use of visual information in multimedia applications practical, data compression algorithms that operate at very low bit rates are needed. Moreover, progressive transmission is required in many multimedia applications where a user may only have access to a low bandwidth communication channel. For example, if progressive transmission is used in a telebrowsing application, the user can stop the transmission of an intermediate version of an image, if it is of no interest to him. This can effectively reduce the required search time and bandwidth. In general, a good image compression technique for interactive multimedia systems should be able to operate at very low bit rates (below 0.25 bits/pixel), produce an embedded bit stream, and be easy to implement.

The main objective in any lossy compression technique is to optimize the trade off between the amount of compression, measured in bits per pixel (bpp), and the reconstructed image quality, measured by the Peak Signal-to-Noise Ratio (PSNR) or subjective evaluation. The above mentioned requirements motivate the use of *multiresolution* image coding techniques in future communication systems. The current techniques used in very low bit rate lossy image compression include fractal coding, segmentation based coding, and subband/wavelet based coding.

Fractal coding techniques were developed based on the theory of ***iterated contractive transformations*** and ***collage theorem*** to exploit the existing self similarities of natural images [53].

Segmentation based or so called ***second generation*** image coding techniques try to exploit ***structural*** properties of the image in order to achieve compression at very low bit rates [54]. In these techniques the image is segmented using edge maps or hierarchical data structures. Hierarchical data structures have gained popularity because they are relatively easy to implement and are multiresolution in nature [55].

Subband coding techniques have been developed based on the theory of filter banks in signal processing [2] and compactly supported wavelets in applied mathematics [22]. These multiresolution image coding techniques are rich in theory and easy to implement. The initial wavelet coding algorithms were designed to exploit the wavelet's transform ability to compact energy into low frequency coefficients. These early subband coding algorithms demonstrated modest improvement in coding efficiency over standard transform based algorithms [2, 3]. In recent years, a new class of wavelet based algorithms that exploit the wavelet's space-frequency compaction properties have achieved significantly improved performance over the previous techniques. These algorithms have been developed based on the fact that the wavelet decompositions can efficiently ***frequency compact*** the energy into a small set of low frequency coefficients, and also ***spatially compact*** the energy into a small set of high-frequency coefficients around the edges. The most popular wavelet based technique in this class is the Embedded Zerotree Wavelet (EZW) coder [56], that was further improved in [57]. Further improvement was achieved in [58] by using a Space-Frequency Quantizer (SFQ) for encoding of the wavelet coefficients. SFQ uses a jointly optimized spatial zerotree quantizer and scalar frequency quantizer in a rate-distortion sense. A novel algorithm that unifies the EZW data structure and fractal coding was proposed in [59]. Finally, adaptive image compression techniques based on wavelet packets have shown promising results at lower bit rates.

Because of the sparsity, high resolution, and robustness properties of signal representation with the methods of Matching Pursuit, it is possible to use these representations to achieve compression at lower bit rates. In this chapter two new adaptive multiresolution algorithms are proposed for image compression at low bit rates. The first algorithm is based on the encoding of QuadTree hierarchical data structures with Matching Pursuit (QTMP) [60]. The second algorithm uses the Quantized Segmented Matching Pursuit (QSOMP) to represent an image by a quadtree data structure. It will be shown that the QSOMP coding technique performs better than the existing wavelet based algorithms at rates below 0.5 bits/pixel and its performance is comparable to other techniques at higher bit rates. The organization of this chapter is as follows. Section 4.2 presents the QTMP coding algorithm. Section 4.3 introduces the QSOMP algorithm. Finally, experimental results and concluding remarks are provided in section 4.4.

4.2 Image Compression with Hierarchical Data Structures and Matching Pursuit

The QTMP or QuadTree Projection Pursuit (QTPP) algorithm, is a novel algorithm that adaptively encodes the image segments that have been obtained from.a variance-based quadtree segmentation. In [60] it was shown that this algorithm performs considerably better than JPEG in terms of subjective evaluation and PSNR, and its performance is comparable to the EZW algorithm at rates below 0.25 bits/pixel. In [61] it was shown that the QTMP can be efficiently implemented with a three layer neural network. Finally, the application of the QTMP algorithm in color image compression was studied in [62].

4.2.1 Image Segmentation with Hierarchical Data Structures

Natural gray-level images can usually be divided into regions of different sizes with variable amounts of detail and information. There are a variety of hierarchical data structures for representing spatial data at multiple resolutions [55]. These models have been developed based on the principle of recursive decomposition and have found many applications in computer graphics, computer vision, pattern recognition, solid modeling, image processing, and geographic information systems. Hierarchical data structures are attractive for the following reasons

- They are relatively simple to implement.
- They adoptively decompose the image into subregions.
- The decomposition actually results in image segmentation.
- The tree structure can be efficiently encoded with a negligible amount of overhead.

The most popular hierarchical data structures for image compression applications are QuadTree (QT) and Binary Space Partitioning (BSP) binary tree [5, 63, 64].

Quadtree decomposition is a simple technique of representing images at multiple resolutions. In this technique, the image is recursively divided into four equal ***square*** regions depending on the ***activities*** in the blocks [55]. Quadtree segmentation of a 2^{n}x2^{n}

image results in a tree whose root represents the original image at resolution level zero, and the four equally sized squares represent its children at resolution level one. Each pixel at resolution level j, has its own intensity x', and the parent node intensity is equal to the mean value of the intensities of its children nodes. At resolution level j (except for j = (n+1)), the node intensity is given by

$$x^{j}_{k,l} = \frac{1}{4} \sum_{q=0}^{1} \sum_{m=0}^{1} x^{j-1}_{2k+q,\,2l+m} \tag{4.1}$$

for j = 1,..., n and k, l = 0,..., 2^{n-j}-1. At each node, a decision must be made as to whether to decompose the corresponding block into four squares of equal sizes or to stop the decomposition. In order to arrive at a decision, several measures of activity have been proposed in the literature. The most widely used measure of activity is the absolute *difference* [55]. At each node, the value of the absolute difference is compared to a threshold value $\mathcal{T}$ as

$$\bigcap_{q,m=0}^{1} \left|x_{k,l} - x_{2k+q,\,2l+m}\right| \le \mathcal{T} \tag{4.2}$$

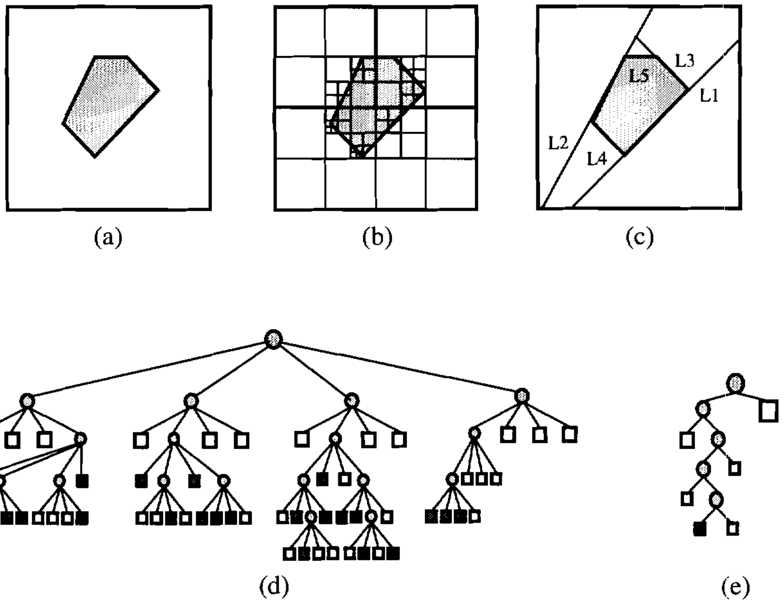
If the absolute difference is smaller than $\mathcal{T}$, the recursive decomposition at that node is stopped. Otherwise, the node is further decomposed into four squares of equal sizes.

The Binary Space Partitioning (BSP) tree is a binary tree whose root represents the original image [65]. This recursive partitioning technique takes as input an unpartitioned region $\mathcal{R}$ (initially the entire image), and a line l (selected according to some criteria) that intersects $\mathcal{R}$, and produces as output two new regions formed by partitioning $\mathcal{R}$ by l into two half-regions, $\mathcal{R}^-$ and $\mathcal{R}^+$. The two half regions can then be similarly partitioned in a recursive manner until a *termination* criterion is met. This results in a hierarchy of regions in which the leaves of the tree correspond to unpartitioned convex regions called cells. A good segmentation is obtained when the pixel values within each cell are homogeneous. This desired feature (i.e. homogeneous cells) can serve as a terminating criterion. The non-leaf nodes of the BSP tree are associated with the partitioning lines, and the leaves represent the cells of the image. Every node in the tree represents a convex region of the image. Finally, the parameters of the partitioning line (p, θ) should be stored in the nodes of the BSP tree. There are two techniques for obtaining the parameters of the partitioning lines in a **BSP** tree. The first one is based on a boundary-based Hough transform technique, and the second one uses an optimization technique based on minimization of the l_1 norm [65, 66]. In the latter case, the parameters of the partitioning line are obtained by minimizing the error ε, given by

$$\varepsilon(\mathcal{R}_l(\rho, \theta)) = \sum_{k,\,l \in \mathcal{R}_l(\rho, \theta)} [x_{k,l} - z_{k,l}]^2 \qquad (4.3)$$

where (ρ, θ) correspond to all possible values for the partitioning line l, $\mathcal{R}_l(\rho, \theta)$ denotes the set of regions associated with l, $x_{k,l}$ represents the pixels in the image, and $z_{k,l}$ represents the model for the cell.

Although the process of generating a **BSP** tree is more complex than a quadtree, the **BSP** tree is more efficient than the quadtree in representing images. These facts are illustrated in Fig. 4.1.



(a) (b) (c)

(d) (e)

Fig. 4.1. Quadtree and **BSP** tree segmentation of a synthetic image. (a) image of a polygon. (b) Quadtree segmentation map of (a). (c) **BSP** tree segmentation map of (a). (d) The quadtree of (a). (e) The **BSP** tree of (s).

Unlike the quadtree representation, which only allows square segmented regions, the segmented regions or cells of a **BSP** tree can be arbitrary shaped polygons. This may result in a more efficient and compact representation of digital images. In quadtree, the partitioning lines are simply vertical and horizontal lines at dyadic intervals and no extra computation or bits are needed to encode these lines, but the representation is considerably more complex than a BSP tree. For the BSP-tree, most of the bits and computational power is used for generating and encoding the partitioning line parameters (ρ,θ), at each node. In [5], the coding efficiency of the BSP and QT trees for natural images were compared. It was shown that the BSP coding gain is marginally higher than the QT tree, but it is computationally more expensive.

4.2.2 The QTMP Image Compression Algorithm

After generating the Quadtree representation of the image, each subregion or block in the tree can be coded using the matching pursuit image approximation technique. In every iteration, a function that best approximates the current image in the given block is selected from the dictionary. In the first step of the iteration, the current image is the original image, and in step p, the current image is the residual (error) image which is obtained by subtracting linear combinations of all $(p\text{-}1)$ previous approximations from the original image. Although various measures of error such as mean square error (or l_2 norm), absolute error (or 1, norm), and uniform error (or l_∞ norm) can be used to assess the quality of the approximation, l_2 is used because of its mathematical tractability.

In order to obtain an efficient representation for each block, the dictionary should include both continuous and discontinuous functions, each with different degrees of smoothness and regularity. Two possible disadvantages of considering a large dictionary are a slight increase in coding overhead, and an increase in computation. In [62], it is shown that even with a dictionary of sigmoidal functions, the QTMP algorithm performs considerably better than JPEG at rates below 0.4 bits/pixel.

The QTMP image compression algorithm is shown in Fig. 4.2. In this algorithm $x_{k,l}$ denotes the intensity of the image at location (k, 1), $\hat{x}_{k,}$, its estimated value, and $r_{k,l}^{(p)}$ the residual image at iteration p. $\Theta^{(p)} = \{\varphi_\gamma^{(p)}, \underline{\alpha}^{(p)}, \beta^{(p)}, \underline{\lambda}^{(p)}\}$ represents the set of parameters at iteration p, $d(.,.)$ the desired error metric, $s^t = (k, l)$, and optimal values are indicated by the superscript "*".

```
1. Create the Quadtree segmentation map of the image;
Start {
    2. Select a dictionary $\mathcal{D} = \{\varphi_\gamma\}_{\gamma \in \Gamma}$;
    3. For every block $\mathcal{B} = x$ in segmentation map, given desired $PSNR_d$;
    4. While ($PSNR \leq PSNR_d$) {
    5. Let $r^{(0)} = x$;
    6. At iteration p, select the optimal parameter vector $\Theta_p^*$;
       $\Theta_p^* = arg\left\{ \min_{\varphi_p \in \mathcal{D}} \left\{ \min_{\underline{\alpha}^{(p)}, \beta^{(p)}, \underline{\lambda}^{(p)}} \sum_{s \in \mathcal{B}} d(r_{k,l}^{(p-1)}(\lambda^{(p)}\varphi_p((\underline{\alpha}^{(p)})^t \bullet s + \beta^{(p)}))) \right\} \right\}$;
       $\hat{x}^{*(p)} = \sum_{i=1}^{p} \lambda_i^* \varphi_i^* (\underline{\alpha}_i^{*t} \bullet s + \beta_i^*)$;
       $r^{(p)} = x - \hat{x}^{*(p)}$;
    7. Compute the $PSNR_p$ and the Bit Rate, $R_p$, at iteration p);
    8. The expansion is $\hat{x}^{*(p^*)} = \sum_{i=1}^{p^*} \lambda_i^* \varphi_i^* (\underline{\alpha}_i^{*t} \bullet s + \beta_i^*)$;
    9. Quantize, and code.
end.)
```

Fig. 4.2. The QTMP compression algorithm.

In the above algorithm the error metric is usually the mean square error, and the linear expansion in step 6 is written in a different format to resemble the. output of a three layer neural network. The neural network corresponding to the above coding scheme is shown in Fig. 4.3.

4.2.3 Quantization and Coding of the Parameters

In order to achieve low bit rate compression, the optimum parameters for each block must be quantized before encoding. The optimum quantizer, in the mean square error sense, is a non-uniform quantizer that matches the probability density function of its input signal.

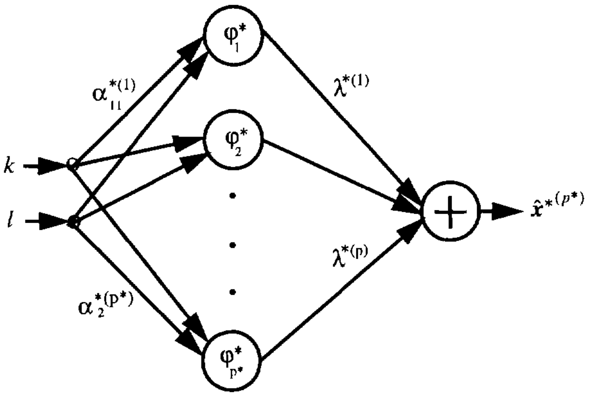


Fig. 4.3. Neural network implementation of the QTMP algorithm. Parameter β is mimicked in the node function φ_j.

Based on the distribution of the weights and biases and their dynamic ranges ranges, separate Lloyd-Max quantizers were designed for each set of block parameters [16]. Finally, the resulting quantized parameters were entropy coded. Various coding schemes such as Huffman coding, Shannon-Fano coding, and arithmetic coding [16] were considered, and arithmetic coding was chosen because it has the following advantages

- It can approach the entropy limit in coding efficiency.
- It requires only one pass through the data.
- It is generally faster than Huffman coding.
- In arithmetic coding, the encoder and decoder can work on-line.
- It requires no *a priori* analysis of the data set for bit allocation.

Experimental results for the QTMP algorithms are provided in section 4.4.

4.3 The QSOMP Image Compression Algorithm

There are two basic problems with the QTMP algorithm. The first problem results from the fact that the processes of segmentation and expansion in this algorithm are independent. The second problem is associated with the sub-optimal quantization and coding of the MP parameters. In the QSOMP algorithm, the segmentation is performed based on the quality of representation, and the quantization of the parameters is performed in an optimal Rate-Distortion (*R-D*) sense. These facts are explained in more details in the following subsections.

4.3.1 Integration of Segmentation with the Quality of Representation

In the QTMP algorithm, the image is first segmented into homogeneous regions. Then the matching pursuit is used to construct a code for each segment. Therefore, the processes of segmentation and expansion are totally independent. In other words, the QTMP algorithm fails to utilize the concept of the adapted bases [3] for signal expansion in a global sense. However, it produces a compact representation for the homogeneous regions of a quadtree segmentation map, by performing a local adaptive search over a small dictionary. For example, consider the encoding of a synthetic image with a dictionary of four atoms, as shown in Fig. 5.4. The QTMP algorithm first segments the image into four quarters, then uses the MP algorithm on each segment, and for this example converges in one step. This results in a quadtree of depth one, with one atom at each leaf. However, if the algorithm would have searched the dictionary for the best match before segmentation, the same image could have been represented with one or at most two atom.

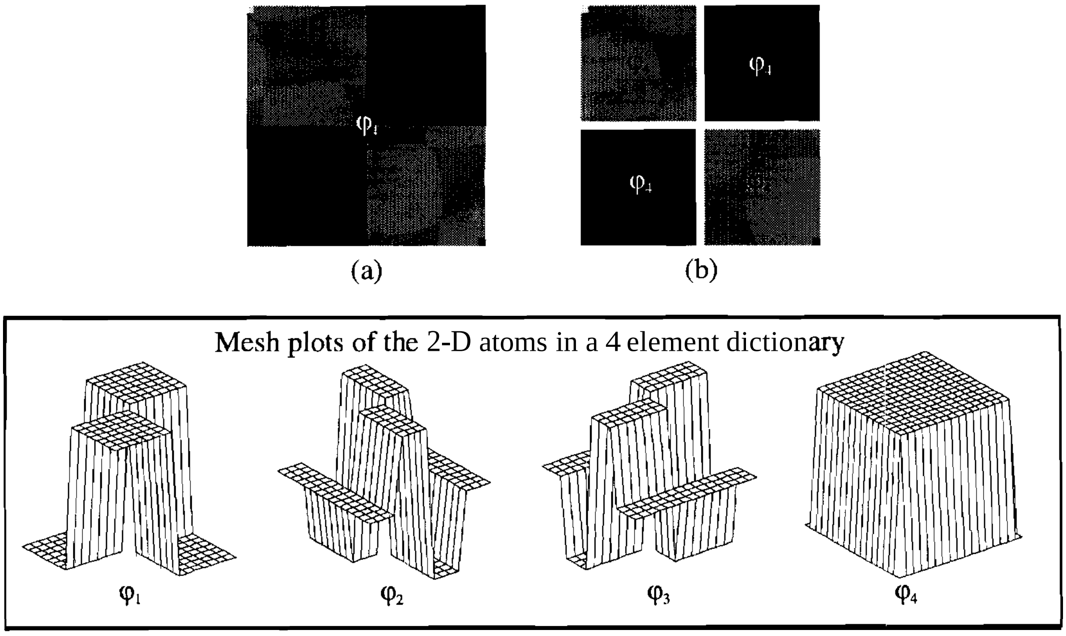


Fig. 4.4. Separation of segmentation and quality of representation in QTMP algorithm. (a) The synthetic image can be represented with a single atom. (b) The QTMP algorithm segments the image and needs four atoms.

In contrast to the QTMP algorithm, the SOMP algorithm searches the dictionary for a best match before it makes a decision regarding the segmentation of the image. It is straightforward to show that the performance of the SOMP algorithm is lower bounded by the QTMP algorithm.

4.3.2 The Quantized SOMP Algorithm

In the QTMP algorithm, quantization of the MP parameters is performed after all parameters have been collected. This procedure is not optimal for a number of reasons. In this algorithm quantization is independent of the greedy algorithm. At every step of the greedy algorithm, the new coefficients of the expansion are obtained by using the results of the previous step. In addition, this algorithm normally produces high precision coefficients that might limit its use by general purpose hardware. In addition, the PSNR computation at step 7 of the algorithm (see Fig. 4.2), is performed on unquantized coefficients and can not be used as a reliable stopping criterion for the QTMP algorithm. To avoid these problems, the coefficients of the expansion can be quantized at every step of the iteration in an R-D sense. This strategy was used in the SOFM algorithm of chapter three, as explained below.

Let $Q(\bullet)$ be a scalar quantizer with a variable step size, and $R(\bullet)$ be the desired rate in bits. At iteration P of the SOMP algorithm, after selection of the best atom φ_{γ_p}, the coefficients of the orthogonal projections should be found and quantized appropriately. The orthogonalization step is performed by orthogonal projections in a recursive manner, as before, and the residue can be computed as

$$r^{(p)}\boldsymbol{x} = r^{(p-1)}\boldsymbol{x} - Q(\langle r^{(p-1)}\boldsymbol{x}, \varphi_{\gamma_p}\rangle)\varphi_{\gamma_p} \tag{4.4}$$

Now, to incorporate an R-D optimal procedure for the selection of the best atom in the dictionary, the variation in rate and distortion should be quantified. The change in distortion can be defined as

$$\Delta D(\varphi_\gamma) = \left\|r^{(p-1)}\boldsymbol{x}\right\|^2 - \left\|r^{(p)}\boldsymbol{x}\right\|^2 \tag{4.5}$$

Although, the terms in the right hand side of equation (4.4) may not be orthogonal because of the quantization, (4.5) and (4.4) can be used to approximate $\Delta D(\varphi_\gamma)$ as

$$\Delta D(\varphi_\gamma) \approx \left|Q(\langle r^{(p-1)}\boldsymbol{x}, \varphi_\gamma\rangle)\right|^2 \tag{4.6}$$

The variation in rate can be quantified by

$$\Delta R(\varphi_\gamma) = R(Q(\langle r^{(p-1)}\boldsymbol{x}, \varphi_\gamma\rangle)) + R(selecting\ \varphi_\gamma) \tag{4.7}$$

Therefore, an R-D optimal quantized SOMP can be obtained by selecting the best atom φ_{γ_p}, as

$$\frac{\Delta D(\varphi_{\gamma_p})}{\Delta R(\varphi_{\gamma_p})} \geq \alpha\frac{\Delta D(\varphi_\gamma)}{\Delta R(\varphi_\gamma)}, \quad \gamma \neq \gamma_p \tag{4.8}$$

The quantized version of the SOMP produces a bit stream that is optimal in an R-D sense. Clearly, this holds only if a variable step size quantizer is used. A good algorithm to find the best quantizer is studied in [45] (see also chapter **3).** When the above quantization strategy and selection criterion is used in SOMP, the resulting algorithm is called the Quantized SOMP (QSOMP) algorithm.

4.3.3 Near Optimal Dictionaries for Image Compression

As mentioned in the previous chapter, the optimal design of dictionaries for the matching pursuit algorithm is analogous to the codebook design problem in vector quantization. Therefore, one should to use suboptimal dictionaries with a heuristic selection criteria. For image representation and compression applications, a rich collection of atoms with various degrees of smoothness or regularity are needed. In this chapter, a set of orthonormal one-dimensional (1-D) atoms were used, based on the study in [67], to create a dictionary of separable 2-D wavelet packets [2].

Using 1-D atoms in the dictionary has two major advantages. First, the directional preference leads to the concept of steerable wavelets [67]. This characterization can be exploited to create a richer menu of atoms for 2-D expansions, as illustrated in Fig. 4.5. The second advantage is the considerable reduction in the computatiorial complexity of searching for the best atom in the dictionary. The computation of inner products for 2-D atoms can be performed more efficiently with two 1-D atoms. Because of these advantages, this dictionary of steerable wavelet packets, is called a near optimal dictionary.

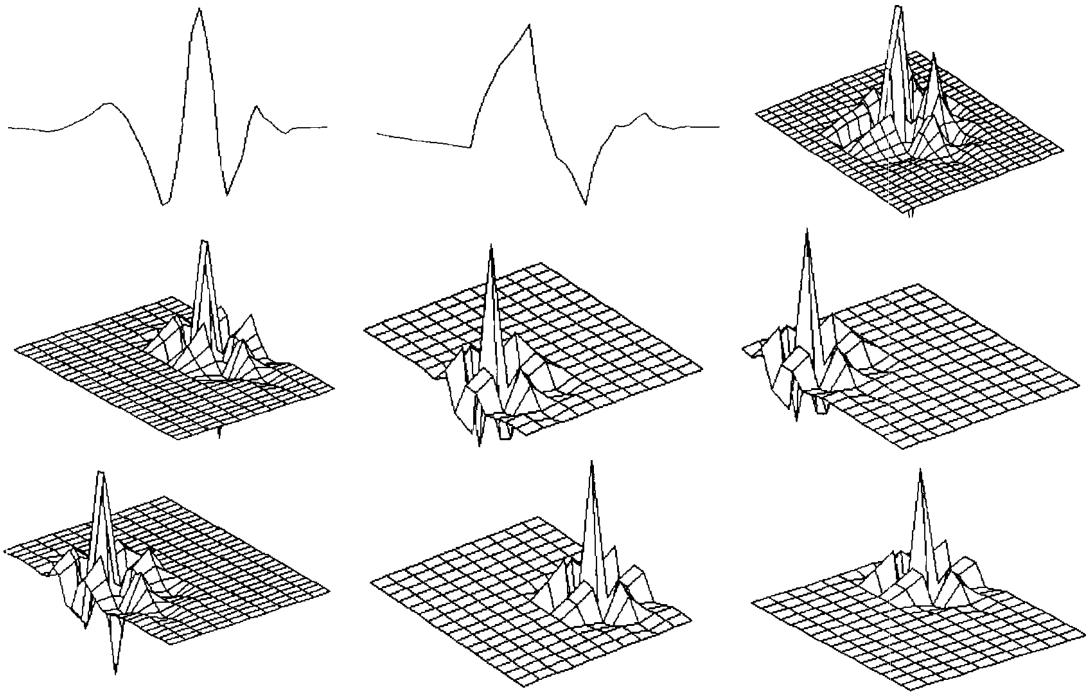


Fig. 4.5. Examples of steerable 2-D atoms obtained from product of 1-D atoms.

4.3.4 The QSOMP Image Compression Algorithm

The QSOMP image compression algorithm is shown in Fig. 4.6. The main features of this algorithm are

- It adaptively finds the best atomic representation of the image over a dictionary of functions.
- It creates a multiresolution hierarchy of atoms with lower frequency information at the beginning of the stream (near the node of the tree), and the detail information appended to the bit stream at every iteration of the algorithm.
- The code is embedded. The encoder (decoder) can cease at any time and provide the best achievable representation of an image over the given atoms in the dictionary.
- It is possible to control the algorithm by constraining the PSNR, target bit rate, and number of atoms.
- It performs better than the existing wavelet based image compression algorithms at lower bit rates, and its performance is comparable at higher rates.

```
Start {
    I. Select a dictionary of steerable I-D wavelet packets $\mathcal{D} = \{\varphi_\gamma\}_{\gamma \in \Gamma}$;
    2. Let $r^{(0)} = x$; (i.e. let the initial residual be the original image)
    3. Let $R_d$, equal to bit budget, PSNR, desired quality, $NC_d$ desired no. of coefficients;
    4. While ($PSNR \le PSNR$, or $R \ge R_d$ or $NC \le NC_d$) {
        5. Perform QSOMP on x {
        6. At iteration P select the best atom;
        7. Pack the atom and index at the quadtree node;
        8. If segmenting {
                9. Create the embedded code by adaptive arithmetic coding;
                10. Mark end of tree if segmented (insert EOT symbol);
            I
        I I. Compute PSNR„ $R_p$, and NC at iteration p;
        };
    };
```

Fig. 4.6. The QSOMP adaptive multiresolution compression algorithm.

In the QSOMP algorithm a quadtree data structure is used. QT has a simple structure, and can be efficiently encoded [55, 68], since the quantized coefficients are available at each step of the iteration. If the algorithm decides to segment the image, it will store the required information regarding the atoms and atom indices at that node using link lists. The: atoms in the dictionary are wavelet packets that can be efficiently indexed on a quadtree as well. The optimal quantizer for this algorithm is a variable step size quantizer, however using a variable step size quantizer requires extra overhead. When the data is segmented and the tree expanded to the next resolution, the available data on the parent nodes is entropy coded with an adaptive arithmetic encoder and inserted into the bit stream followed by an End of Tree (EOT) symbol. In contrast to most of wavelet based image compression algorithms, which first expand the signal and then select the best atoms by exploiting the space-frequency characteristics of the wavelets, this algorithm adaptively selects the best atom and progressively improves its estimate by adding more detail to the image. Experimental results are presented in section 4.4. The PSNR values and rate distortion curves of the test image Lena at rates below one bits/pixel for QTMP, QSOMP, JPBG [1], EZW [56], Improved EZW (IEZW) [57], SFQ [58], Single Tree Algorithm [3], and Double Tree Algorithm [3] are presented at the end of section 4.4.

4.4 Experimental Results and Conclusions

Simulations were carried out to demonstrate the effectiveness of the proposed algorithms. The test images Peppers and Lena (512x512x8) are shown in Figs. 4.7(a), and 4.8(a), respectively.

For the QTMP algorithm, the quadtree segmentation maps were generated by first dividing the test images into blocks of 32x32. These blocks were further subdivided into blocks of 16x16 and 8x8 based on different measures of activity. Both visual entropy and image variation measures [68] that produced nearly identical segmentation regions were examined. The segmentation map based on image variation measure for the test images are shown in Figs. 4.7(b), and 4.8(b). Once the image was segmented, the resulting blocks were coded using the following procedure. Every function in the dictionary was used to obtain the best approximation for each block by optimizing the parameters of the function in the mean square error sense. The optimal basis function was selected by performing a greedy search over all the functions in the dictionary. This provided the first level approximation for each block. The next levels of approximation were obtained by repeating the above process on the residual errors. The process was tenninated when the overall error dropped below the desired threshold, or the desired bit rate was achieved. The dictionary considered in the experiments included the Daubechies orthonormal wavelets of order 4 or less. Although theoretically a greedy search on a large dictionary should produce a better approximation of the image, the experimental results showed that a few basis functions (3 to 5, with various degrees of smoothness) are sufficient to produce almost identical results. The parameters of the optimal bases for each block were quantized using Lloyd-Max quantizers. The histograms of the parameters of each block were used to design the quantizers. For the experiments, Gaussian and Laplacian Lloyd-

Max quantizers with 5 or 6 bits were used, which provided signal-to-quantization ratios in the range of 33-35 dB for all the parameter sets. Finally, the quantized parameters for each block were separately encoded with an adaptive arithmetic encoder [56] to form the compressed image. The decoded Peppers test image at the bit rate of 0.125 bit/pixel with PSNR of 30.38 dB is shown in Fig. 4.7(c). For comparison, the JPEG algorithm was used to encode Peppers at a bit rate of 0.125 bit/pixel and PSNR of 23.75 dB. The severe blocking artifact present at the JPEG decoded image can be seen in Fig. 4.7(d). The decoded Lena test image at the bit rate of 0.125 bit/pixel with PSNR of 30.25 dB is shown in Fig. 4.8(c). For comparison, the JPEG algorithm was used to encode Lena at a bit rate of 0.125 bit/pixel and PSNR of 26.75 dB. The severe blocking artifact present at the JPEG decoded image can be seen in Fig. 4.8(d). The SQOMP algorithm was tested on the same test images. The results for Peppers and Lena at rates of 0.5 bitslpixel, 0.25 bits/pixel, 0.15 bits/pixel and 0.0625 bits/pixel, are shown in Fig. 4.9 and 4.10 respectively. The subjective quality of the reconstructed images are excellent. The dictionary consisted of steerable Daubechies wavelet packets. Fixed step size scalar quantizers were used in these experiments. The computational complexity of searching for a single atom in the QSOMP algorithm, for a block of NxN in the image, with L I-D atoms of size N is given by $L^2N + 2LN^2$. Typical values for N are 16, 32, and 64. Therefore, the complexity of the algorithm highly depends on the size of the dictionary. However, the experimental results showed that the performance of our algorithm remains stable even for very small size dictionaries. Moreover, In [62] it was shown that even with a single element dictionary, QTMP algorithm outperforms JPEG at rates below 0.25 pits/pixel. For the QSOMP algorithm, the PSNR values along with the number selected atoms for the test image Lena are listed in Table 4.1.

For the test image Lena, the PSNR of a few recent wavelet based image compression algorithms is shown in Table 4.2 and Fig. 4.11. The results show that the QSOMP algorithm performs better than the existing wavelet compression techniques at lower bit rates, and its performance is comparable at higher rates. Finally, the computational complexity of the algorithm can be further reduced by adopting a sub-optimal search criterion, and can be justified with the increasing power of digital computers.

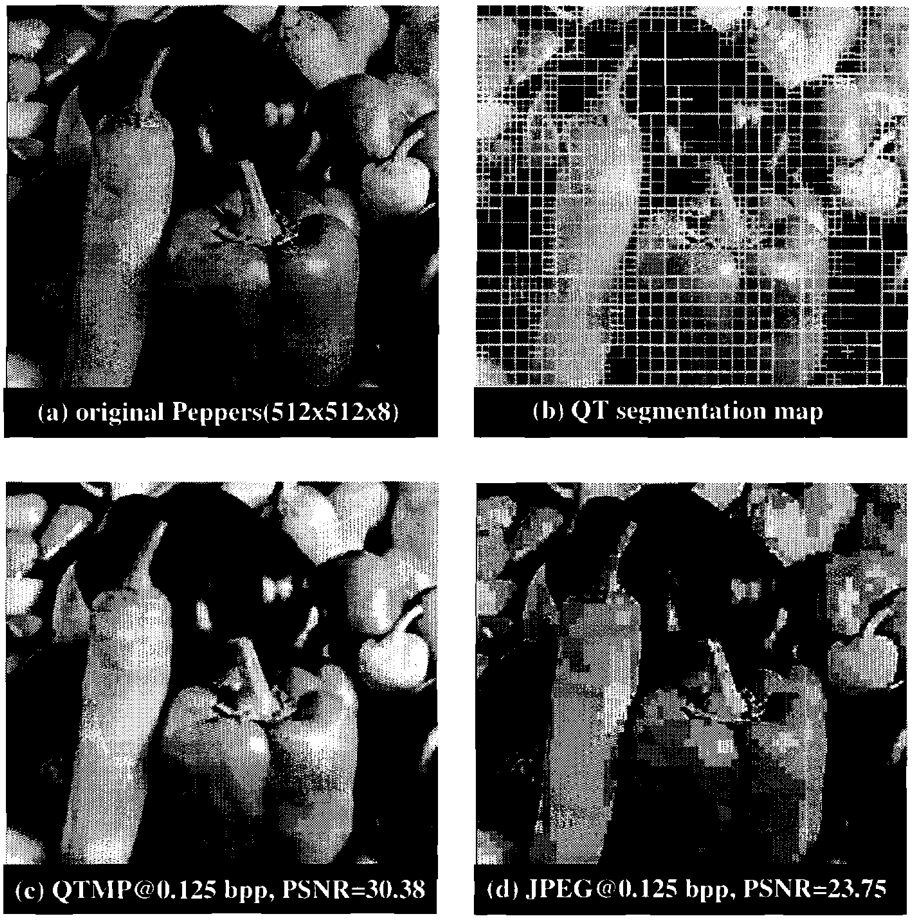
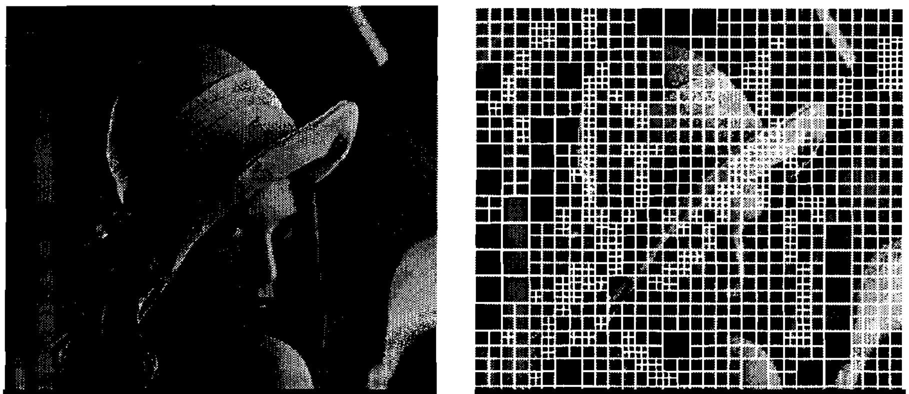
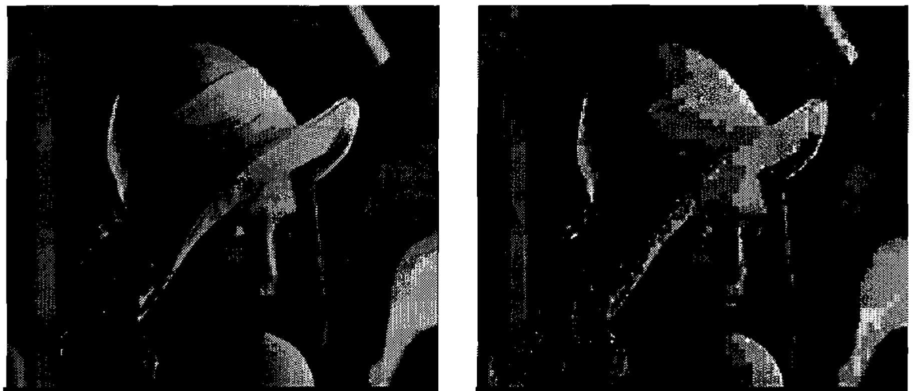

(a) original Peppers(512x512x8)
(b) QT segmentation map
(c) QTMP@0.125 bpp, PSNR=30.38
(d) JPEG@0.125 bpp, PSNR=23.75

Fig. 4.7. Comparison of QTMP & JPEG compression algorithms for the test image Peppers. (a) Original Peppers. (b) QT segmentation map of (a). (c) QTMP encoded at 0.125 bpp, PSNR = 30.38 dB (d) JPEG, encoded at 0.125 bpp, PSNR = 23.75 dB.



(a) original Lena(512x512x8)

(b) QT segmentation map



(c) QTMP@0.125 bpp, PSNR=30.25

(d) JPEG@0.125 bpp, PSNR=26.75

Fig. 4.8. Comparison of QTMP & JPEG compression algorithms for the test image Lena. (a) Original Lena. (b) QT segmentation map of (a). (c) QTMP encoded at 0.125 bpp, PSNR = 30.25 dB (d) JPEG, encoded at 0.125 bpp, PSNR = 26.75 dB.

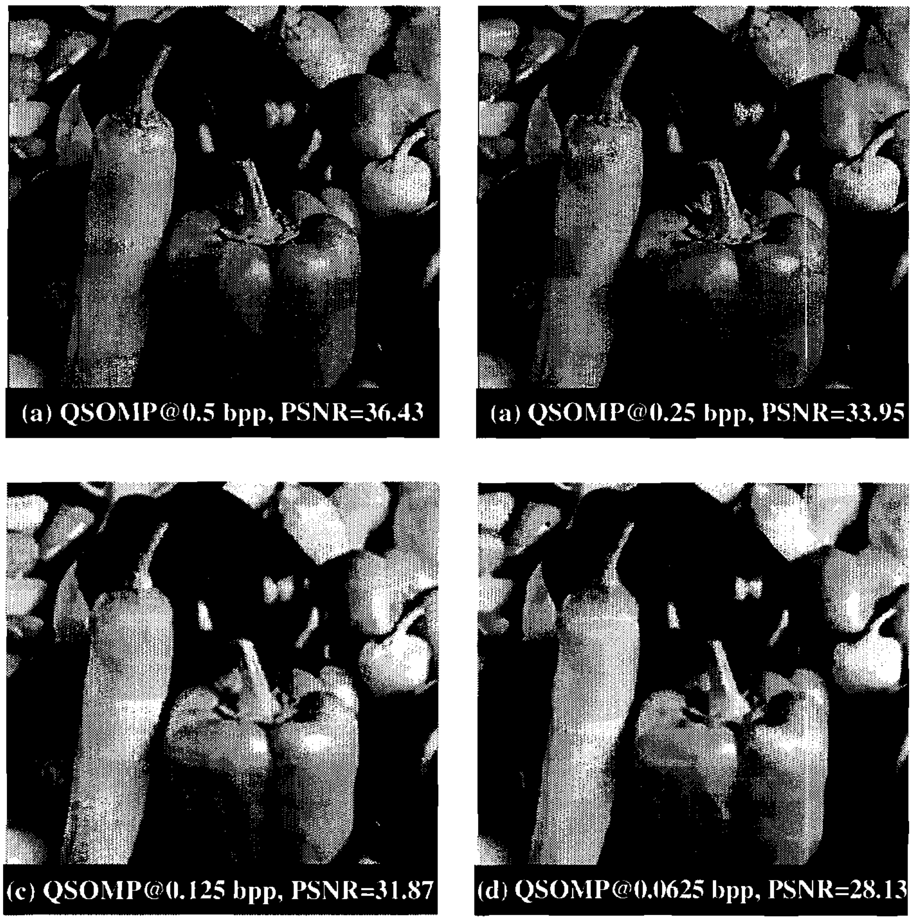

(a) QSOMP@0.5 bpp, PSNR=36.43
(a) QSOMP@0.25 bpp, PSNR=33.95
(c) QSOMP@0.125 bpp, PSNR=31.87
(d) QSOMP@0.0625 bpp, PSNR=28.13

Fig. 4.9. QSOMP compression results for the test image Peppers. (a) QSOMP encoded at 0.5 bpp, PSNR = 36.43 dB. (b) QSOMP encoded at 0.25 bpp, PSNR = 33.95 dB. (c) QSOMP encoded at 0.125 bpp, PSNR = 31.87 dB. (d) JPEG, encoded at 0.0625 bpp, PSNR = 28.13 dB.

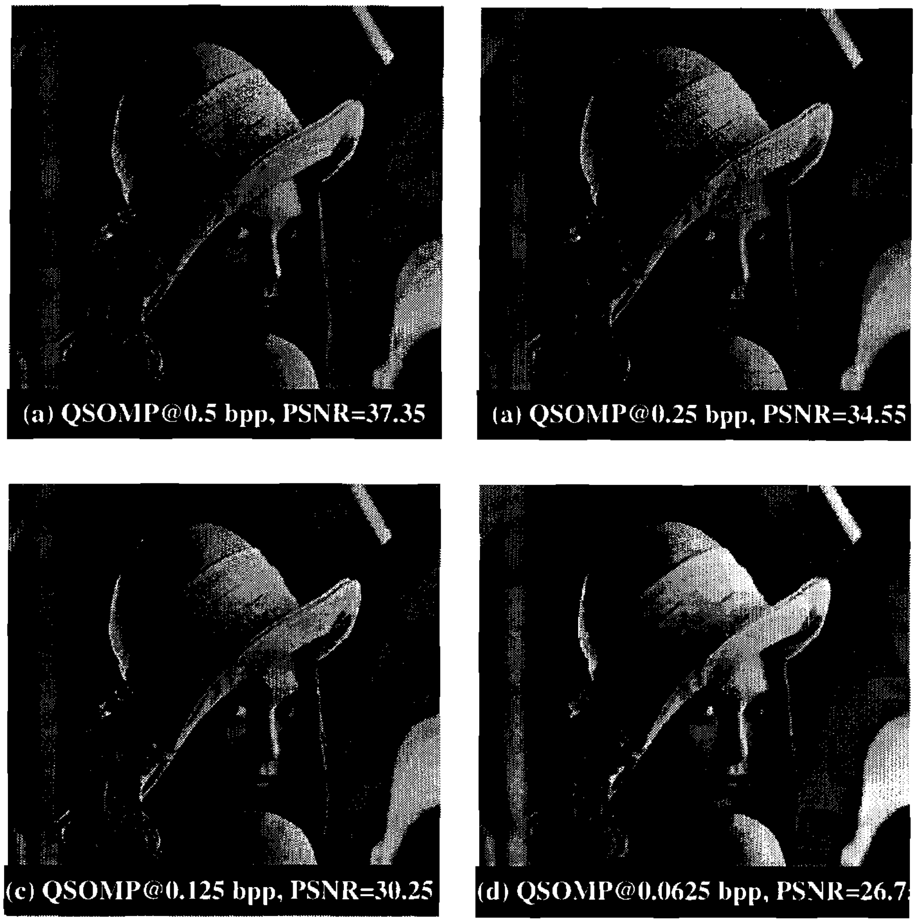

(a) QSOMP@0.5 bpp, PSNR=37.35
(a) QSOMP@0.25 bpp, PSNR=34.55
(c) QSOMP@0.125 bpp, PSNR=30.25
(d) QSOMP@0.0625 bpp, PSNR=26.7

Fig. 4.10. QSOMP compression results for the test image Peppers. (a) QSOMP encoded at 0.5 bpp, PSNR = 37.35 dB. (b) QSOMP encoded at 0.25 bpp, PSNR = 34.55 dB. (c) QSOMP encoded at 0.125 bpp, PSNR = 31.46 dB. (d) JPEG, encoded at 0.0625 bpp, PSNR = 27.92 dB.

Table 4.1
PSNR and number of atoms encoded by QSOMP compression algorithm for the test image Lena.

Bit-Rate (bpp)	**Compression Ratio**	**No. of Encoded Atoms**	**PSNR (dB)**
1	8:1	37546	40.50
0.5	16:1	17852	37.35
0.25	32:1	9172	34.55
0.125	64:1	4750	31.46
0.0625	128:1	2110	27.92

Table 4.2
PSNR's of five different wavelet compression techniques for the test image Lena.

Bit-Rate	**Compression Ratio**	EZW	IEZW	STE	**DTE**	**SFQ**
1	**8:1**	39.55	40.23	39.34	40.00	**40.52**
0.5	16:1	36.28	36.90	36.35	36.73	37.36
0.25	32:1	33.17	33.53	33.40	33.65	34.33
0.125	64:1	30.23	31.10	30.25	30.28	31.09
0.0625	128:1	27.54	27.72	27.52	27.56	27.62

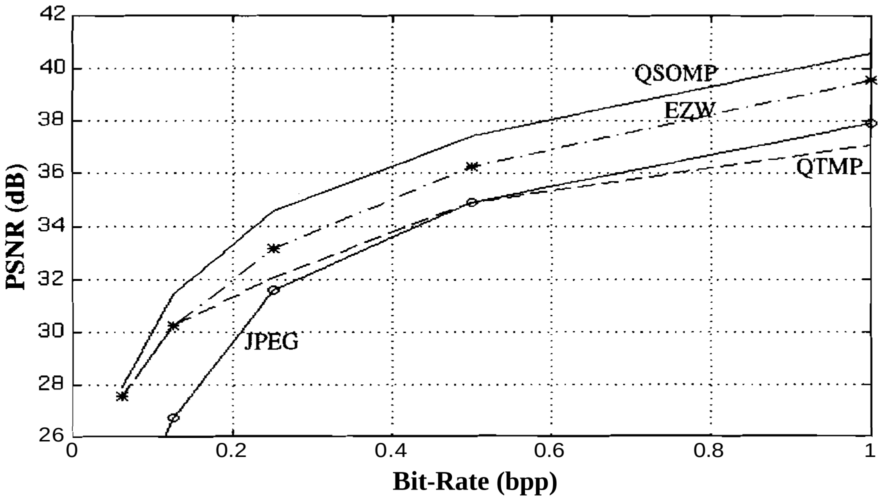


Fig. 4.11. Rate-distortion curves for the test image Lena and JPEG, EZW, QTMP, and QSOMP algorithm.

5. MOTION ESTIMATION AND COMPENSATION WITH QSOMP

5.1 Introduction

Due to the increased use of digital video in multimedia systems, in recent years, video compression has emerged as an area of intense research in recent years [2, 6, 7]. Video is a three dimensional signal consisting of a sequence of images. The individual images in a video signal are often called *frames*. Usually, video camera systems capture about thirty frames per second for a smooth motion to be perceived by the human visual system. Therefore the consecutive frames in a video clip are highly correlated except for instances in which a change of scene occurs. The presence of correlation structures related to motion in the successive video frames can be exploited to achieve high compression. In this case, a single frame, which is a two dimensional signal, can be coded as a *reference* frame by an image compression algorithm, and future frames can be predicted from the reference frame by a simple transformation such as translation [6, 7]. The video compression schemes that use this hybrid approach belong to the class of Motion Estimation and Motion Compensation (MEMC) video compression algorithms.

Although it is possible to extend two-dimensional image compression algorithms to video compression in three dimensional spaces [2], the hybrid MEMC techniques are widely used for compression of video signals because of their simplicity and good performance. For example, in emerging video compression standards such as H.261, MPEG, and the grand alliance HDTV broadcasting, a hybrid MEMC algorithm based on the concept of ***intraframe*** and ***interframe*** coding with Block-Based Discrete Cosine Transform is being used [1]. In the intraframe mode of operation, spatial redundancy is exploited by structuring a single frame into many small square blocks. These blocks are then DCT transformed, quantized, and coded. In the interframe mode of operation, the temporal redundancy between adjacent frames is exploited by using the MEMC algorithm to generate a prediction of the current video frame from previous (arid in the case of

MPEG possibly from future) frames. The difference between this prediction and the actual frame is then DCT transformed, quantized, and coded. The block diagram of hybrid motion-compensated transform-based coding algorithm is shown in Fig. 5.1.

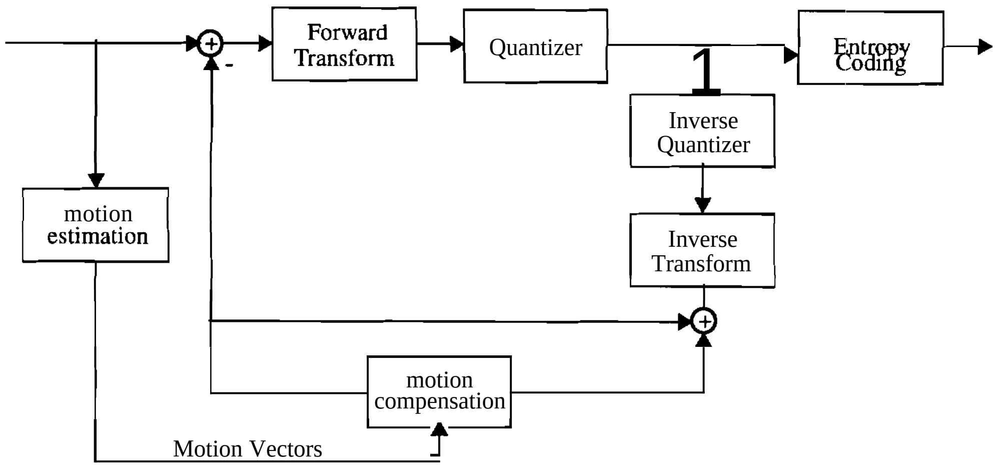


Fig. 5.1. Block diagram of hybrid motion-compensated predictive transform-based coding.

Finally, the intraframe and interframe information, as well as the motion vector, synchronization, and other side information is structured into a sequence of compressed bit streams for storage or transmission. Although a number of different techniques for implementing the MEMC algorithm have been proposed in the literature, the Block-Matching (BM) algorithm is the most widely used motion estimation technique in video compression applications [6, 7].

The main disadvantage of a hybrid BM-MEMC video compression algorithm is the fact that the processes of prediction and transform coding of the residuals are independent. As a result, the residuals can not be efficiently represented with the transform coefficients. More specifically, at lower bit rates where only a few number of transform coefficients are retained, the performance of the hybrid BM-MEMC video compression algorithms is not satisfactory. In this chapter, it is shown that the processes of prediction and residual coding can be unified by using a Quantized Segmented Matching Pursuit (QSOMP) MEMC algorithm. Experimental results show that this unified framework performs considerably better than the traditional hybrid BM-MEMC algorithm. The organization of this chapter is as follows. Section 5.2 provides a brief overview of the BM algorithm. Section 5.3 introduces the QSOMP-MEMC algorithm. Finally, section 5.4 presents experimental results and concluding remarks.

5.2 Block-Matching Motion Estimation

Block-Matching (BM) is the most widely used motion estimation technique in video compression algorithms. Its widespread use is due to its simplicity and relatively good performance [7]. With the assumption of a smooth translational motion, the BM algorithm searches for local correlation maximums between neighboring blocks in successive frames, as depicted in Fig. 5.2.

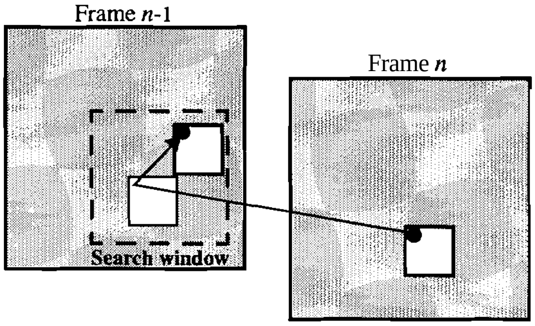


Fig. 5.2. Block-matching in the block-based motion estimation and compensation technique.

Let $z(k,l,n)$ represent the pixel values in a video sequence, where the variables k, I, and n represent the horizontal, vertical, and time dimensions, respectively. Given a block $\mathcal{B}(i, j, n) = [z(i, j, n), \ldots, z(i + N - 1, j + N - 1)]$, of NxN pixels from the current frame n, with the top left corner at (i, j), the best match for $\mathcal{B}$ should be searched for in frame n-1. The search is usually limited to an $(N+2m \times N+2m)$ region called the search window. Let $x_{k,l} = \mathcal{B}(kN, lN, n)$ denote the block that is to be coded in frame n, then the set of blocks in the search window is given by

$$\{\mathcal{B}(kN - i, lN - j, n - 1);\ \ i, j \in [-m, \ldots, m]\} \qquad (3.1)$$

In the first step of BM algorithm, a match for the current block should be found in the above set. BM algorithms usually differ in search strategies and matching criteria. Search strategies include the exhaustive search, three-step search, and cross search. The matching of blocks can be quantified by using different criteria such as Maximum Cross Correlation (MCC), maximum Matching Pel Count (MPL), minimum Mean Absolute Difference (MAD), and minimum Mean Square Error (MSE). The most widely used matching criterion is MSE. In this case, an estimate for the motion vector components, or the displacement vectors (i_m, j_m), is obtained by

$$(i_m, j_m) = \arg \min_{i,j} \|x_{k,l} - \mathcal{B}(kN - i, lN - j, n - 1)\| \tag{3.2}$$

and the corresponding residual signal is given by

$$\varepsilon_{k,l} = x_{k,l} - \mathcal{B}(kN - i_m, lN - j_m, n - 1) \tag{3.3}$$

Finally, in the second step of the algorithm the prediction error in (3.2) is transform coded and quantized. In the following section it is explained how these: two steps can be unified by using a modified matching pursuit algorithm.

The above procedure results in a single pixel accuracy motion estimation. It is possible to considerably improve the performance of this type of motion estimation by using a sub-pixel accuracy motion estimation algorithm [6, 7]. The sub-pixel accuracy motion estimation can be carried out in two steps. In the first step the previous frame is interpolated by a factor of two in both directions. The second step is similar to single-step motion estimation. That is, in the second step a current block is matched to a block in the previous interpolated frame inside a search window.

5.3 Block-Matching Motion Estimation with Matching Pursuit

The residuals obtained in (3.3) by using the MB-MEMC algorithm usually consist of high frequency components and can not be efficiently represented with a small number of DCT coefficients. Therefore, at rates below 64 Kbps, where only a few number of DCT coefficients are allowed, the resulting decoded video suffers from severe blockiness artifacts [1, 69]. In [69] matching pursuit with a dictionary of Gabor functions was used to encode the residuals. It was shown that at the cost of computational complexity, the matching pursuit encoding of the residues produces better results, both in terms of PSNR and subjective image quality.

In this algorithm, a different approach is taken by unifying the BM algorithm and the residual coding into a QSOMP framework. It is argued that the set of blocks in the search window of the previous frame can be considered as a good dictionary of atoms for the expansion of the current block with the greedy QSOMP algorithm. This argument holds if smooth translational motion is assumed, which is the case in the MB-MEMC algorithm. The QSOMP-MEMC algorithm is shown in Fig. 5.3.

```
Start {
Given a current block $x_{k,l} = \mathcal{B}(kN, lN, n)$ in frame n:
1. Interpolate the previousframe by a factor of 2 (use bilinear interpolation)
2. For  $i, j \in [-m, \ldots, m]$ {
                    Let $\varphi_{i,j} = \frac{\mathcal{B}(kN-i, lN-j, n-1)}{\|\mathcal{B}(kN-i, lN-j, n-1)\|}$,
                 };
3. Define the block dictionary $D_{\mathcal{B}} = \{\varphi_{k,l};\ i, j \in [-m, \ldots, m]);$
4. Define an auxiliary dictionary $D_A = \{m\ Daubechies\ D2\ \text{atoms}\};$
5. Iteration 1: use QOMP and $D_{\mathcal{B}}$;
6. Stop if energy of residue $< \varepsilon$;
7. Iteration 2 top: {
                    Use QSOMP and $D_A$;
                    Stop if energy of residue $< \varepsilon$;
               };
End.}
```

Fig. 5.3. The QSOMP-MEMC algorithm.

In step one, the interpolation of the previous frame is performed for sub-pixel accuracy. The increase in the computational complexity is justified by the small size of the dictionary. Furthermore, the computational complexity may be reduced by using a two step sub-pixel accuracy algorithm [6, 7]. In this method, a single pixel accuracy approximation is perfrmed over the nine nearest neighbors of the current block in the previous frame, and then the approximation is refined by performing a sub-pixel accuracy approximation on the neighborhood of the best match obtained by the single pixel accuracy.

The reason for using an auxiliary dictionary in step 4 is the fact that after the first iteration, the residues usually consist of high frequency components, and the atoms in the block dictionary are not adequate for representing the residues.

The experimental results showed that the motion vectors obtained with the QSOMP-MEMC algorithm are usually different than those obtained from the MB-MEMC algorithm, and the residual energy of the QSOMP-MEMC algorithm, after the first iteration, is upper bounded by the residual error in the MEMC algorithm given in equation (3.3).

In general, the value of the inner product $\langle \varphi_{i_1, j_1}, x_{k, l} \rangle$, at the first iteration of the QSOMP-MEMC algorithm should be transmitted to the decoder. However, this value can be predicted from its corresponding block in the previous frame [6] as

$$\langle \varphi_{i_1, j_1}, x_{k, l} \rangle = \alpha cos(\theta) \| \mathcal{B}(kN - i_1, lN - j_1, n - 1) \| \tag{3.4}$$

where a is a scaling factor representing the illumination change in successive frames, and θ is the angle between the current and the matching block. Since the illumination is relatively constant between successive frames, and the angle θ is normally small, the above equation reduces to

$$\langle \varphi_{i_1, j_1}, x_{k, l} \rangle = \| \mathcal{B}(kN - i_1, lN - j_1, n - 1) \| \tag{3.5}$$

The decoder in the QSOMP-MEMC algorithm should perform more computations than the BM-MEMC decoder, because it needs to compute the required inner product operations in order to reconstruct the encoded frame. The experimental results are provided in section 5.4.

5.4 Experimental Results and Conclusions

Simulations were carried out on the ***Salesman*** and ***Claire*** video sequences. The test sequences were all QCIF (144 x 176) and 100 frames long. A single frame of each sequence is shown in Fig. 5.4. For the BM-MEMC algorithm, 6 and 12 DCT coefficients were used, with an exhaustive search with the MSE as the matching criterion, and the motion estimation was performed with sub-pixel accuracy using bilinear interpolation. The performance of the coding was measured by computing the first order entropy of quantized

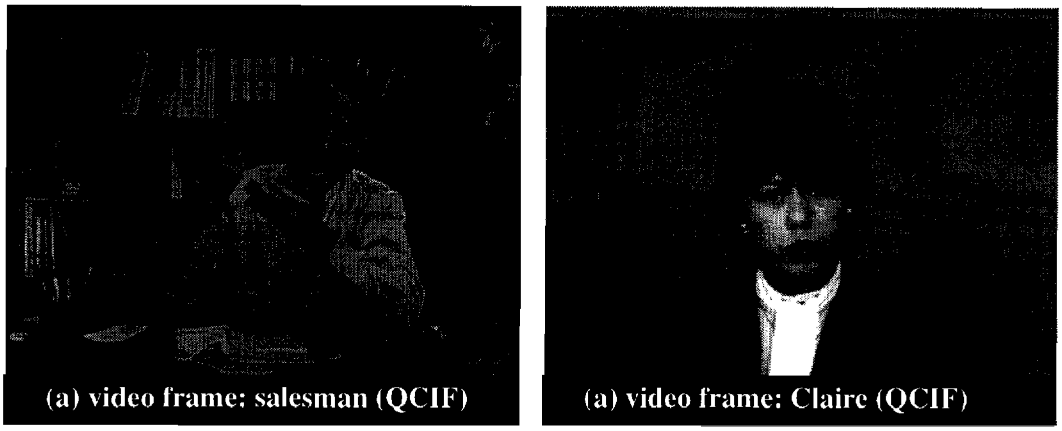
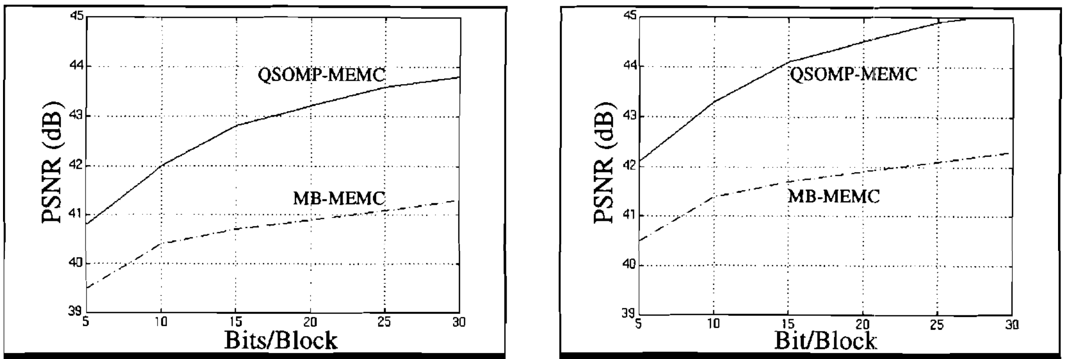

(a) video frame: salesman (QCIF)
(a) video frame: Claire (QCIF)

Fig. 5.4. Frames from QCIF test video sequences. (a) Salesman. (b) Claire.

coefficients. The bit rates were computed as the average number of bits required to encode a single block of data, and the block size was 8x8. The results for the MB-MEMC algorithm with 6 and 12 DCT coefficients are shown in Fig. 5.5.

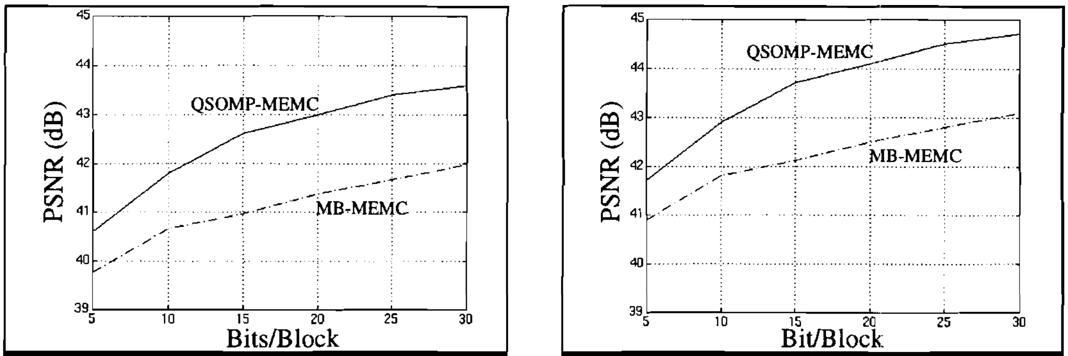
For the QSOMP-MEMC algorithm an optimal scalar quantizer was used in a rate distortion sense (see chapter4), and the approximations were performed with two step sub-pixel accuracy. The bit rates and performance were measured in the same manner. The results for the QSOMP-MEMC algorithm with 6 and 12 Daubechies orthonormal atoms are shown in Fig. 5.5.

In both cases, the associated motion field information was not included in the bit rate computations, because these rates are equal for both methods. In both cases, the prediction of the current frame was based on the uncoded previous frame. In order to compare the results for low bit rate video coding applications, only the performance for 5 bits per block to 30 bits per block was considered. The results illustrated in Fig. 5.5 indicate that at these low bit rates the QSOMP-MEMC algorithm performs considerably better than the traditional BM-MEMC algorithm. However, the decoder for the QSOMP-MEMC algorithm is more complex, because it has to compute the required inner products to reconstruct the encoded image. The experimental results, also showed that at higher bit rates, there is no advantage in using the QSOMP-MEMC algorithm.



(a) Salesman with 6 coefficients

(b) Salesman with 12 coefficients



(c) Claire with 6 coefficients

(d) Claire with 12 coefficients

Fig. 5.5. Rate-Distortion curves for QSOMP-MEMC and MB-MEMC algorithm. (a) Salesman with 6 coefficients. (b) Salesman with 12 coefficients. (c) Claire with 6 coefficients. (d) Claire with 12 coefficients.

However, due to the small size of the dictionary, the computation of the motion field in QSOMP-MEMC is not very costly. Given a search window of size *MxM*, and a block size of *N*x*N*, and assuming that there are L one-dimensional atoms of length N in the dictionary, the required number of multiply-accumulation operations to search for an atom in the dictionary can be obtained by $L^2M^2N + LMN(M + 2N)$. For L = 6, M = 9, and N = 8, 32400 operations have to be performed per atom.

6. PRE-PROCESSING AND POST-PROCESSING OF IMAGES AND IMAGE SEQUENCES

6.1 Introduction

Visual information is often subject to different kinds of degradations. These degradations may be in the form of additive or multiplicative sensor :noise, blur due to camera misfocus, ***blockiness*** or ***motion jerkiness*** due to compression at lower bit rates with block based motion compensated techniques, or errors due to faulty communication channels. Therefore, pre-processing and post-processing units for digital image filtering are an essential part of any integrated imaging or video system that uses an intensity image as an input. These kinds of processing are normally multiple criteria optimization problems that may involve restoration, enhancement, or just a suitable representation of the data. While for still images only spatial processing is required, for image sequences both spatial and temporal processing are needed.

In modem multimedia communication systems, digital filters are often used for the processing of data, voice, image, and video streams. In fact, they are an essential part of pre or post-processing modules in these systems. In general, a digital filter for the processing of visual information should be capable of satisfying one or more of the following requirements:

- Restoring the original image from its noisy version (smoothing).
- Enhancing certain features (edges) of the degraded image (sharpening).
- Preserving the information bearing details of the image (detail preservation).
- Feasibility of implementation in real time (computational efficiency).

Most of the traditional methods of image restoration and enhancement are linear and assume an additive Gaussian noise model for the data [70, 71, 72]. These statistical procedures are optimal under exact models of noise distribution, but are (generallyunstable under small deviations from these models. Moreover, they can not fully exploit the nonlinearity of image formation models and the human visual system [73]. Mean filter is the most well known linear filter. It achieves noise reduction by averaging over the neighborhood of pixels. However, if the noise distribution is long-tailed or impulsive, the result is not satisfactory. Another disadvantage of the mean filter is that it tends to blur the edges, and often eliminates fine details of the image. Therefore, the mean filter may not be useful as a front end operator in image or video processing systems [72]. These disadvantages have led some researchers to study the use of nonlinear filters as an alternative [73]. Nonlinear filtering techniques for signal and image processing emerged as early as 1958 [74] and have had a dynamic development in the last few decades.

Order Statistics (OS) filters are one of the most important families of nonlinear image filters [73, 75]. These filters have shown excellent robustness properties in the presence of impulsive noise while preserving the important information bearing features of the image. The majority of recent work in nonlinear order statistic filters has focused on smoothing and preserving the details of digital images. Only a few authors have designed filters with edge sharpening properties [76, 77].

In this chapter, the single and multi-stage implementations of a new robust nonlinear filter based on the theory of Generalized Maximum Likelihood estimation and Order Statistics (GMLOS) are presented [12]. This new class of filters is not only capable of smoothing the noise and preserving the details, but also has the ability to sharpen edges. The GMLOS filter can be used in a variety of algorithms for the processing of visual information [12, 13, 15]. This chapter presents the theory, implementation, and application of the GMLOS filter to noise smoothing. The following chapter presents a novel algorithm that uses the GMLOS filter for the concealment of errors due to packet loss in encoded image and video streams [15].

The organization of this chapter is as follows. A summary of the previous work on nonlinear filters is presented in Section 6.2. Section 6.3 is devoted to the theory, implementation, and some applications of the GMLOS filter. A brief comparative study of different edge-enhancing nonlinear filters with the GMLOS filter is also provided in this section. Finally, the experimental results and the concluding remarks are presented in Section 6.4.

6.2,Nonlinear Filters

Homomorphic filters are one of the oldest classes of digital nonlinear filters [73, 78]. They satisfy the generalized principle of superposition and have found applications in seismic signal processing, digital speech processing, and ultrasonic imaging. Nonlinear mean filters can be considered to be special cases of homomorphic systems [79]. They can be defined as a general nonlinear function of the weighted average of the neighboring gray values of a pixel. They are better than median filters in smoothing the additive Gaussian or uniform noise. They are also better than mean filters in suppressing impulsive noise and preserving edges. However, they are only capable of removing either positive or negative spikes but not both at the same time.

Polynomial filters belong to the class of nonlinear filters that are based on the Volterra series representation [80]. Although the classes of nonlinear systems that can be represented by Volterra series are limited, a subclass of polynomial filters, known as quadratic filters, has been used for image enhancement, edge detection, and nonlinear interpolation of image sequences [73].

Morphological filters belong to the class of nonlinear filters that have originated from shape analysis and set theory in mathematics [81, 82]. The opening and closing filters [75], which are formed by various combinations of erosion and dilation operators in set theory, have shown to be excellent in preserving details and edges. However, they can not effectively suppress a high percentage of impulsive noise.

Many classes of nonlinear filters have been developed based on the theory of robust statistics [83, 84]. These filters fall mainly into one of the three categories of M-filters, R-*filters*, and L-filters. The M-estimators[1] were proposed by Huber [84] as a generalization of maximum likelihood estimators. The M-estimators of location have been used as nonlinear image processing filters in STM [85] and adaptive mean filter [86]. R-estimators have been proposed by Hodges and Lehmann [87]. They have been developed based on the concept of rank estimate in statistical theory. Therefore, the output of an R-filter is determined by the relative ranks of the data instead of the actual values. Examples of R-

---

[1] The terms filters and estimators have been used interchangeably in this chapter.

filters include the Wilcoxon [88], LDW [73], and FMH filters [89]. The L-estimators [73, 84] are defined as fixed linear combinations of order statistics. Some examples of this type of filters are the a-Trimmed Mean (a-TM) [73], MTM [85], and K-Nearest Neighbor (KNN) filters [90].

L-filters and *R-filters* are related to filters based on order statistics [91]. Order statistics filters are one of the most important families of nonlinear image filters, and have been shown to posses excellent robustness properties in the presence of impulsive noise while preserving the important information bearing features of the image [75]. The median filter is the most popular order statistics filter. It was first introduced by Tukey [92] as a smoothing device for discrete signals. It is a nonlinear technique in which a given pixel in the image is replaced by the sample median of its neighbors. It does not posses the drawbacks of mean filters and can effectively eliminate the impulsive noise while preserving the edge information. However, it also preserves any monotonic degradation of the edge and therefore is not capable of enhancing blurred or ramp edges. In addition, it often eliminates or disrupts fine details such as thin lines or small objects in the image. A comprehensive analysis of median filter characteristics can be found in [93, 94, 95].

More recently, there has been a growing interest in generalizing median filters by using a combination of different order-statistics. Bovik, Huang, and Munson [96] have used a weighted linear combination of order-statistics of the input sequence. The weights are chosen to minimize the output mean-square error. This filter combines the properties of both averaging and median filters. Bernstein [97] has introduced the concept of the signal adaptive median filter. It uses a variable size window and is capable of simiultaneously removing a combination of signal dependent additive and random impulsive noise. Arce and Foster [98] have provided an extensive analysis of multi-stage median filters. These filters are constructed by combining the output of basic subfilters that are designed to preserve edges or lines in the image. They have shown that multi-stage median filters have the same impulse rejection properties as ordinary median filters but performs better in preserving details.

Some researches have tried to develop a unified theoretical framework for the analysis and design of nonlinear filters. Longbotham and Bovik [99] have used the relationships between order-statistic and linear FIR filters to develop a firm theoretical foundation for order-statistic filters. Coyle, Lin, and Gabbouj [100] showed that stack filters, which are defined by a weak superposition property and an ordering property, contain all compositions of the 2-D ranked order operations. Finally, Maragos and Schafer [101] have explored the relationships between the morphological, order-statistic, and stack filters.

Most of the recent work in nonlinear filtering of digital images has focused on smoothing the noise while preserving details. Only a few authors have designed filters with edge sharpening properties [77]. Edges are one of the most important features of an image in many image analysis and computer vision applications and have a great impact on human visual perception [71]. The Comparison and Selection (CS) filter [102], Lower-Upper-Middle (LUM) filter [38], and Weighted Majority of m values with Minimum Range ($WMMR^m$) filter [104] have been shown to be effective in smoothing the noise while sharpening the edges.

6.3 The GMLOS Filter

It is assumed that members of the degraded input data set $W=\{z_i: i=1,...,n\}$, obey an additive model

$$z_i = x_i + v_i \tag{6.1}$$

where the original data x_i and the noise process v, are statistically independent. In this development, an approximate parametric model has been used for the data [84]. Parametric data analysis is one of the major approaches to the analysis of information in statistical theory. In this approach, one has to make an assumption about the model underlying the data set W. Such a model is usually the probability distribution $p(z;\theta)$, where 8 is a parameter vector of dimension m, taking values in the Cartesian product space $\Theta = \Theta_1 \times \Theta_2 \times ... \times \Theta_m$. As an example, considering a uni-variate Gaussian model for the data, the parameter vector can be defined as $\theta' = [\mu\ \delta]$ where μ and 6 represent the mean and standard deviation of the Gaussian distribution, respectively. The goal of parametric analysis is to find an efficient and consistent estimate of θ. The estimation of μ corresponds to the problem of location estimation, and the estimation of 6 corresponds to *scale* estimation. In this work, the main concern is the estimation of the location parameter in which the model has the form $p(z;\theta) = p_1(z-\theta)$, where θ is a real scalar (location parameter). The objective is to estimate θ from the contaminated data set W.

The Maximum Likelihood Estimator (MLE) is the most widely used estimator for parametric data analysis [84, 87]. For example, the MLE of a Gaussian model is the sample mean and the MLE of a Laplacian model is the sample median. The major problem in classical parametric data analysis via MLE is the fact that the exact probability distribution of the data must be known *a priori*. In practice this is rarely the case, due to the following

- The existence of gross errors or outliers.
- The presence of truncation and rounding errors.
- The probability distribution model being only an approximation of the real model.

The most severe deviation from the underlying model of the data is caused by the presence of outliers in the observed data [84]. The precise definition of an outlier set is given below.

Definition 6.1 *Let* W *be a data set with i.i.d members* z_i, *obeying the density* $p(z|x;\phi)$, *where* ϕ *is an unknown location parameter. Let* $\hat{\phi}_{ML}$ *be the MLE of* ϕ, *computed solely from* W, *and consider the set* W^o *with members* z'_i . *Then the set* W^o *is said to be an outlier set with respect to* W *if the following inequality holds*

$$\left|\{min\, \ln p(z|x;\hat{\phi}_{ML});z\in W\} - \{min\, \ln p(z'|x';\hat{\phi}_{ML});z'\in W^o\}\right| > \gamma \qquad (6.2)$$

where γ *is equal to* $\hat{\phi}_{ML}/\mu$, *and* μ *is the sample mean of the set* W.

It is well known that even a small deviation from the assumed model may cause the classical estimators, such as MLE, to produce unreliable results [84]. During the last few decades, formal theories, known as *robust* statistics have been developed to cope with these deviations from the underlying model. In fact, robust estimation is a branch of estimation theory that deals with approximate parametric models for data [83, 84]. In this thesis, an approximate model has been used for the input data to construct the GMLOS filter based on the concept of generalized maximum likelihood estimation, by taking advantage of the order statistics for outlier detection.

Given a set of n mutually independent data points $W=\{z_i$: $i=1,...,n)$ within the processing window, it is assumed that a fraction (1-ε) of them (the *inlier* set W^I) obey the density $p_1(z|\theta)$, and the remaining fraction ε (the *outlier* set Wo) obey the density $p_2(z|\phi)$, where $0 < \varepsilon < 0.5$. Therefore the approximate model for the data is given by

$$p(z) = (1-\varepsilon)p_1(z|\theta) + \varepsilon p_2(z|\phi) \tag{6.3}$$

Clearly, both W^I and W^O are subsets of W. The objective is to recover the expected value of the inliers based only on the contaminated data. That is, the data set W should be partitioned into 2 mutually exclusive sets W^O and W^I, according to the following definition.

Definition 6.2 *A partition $\{W^I, W^O\}$ of the set W,* Wo=W-W^I, *is said to be valid if $\#W^I > \#W^O$ ('#' denotes the cardinality of a set), where W^O is an outlier set with respect to the inlier set W^I according to the Definition 6.1.*

In general, the mean of outliers can be quite different from that of inliers, and the variance of outliers is much greater than that of the inliers, even though clustered outliers are not uncommon [83]. Consequently, the likelihood associated with a single inlier measured by its probability density function is much greater than the likelihood of an outlier computed by the same type of density function. Thus a subset of L observations consisting of only inliers is expected to have a greater likelihood than a subset of L observations having both inliers and outliers or one having mostly outliers. Assuming that the number of inliers is equal to L, then the optimum inlier set, W^I, can be chosen as the set having the highest likelihood among all subsets of W with size L. Assuming that a Gaussian distribution the members of the inlier set W^I, after omitting some nonessential terms, the negative log likelihood expression is

$$J(W^I,\theta) = \sum_{z_i \in W^I} (\theta - z_i)^2 \tag{6.4}$$

if the members of W^I are known, then the value of 0 which minimizes $J(W^I,\theta)$ is

$$\theta_M = \frac{1}{L} \sum_{z_i \in W^I} z_i \tag{6.5}$$

and the new criterion function $J_1(W^I) = \min_{\theta} J(W^I,\theta)$ can be written as

$$J_1(W^I) = \sum_{z_i \in W^I} z_i^2 - \frac{1}{L}\Big(\sum_{z_i \in W^I} z_i\Big)^2 \tag{6.6}$$

$J_1(W^I)$ should be minimized with respect to W^I over all subsets of W with size L. It may appear that such a procedure may not be computationally feasible, since the number of such subsets is $n!/[L!(n-L)!]$. However, for uni-variate data and with inliers obeying an exponential family density function, the optimal subset of size L of candidate inliers can be obtained by comparing only $n-L+1$ contiguous subsets of the rank ordered data according to the following definition and theorems.

Definition 6.3 *A subset W^I of the data set $W=\{z_i$: $i=1,...,n\}$ is said to be contiguous if for every pair of members z_i and z_j of W^I, where $z_i < z_j$ the existence of a member z_k of W, with $z_i < z_k < z_j$ implies $z_k \in W^I$.*

Theorem 6.1 *Given a subset W^{NC} of W with size L, which is not contiguous, there always exists a contiguous subset W^C of W with size L, such that $J(W^C)$ I $J(W^{NC})$.*

Proof: The proof is given in the *Appendix.*

The following theorem follows directly from *Theorem* 6.1 and equation (6.6).

Theorem 6.2 *Let $W^R=\{z_{(1)},z_{(2)},...,z_{(n)}\}$ be the ordering of the data set $\{z_i$: $i=1,...,n\}$, such that $z_{(1)} < z_{(2)} < \ldots < z_{(n)}$, then W_L^I, the global minimum of $J_I(W^I)$, has the following structure*

$$W_L^I = \{z_{(i)}; M_L \le i \le M_L + L - 1\} \tag{6.7}$$

where M_L is given by

$$M_L = \arg\min_M \left\{ \sum_i z_{(i)}^2 - \frac{1}{L}\left(\sum_i z_{(i)}\right)^2 \right\} \tag{6.8}$$

and the index i runs from M_L to M_L+L-1.

Therefore, the criterion function J can be refined as[2]

$$J(\theta,M) = \sum_{i=M}^{M+L-1} (\theta - z_{(i)})^2 \tag{6.9}$$

J has to be minimized with respect to both **0** and M, where the optimal values of **0** and M correspond to the output of the filter and the starting point of the contiguous and ordered inlier data set of length L, respectively. This optimization problem can be carried out in

---

[2] *For simplicity, the index L has been dropped in later developments.*

two steps.

- ***Step 1:*** assume the value of M is known, then θ_M, the optimal value of 8 can be found from

$$\theta_M = \arg(\min_{\theta} \mathrm{J}(\theta, M) | M) \tag{6.10}$$

this is the well known least squares criterion function [84], and hence for $\mathrm{I} \le \mathrm{M} \le n - L + 1$,

$$\theta_M = \frac{1}{L} \sum_{i=M}^{M+L-1} z_{(i)} \tag{6.11}$$

- ***Step 2:*** Assume 8, is given, and define b, as

$$\mathrm{b}_M = \sum_{i=M}^{M+L-1} (\theta_M - z_{(i)})^2 \tag{6.12}$$

Now (6.9) has to be minimized with respect to M. b_M will be computecl for i=1,...,n-L+l and.the M for which b, has the minimum value will be found. This optimization could also be carried out recursively in two simple steps.

- ***Step 1***: Compute 8, and b, by using equations (6.6) and (6.7).
- ***Step*** 2: Compute θ_{M+1} and b_{M+1} from 8, and b_M, recursively as

$$\theta_{M+1} = \theta_M + \frac{1}{L}(z_{(M+L)} - z_{(M)}) \tag{6.13}$$

Now, define c, as

$$\mathrm{c}_M = \sum_{i=M}^{M+L+1} z_{(i)}^2 \tag{6.14}$$

then

$$\mathrm{c}_{M+1} = \mathrm{c}_M + (z_{(M+L)}^2 - z_{(M)}^2) \tag{6.15}$$

and

$$\mathrm{b}_{M+1} = \mathrm{c}_{M+1} - (L\theta_{M+1}^2) \tag{6.16}$$

Equations (6.13), (6.15), and (6.16) can be used for M=1,..., n-L to find the optimal value of the parameter M, where the length of the filter L can take any value between 1 and n.

It is straightforward to construct a weighted GMLOS filter by assuming unequal weights for the members of the inlier set. The rationale for introducing the weighting factors into the error criterion is to emphasize the contribution of those: samples that are considered to be more reliable. In this case, the criterion function in (6.9) can be written as

$$\mathbf{J}(\theta, M) = \sum_{i=M}^{M+L-1} (\theta - \omega_i z_{(i)})^2 \tag{6.17}$$

and the output of the Weighted GMLOS (WGMLOS) filter can be computed as

$$\theta_M = \frac{1}{S} \sum_{i=M}^{M+L-1} \omega_i z_{(i)} \tag{6.18}$$

where

$$S = \sum_{i=M}^{M+L-1} \omega_i \tag{6.19}$$

and. ω_i represents the weight associated with the ith sample of the inlier set. Assuming that the members of the inlier set are degraded with a zero mean and uncorrelated additive noise process with variance δ_n^2 , then

$$\omega_i = \frac{1}{\delta_n^2} \tag{6.20}$$

and θ_M represents the Best Linear Unbiased Estimator (BLUE) of the location parameter 8.

6.3.1 Properties of the GMLOS Filter

Among the various properties of the GMLOS filter, particular attention will be paid to the ones that are crucial to the processing of digital images. Although the following properties of the GMLOS filter hold even if unequal weights are used, for simplicity throughout this section equal weights are assumed for all members of the inlier set W^I (*i.e.* $\omega_i = 1$ for all i in [M, $M+L$-1]), throughout this section.

Definition 6.4 A *filter T operating on an input sequence* $\{z_i : i=1,\ldots,n\}$*, is said to be location equivariant if*

$$\mathbf{T}(z_1 + c, \ldots, z_n + c) = \mathbf{T}(z_1, \ldots, z_n) + c, \forall c \in \boldsymbol{R} \tag{6.21}$$

where R denotes the set of real numbers.

Property 6.1 ***The GMLOS filter is location equivariant.***

Proof: This property is implicit in the notion of a location estimator. Clearly, adding a constant to all the samples of an order set does not change the rank of its elements. Therefore, the optimizing value of ***M*** as obtained from (6.13), (6.15), and (6.16) will not change. From equations (6.11) and (6.17)

$$\begin{aligned} \mathbf{T}(z_1+c,\ldots,z_n+c) &= \frac{1}{L}\sum_{i=M}^{M+L-1}(z_{(i)}+\mathrm{c}) = \frac{1}{L}\left(\sum_{i=M}^{M+L-1} z_{(i)} + \sum_{\mathrm{i=M}}^{M+L-1}\mathrm{c}\right) \\ &= \frac{1}{L}\sum_{i=M}^{M+L-1} z_{(i)} + \frac{\mathrm{c}(M+L-1-M+1)}{L} \\ &= \mathbf{T}(z_1,\ldots,z_n)+c \end{aligned}$$

□

Definition 6.5 *A* ***filter*** **T** ***operating on an input sequence {z***

$$\mathbf{T}(cz_1,\ldots,cz_n) = c\mathbf{T}(z_1,\ldots,z_n),\ \forall c \in \boldsymbol{R} \tag{6.22}$$

Property 2 ***The GMLOS filter is scale equivariant.***

Proof: Clearly, if $\mathrm{c} \geq 0$, multiplying samples of an order set by c does not change the ranks of its elements and (6.18) follows from (6.10). If c < 0, the ranks of the order set would be reversed. However, the optimum value of ***M*** would not be changed. This is true because the objective function as given in (6.11) remains unchanged since $(\theta_M - z_{(i)})^2 \equiv (-\theta_M + z_{(i)})^2$. Similarly, (6.18) follows from (6.11). □

Definition 6.6 ***The breakdown point of an estimator may be defined as the smallest percentage of gross errors (or equivalently 100ε, where ε is defined in (6.3)), that may cause the estimator to take on arbitrary large values [105].***

For example, if only one of the samples in the data set tends to infinity, the mean of the samples will also approaches to infinity. This implies that the mean filter has a breakdown point of 0% and hence it is not robust. In the other hand, the median filter is robust and has a breakdown point of 50%. Obviously, the upper bound for the breakdown point is 50%, since beyond 50% differentiating between the good data (set of inliers) and the gross errors (set of outliers) is not possible.

Property 6.3 The *GMLOS filter* has a breakdown point of 50%, if $L = \lfloor n/2 \rfloor$.

Proof: Recall the distribution function for the approximate model given in (6.3). It will now be shown that the response of the GMLOS filter given by (6.11) is finite, even if ϕ approaches infinity, as long as ε is less than 0.5. In other words, suppose that the input data set $W = \{z_i : i=1,\ldots,n\}$ has $\lfloor n/2 \rfloor$ data points that can take arbitrarily large values, then the estimated value of θ given by this method is finite.

Consider the worst case when the outlier set is consist of impulses. In this case $p_2(z|\phi) = \delta(z-\phi)$, where ϕ is a fixed and positive number, and $\delta(\bullet)$ is the usual delta function. Assuming a Gaussian density for $p_1(z|\theta)$, the objective function given by (6.9) can be written as

$$J(\theta,M) = \begin{cases} J_1(\theta,M) \ \ if\ \mu > \beta \\ J_2(\theta,M) \ \ if\ \mu \leq \beta \end{cases}$$

where J_1 and J_2 are given by

$$J_1(\theta,M) = \int_M^{\beta} (1-\varepsilon)(\theta - z)^2 p_1(z)dz$$

and

$$J_2(\theta,M) = \int_M^{\beta} (1-\varepsilon)(\theta - z)^2 p_1(z)dz + \int_M^{\beta} \varepsilon(\theta - z)^2 \delta(z-\phi)dz$$

applying the sifting property of delta functions to $J_2(\theta, M)$ yields

$$J_2(\theta,M) = \int_M^{\beta} (1-\varepsilon)(\theta - z)^2 p_1(z)dz + \varepsilon(\theta - \phi)^2$$

where the parameter M is related to the parameter β by

$$\int_M^{\beta} p(z)dz = 0.5$$

Now, let $(\theta_1, M_1) = arg \min_{M,\theta} J_1(\theta, M)$ and $(\theta_2, M_2) = \arg \min_{M,\theta} J_2(\theta, M)$. Then from the expressions for $J_1(\theta,M)$ and $J_2(\theta,M)$ given above, and the Gaussian assumption for $p_1(z|\theta)$, it is clear that as long as $\varepsilon < 0.5$, $J_1(\theta,M)$ is finite and $J_1(\theta,M) < J_2(\theta,M)$, even if $\phi \to \infty$. This implies that the output of the GMLOS filter is finite with a breakdown point of 50%. □

Ideal step and ramp edges are frequently present in synthetic and real images. Their definitions are given below.

Definition 6.7 *An ideal one dimensional (1-D) step edge with height h consists of two adjacent constant segments that differ in magnitude by an integer h (see Fig. 6.1(a)).*

Definition 6.8 *A ramp edge of degree d consists of two unequal constant segments* S_1 *and* S_2 *with sizes M and N, which are separated by a monotonic sequence* S_3 *of length d, where* $0 < d < \min(M,N)$. *The monotonic sequence* S_3 *is bounded by the constant segments* S_1 *and* S_2. *A 1-D ramp edge of degree* $d = 2$ *is shown in Fig. 6.1(b).*

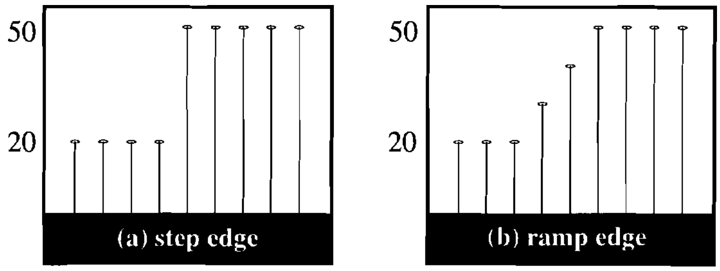


Fig. 6.1. Ideal step and ramp edges. (a) 1-D ideal step edge, h=30. (b) 1-D ramp edge, d=2.

Property 6.4 *The GMLOS filter preserves the shape of an ideal 1-D step edge*[3].

Proof: Assume a monotonically non-decreasing ideal step edge of height h and length N that consists of two constant segments S_1 and S_2, with values g and $g+h$, respectively. Consider a 1-D processing window W of size n as illustrated in Fig. 6.2. If all members of the set W belong to only one of the constant segments S_1 or S_2 (i.e. if W is entirely in a homogeneous region), then the L members of the inlier set W^I also belong to that constant segment (since $W^I \subset W$ by definition). Therefore, without loss of generality assuming that all members of W^I belong to S_1, then the output of the GMLOS filter as given by (6.11) can be computed as

$$\theta_M = \frac{1}{L}\sum_{z_{(i)} \in W^I} z_{(i)} = \frac{1}{L}\sum_{z_{(i)} \in W^I} g = g$$

which leads to the correct classification (value) of the data at the center of the processing window.

---

[3] ***For simplicity the proofs of properties 6.4 and 6.5 are given for 1-D edges. The extension to 2-D edges is straight forward.***

When the members of the set W belong to both S, and S_2 (i.e. when W falls somewhere across the two segments), it has to be shown that the GMLOS correctly classifies the edge. Without loss of generality, assume that the center cell of the processing window is an edge pixel that belongs to S,, as depicted in Fig. 6.2.

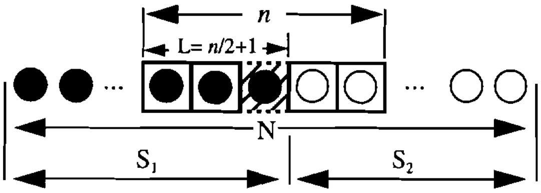


Fig. 6.2. A 1-D monotonically non-decreasing step edge (the members of the set W are shown inside the square boxes).

Since the length of the window is an odd number, and from the monotonicity of the edge sequence, it follows that the majority of the pixels in the set W are members of S,. According to ***Theorem*** 6.1, the set of inliers, W^I, is a contiguous set of size L such that $J(W^I) < J(W^A)$, where W^A is any subset of W with size L. Due to the contiguity requirements, and the above observations, and from (6.13), (6.15) and (6.16), it follows that the inlier set W^I is a minimizer of (6.6) if and only if all its members belong to only one: of the segments (e.g. S, in this case). This means that the GMLOS filter correctly classifies the edge. This result also could have been concluded from ***Property*** 6.2, since the members of the set S_2 in W would have been classified as an outlier set with respect to the set S,.

□

Property ***6.5 The response of the GMLOS filter to a ramp edge (or blurred step edge), approaches an ideal step edge by repetitive filtering of the data, if the degree of the ramp, d, and the size of the processing window, n, satisfy d/2 < n.***

Proof: Consider ***a*** monotonically non-decreasing ramp edge of degree ***d*** and length ***N***. According to ***definition*** 6.7, a ramp consists of three distinct segments S,, S,, and S,. S, and S_3 are constant segments of different magnitudes, ***g*** and $g+h$, respectively. S_2 is a monotonic sequence of size ***d*** that is bounded from below by S, and from above by S_3. Consider a 1-D processing window W of size *n* as shown in Fig. 6.3. If all members of the set W belong to only one of the constant segments S, or S,, then as shown in the proof of *Property* 6.4, the output of the GMLOS filter is equal to g or $g+h$, respectively. The members of the set W can not entirely belong to the set S,, since by assumption d < $n/2$. It can be assumed, without loss of generality, that the center of the processing window is at the first pixel of the monotonically increasing sequence S,, as depicted in Fig. 6.3.

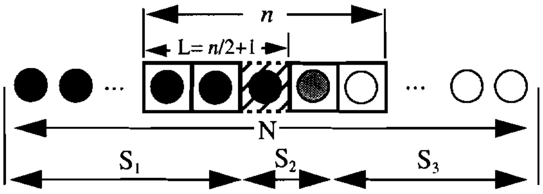


Fig. 6.3. A I-D monotonically non-decreasing ramp edge (the members of the set W are shown inside the square boxes).

According to Theorem 6.1, the set of inliers W^I, is a contiguous set with size L, such that $J(W^1) < J(W^A)$, where W^A is any subset of W with size L. Due to the contiguity requirements, and the above assumption, and from (6.13), (6.15) and (6.16), It can be concluded that the inlier set W^I is a minimizer of (6.6) if and only if all its members belong to only one segment (e.g. S, in this case), except for one member at the center of the processing window, which belongs to the set S,. Therefore the output of the GMLOS filter can. be written as

$$\theta_M = \frac{1}{L}\sum_{z_{(i)}\in W^I} z_{(i)} = \frac{(L-1)g+(g+\Delta g)}{L} = g+\frac{\Delta g}{L}$$

since $\Delta g \ll$ L, after a few passes of the filter over the data the output of the GMLOS filter approaches the mean value of the set S,. Similarly, by repetitive application of the GMLOS filter, the members of the set S,, which are closer in magnitude to the members of S,, will tend to the mean value of the set S, and the rest of the members of S_2 will tend to the mean value of the set S,. Therefore, the response of the GMLOS filter to a ramp will approach a step edge by repetitive filtering of the data.

□

6.3.2 GMLOS for Noise Smoothing and Edge Enhancement

The GMLOS filter has the desired properties of a nonlinear digital filter and can be used for 2-D spatial filtering of digital images. A single-stage 2-D spatial filter operates on the neighborhood of each pixel in the corrupted input image to estimate its value. This neighborhood is called a ***processing window.*** Only windows with a support of size ***n have been considered,*** where ***n*** is an odd integer. The most popular 2-D processing windows for image processing applications are shown in Fig. 6.4.

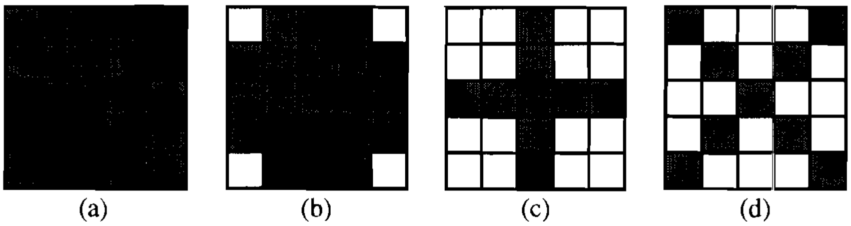


Fig. 6.4. Examples of 2-D processing windows (shaded areas correspond to support of the window). (a) Square window (n = 25). (b) Circular window (n = 21). (c) Cross window (n = 9). (d) X-shaped window (n = 9).

Let $W_p = \{z_i: i=1,\dots,n\}$ be the set of pixels within the support of the processing window, where the subscript p corresponds to the location of the center pixel in the window. Then a 2-D filter can be defined as an operator **T** that maps the input data into the output y_p as

$$\mathbf{y}_p = \mathbf{T}(W_p) \tag{6.23}$$

This value is then assigned to the center pixel of the window as illustrated in Fig. 6.5.

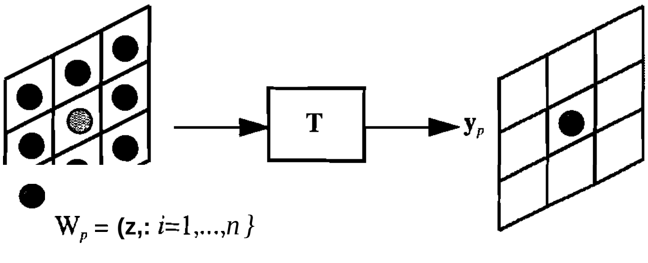


Fig. 6.5. Spatial 2-D filtering with a 3x3 square window.

Properties 6.1 and 6.2 of the GMLOS filter are usually desired in 2-D nonlinear spatial filtering, because the operator T may not satisfy the superposition and proportionality principles in a 2-D nonlinear filter. Property 6.3 is also plays an important rule in 2-D spatial filtering. In many applications the existence of outliers (impulsive or salt-and-pepper noise) is almost unavoidable. They frequently occur at the acquisition time, during the image formation process (due to the existence of long-tailed noise processes), or during transmission over communication channels. Edge preservation is another important property of filters in image processing applications, because human visual perception is very sensitive to this feature [79]. Step edges are rarely found in natural images and most of the edges in these types of pictures have a smooth and monotonic transition from one surface to another (ramp edges). However, in many image analysis and computer vision applications, where edges are being used as the primary features for segmentation and classification, step edges are more desirable. Therefore, the edge enhancement property is an essential attribute of a preprocessing unit in image analysis systems. The majority of the most frequently used filters, such as the median filter and its extensions, do not posses this property. Actually, the median filter tends to preserve any monotonic degradation of the edge, and therefore is not capable of enhancing blurred or ramp edges. This is illustrated in Fig. 6.6.

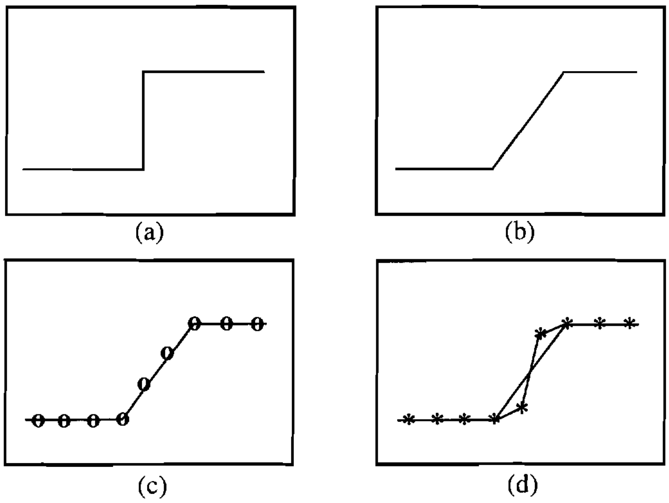


Fig. 6.6. (a) an 1-D ideal step edge, (b) blurred edge (ramp edge), (c) response of the 1-D median filter ('o') superimposed on (b) after 2 passes, (d) response of the 1-D GMLOS filter ('*') superimposed on (b) after 2 passes.

A major drawback in single-stage 2-D spatial filtering is the fact that these filters tend to distort or obliterate some of the fine details of the image [98, 73, 12]. For example, a feature of size $l < L$ in a homogeneous background will be eliminated when filtered by a single-stage GMLOS filter (SS-GMLOS) that has a square window support. To over come this problem in applications in which the preservation of fine details is the major issue of concern, a multi-stage filtering scheme is proposed in [89].

6.3.3 The Multi-Stage GMLOS Filter

A single-stage filter with a relatively large window size, usually destroys the fine details of the image [73]. Moreover, the rank ordering process in order statistic filters may disrupts the structural and spatial information of the data. A uni-directional multi-stage filtering scheme was used in [89] to overcome some of these problems. In this algorithm, a few subfilters with different supports are used to filter the data. The output of the subfilters are then combined to produce the final estimate. These uni-directional subfilters are designed to preserve the image features in different directions. Therefore, by including sufficient number of subfilters, a feature oriented in any direction can be preserved [98]. The uni-directional subfilters for the Multi-Stage GMLOS filter (MS-GMLOS) are shown in Fig. 6.7.

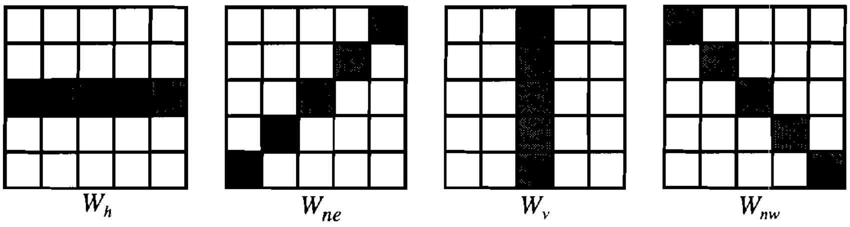


Fig. 6.7. The uni-directional window supports for the MS-GMLOS filter.

The output of the unidirectional multi-stage GMLOS filter is determined by selecting the output of the subfilter for which the output variance is minimum. That is

$$y_{MS-GMLOS} = arg \min_{\hat{\delta}_d^2} (y_{W_d}) \tag{6.24}$$

where y_{W_d} corresponds to the output of the horizontal (W„), north-east (W_{ne}), vertical (W_v), and north-west (W_{nw}) uni-directional subfilters, and the $\hat{\delta}_d^2$'s represent their variances. The uni-directional filtering algorithm is illustrated in Fig. 6.8.

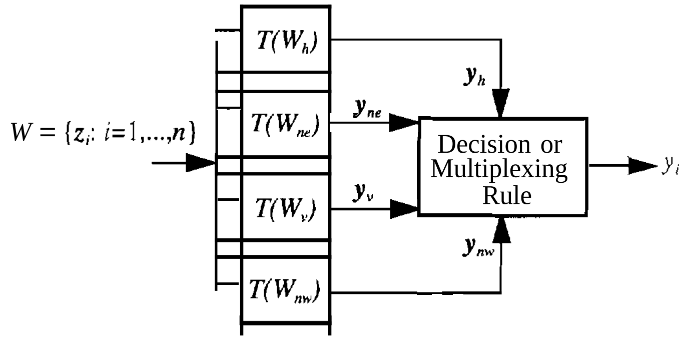


Fig. 6.8, Multi-stage filtering with Four uni-directional subfilters.

In a bidirectional multi-stage filtering scheme, the subwindows span two line segments in orthogonal directions. Two examples of such subfilters were shown in Fig. 6.4 (the Cross and X-shaped windows). The output of the bidirectional MS-GMLOS filter is also determined by selecting the output of the subfilter for which the output variance is minimum. The filtering operation is similar to the one shown in Fig 6.8, with the exception of having only two directional filters, corresponding to $W_{ne\text{-}nw}$ and $W_{h\text{-}v}$.

Although multi-stage filters have superior performance in preserving the fine details of an image, they are not as effective as their single-stage counterparts in attenuating noise or enhancing edges. Within the class of multi-stage filters, bidirectional subfilters have better smoothing properties, but uni-directional subfilters are more successful in preserving fine details of the image. In section 6.4, these facts are verified by experimental results on real images.

6.3.4 GMLOS and Other Nonlinear Edge Enhancing Filters

As mentioned earlier, only a few researchers have designed nonlinear filters with edge sharpening properties. Edges are one of the most important features of an image in many computer vision and image understanding applications, and have great impact on human visual perception [71, 75]. The Comparison and Selection (CS) filter, Weighted Majority of m values with Minimum Range (WMMR^m) filter, and Lower-Upper-Middle (LUM) filter have been shown to be effective in smoothing noise while sharpening edges [77]. These filters and some of their important characteristics are summarized in the following.

The output of the Comparison and Selection (CS) filter [102] with parameter k is given by

$$y_{CS} = \begin{cases} z_{(k)} & \text{if} \quad \hat{\mu} \geq z_{((n+1)/2)} \\ z_{(n-k+1)} & \text{otherwise} \end{cases} \tag{6.25}$$

where $\hat{\mu}$ is the sample mean of the input set W, and $1 \leq k \leq (n+1)/2$. Clearly if $k=(n+1)/2$, then the output of the CS filter is identical to that of the median filter. In general, by selecting different values for the tunable parameter k, different levels of enhancement can be achieved with the CS filter. This filter has good sharpening properties but tends to distort or obliterate fine details of the signal and is not as effective as other nonlinear edge sharpeners in smoothing noise [77].

The output of the Weighted Majority of samples with Minimum Range (WMMR) filter [104] is obtained by averaging the weighted sum of all subsets of W with cardinality $(n+1)/2$ that possess the minimum range property

$$y_{WMMR} = \underset{\forall W^{r_{min}}}{\text{average}} \left(\sum_{z_{(k)} \in W^{r_{min}}} \omega_k z_{(k)} \right) \tag{6.26}$$

where $W^{r_{min}}$ denotes a set with the minimum range property as defined below. Let $r_M = z_{((n-1)/2+M)} - z_{(M)}$ be the range of the elements in W^R, for $1 < M < (n-1)/2+1$, and $r_{min} = \min\ \{r_M, 1 < M < (n-1)/2+1\}$ be the minimum range of the input set W. Then any subset of W with cardinality $(n+1)/2$ is said to possess the minimum range property if all of its members have a range less than or equal to r_{min}. In (6.24), the ω_k's represent the filter weights, and a useful choice of weights for edge enhancement [25] is

$$\omega_k = \begin{cases} 1 & \text{if} \quad k = (n-1)/4 + M \\ 0 & \text{otherwise} \end{cases} \tag{6.27}$$

The WMMR filter with this choice of weights simply computes the sample median of the $W^{r_{min}}$'s, and is referred to as the WMMR-MED filter. A good choice of weights for noise smoothing is the equal normalized weights. The resulting filter is called the WMMR-AVE filter. This filter also distorts or obliterates the fine details of the image, and its edge enhancement property is poorly controlled. However, its noise suppression characteristics are better than CS and LUM filters [77].

The output of the Lower-Upper-Middle (LUM) filter [103] with parameters k and l is given by

$$y_{LUM} = \begin{cases} z_{(k)} & if \quad z_{((n+1)/2)} < z_{(k)} \\ z_{(l)} & if \quad z_{(l)} < z_{((n+1)/2)} \le t_l \\ z_{(n-l+1)} & if \quad t_l < z_{((n+1)/2)} < z_{(n-l+1)} \\ z_{(n-k+1)} & if \quad z_{(n-k+1)} < z_{((n+1)/2)} \\ z_{((n+1)/2)} & otherwise \end{cases} \tag{6.28}$$

where $1 \le k \le l \le (n+1)/2$ and $t_l = (z_{(l)} + z_{(n-l+1)})/2$. The parameters k and l can be considered as tuning parameters that allow the LUM filter to have different characteristics. The output of this filter is bounded by the ranked ordered vales $z_{(k)}$ and $z_{(n-k+l)}$ to remove the outliers. If $k = l = (n+1)/2$, the LUM filter acts as a median filter. In the case where $l = (n+1)/2$ and k is varied, the LUM filter acts as an smoothing filter. When $k = 1$ and l is varied, the LUM filter acts as a sharpener. Finally, when $1 \le k \le l \le (n+1)/2$, sharpening and outlier rejection can be achieved simultaneously. This filter may also distort or obliterate the fine details of the image but is better than the WMMR-MED and CS filters in preserving the details [77]. The optimal values of the tunable parameters k and l for this filter should be obtained through trial and error in different applications.

In general, the CS and LUM filters share a common philosophy because they both select samples that are away from the median, as their outputs. Therefore, different levels of enhancement can be achieved by selecting different values of the tunable parameter k for the CS filter, and the parameters k and l for the LUM filter. However, these tunable parameters are obtained heuristically or by trial and error in different applications. The WMMR and GMLOS filters also share a common philosophy. They first partition the data into inlier and outlier sets before computing their outputs as a weighted sum of the members of the inlier set. For these filters different levels of enhancement can be achieved by choosing appropriate weights and filter length in different applications. While the WMMR filter selects its inlier set and weights by using a heuristic approach, the inlier set and weights in GMLOS are obtained with a recursive maximum likelihood algorithm, which is l_2 norm optimal (see equations 6.9-6.11). The major characteristics of these filters are explained in more detail in the following.

- ***Selection of the inlier set*** - The WMMR filter selects its inlier set(s) by finding all subsets of the ordered input data with cardinality $(n+1)/2$ that have an l_1 norm less than or equal to the minimum range. For digital images this algorithm may often select multiple inlier sets with the required minimum range. For example, consider a 1-D ramp edge $W=\{1,1,1,3,3,3,5,5,5\}$ of degree $d=3$. This set has a minimum range of 2, and there are four subsets $W^{1'}=\{1,1,1,3,3]$, $W^{l2}=\{1,1,3,3,3\}$, $W^{l3}=\{3,3,3,5,5\}$ and $W^{l4}=\{3,3,5,5,5\}$, corresponding to this minimum range. In these cases the WMMR filter uses a heuristic algorithm to compute the output. WMMR-MED averages the medians of the inlier sets, and WMMR-AVE averages the sample mean of the inlier sets. In contrast, the members of the inlier set in the GMLOS filter are recursively chosen based on the theory of maximum likelihood, leading to a recursive optimal l_1 norm solution. For the above example, the GMLOS filter selects a unique inlier set $W^1=\{1,1,3,3,3]$, which has the highest likelihood of being an inlier set. The output is then computed by selecting the weights that yield the BLUE estimate of the location parameter θ, as was explained earlier in this section. The CS and LUM filters do not explicitly partition the data into inlier/outlier sets, however, they can be thought of as having an inlier set with only one member (the output of the filter). In these algorithms the output is obtained by some heuristic choice of the tunable parameters k and l.
- ***Edge enhancement property*** - The edge enhancement properties of the LUM, CS, and WMMR-MED filters are studied in [77]. They are shown to be effective in enhancing blurred edges in presence of impulsive noise. Although the edge enhancement properties of the LUM and CS filters are very similar, in some cases it has been shown that the CS and WMMR-MED filters are somewhat less effective than the LUM filter in enhancing the edges [77]. In general, the GMLOS filter has better edge enhancement properties than the WMMR filter and produces edges that are more localized. The edge localization enhancement properties of the WMMR filter are poor whenever it can not select a unique inlier set. In the above example,

both WMMR-MED and WMMR-AVE produce a value of 3 for the middle pixel, and hence no enhancement is achieved. However, the GMLOS filter is capable of enhancing this ramp edge by producing a value close to 2 in the first iteration, and a value close to 1 in the second iteration (these values are truncated to 2 and 1 respectively).

- ***Noise smoothing property*** - The CS and LUM filters have similar smoothing properties. However, the LUM filter can have potentially better smoothing characteristics than the CS filter, because it has an additional tunable parameter. The WMMR-MED filter has better smoothing properties than the CS and LUM filters [82]. The GMLOS and WMMR filters have comparable noise smoothing properties.
- ***Detail preservation property*** - As mentioned earlier, the single-stage 2-D spatial filters have a tendency to distort fine details of the image (specifically when a large size square window is used). The LUM filter has better detail preservation properties [77] than the CS, WMMR and GMLOS filters. The price to be paid is the selection of two tunable parameters through trial and error in different applications. Moreover, a set of parameters that may perform well in one region of the image that has fine details may not perform as well in other regions with different spatial characteristics. In general, when the preservation of fine details is of prime importance, a multi-stage filtering scheme should be used. While it is possible to adopt an l_2 norm optimal decision rule for computing the output of the MS-GMLOS filter, a heuristic algorithm should be used to implement multi-stage WWMR, CS, and LUM filters.
- ***Filter weights*** - The CS and LUM filters do not use weights in computing their outputs. In the case of the WMMR filters, it is not tractable to compute the optimal values of the filter weights, because this filter uses the l_1 norm to select the inlier set. As a result, only WMMR-MED and WMMR-AVE filters are used in practice. On the other hand, the GMLOS filter has a tractable solution based on a recursive maximum likelihood algorithm for computing the inlier set and filter weights that are l_2 norm optimal.

6.4 GMLOS for Pre-Processing and Post Processing of Images and Image Sequences

Block-based transforms have been extensively used in image and video coding applications [72, 6, 106]. Recently, the Block-Based Discrete Cosine. Transform (BB-DCT) has been adopted by emerging image and video compression standards including JPEG, H.261, MPEG, and the grand alliance HDTV broadcasting [1]. In a block-based image compression Algorithm, such as JPEG, the picture is first divided into many small square blocks that are transformed, quantized, and coded into a sequence of bit streams along with other side information. In a block-based video coding system, such as H.261 and MPEG, a hybrid approach based on the concept of ***intraframe*** and ***interframe*** coding is being used. In the intraframe mode of operation, spatial redundancy is exploited by structuring a single frame into many small square blocks (e.g. 8x8) that are then transformed, quantized, and coded. In the interframe mode of operation, temporal redundancy is exploited by using motion compensation to generate a prediction of the current video frame from previous (and in the case of MPEG possibly from future) frames. The difference between this prediction and the actual frame is then transformed, quantized, and coded along with motion vector information. Finally, the intraframe and interframe information, as well as the synchronization and other side information are structured into a sequence of compressed bit streams for storage or transmission.

The above mentioned standards belong to the class of ***lossy*** data compression and usually introduce visible distortions in the form of ***blockiness*** and/or *motion* ***jerkiness*** to the original data. The amount of distortion is a function of the coding parameters (e.g., quantization step sizes and range of motion vectors), the data rate used, the buffer control algorithm, and the particular source video sequence. In addition to Gaussian and impulsive sensor noise, blockiness, and motion artifacts, the encoded bit streams are vulnerable to transmission or media impairments that may cause loss of block(s) of data or loss of syr~chronizationSome of these impairments can be reduced by performing a set of pre-processing and post-processing algorithms on the encoded or decoded data [10], as shown in Fig. 6.9.

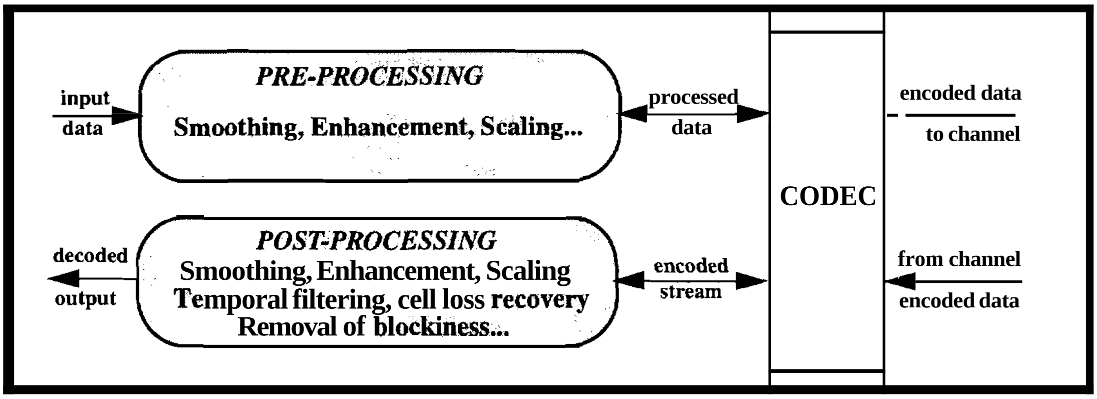


Fig. 6.9. A generic block diagram for pre-processing and post-processing algorithms in modem interactive communication systems.

Pre-processing refers to spatial and temporal image processing algorithms (e.g., noise smoothing, scaling, etc.) that are applied to the data signal prior to compression. Similarly, post-processing algorithms refer to signal or image processing functions that are applied to data after compression.

The removal of additive Gaussian and impulsive noise can be achieved by the GMLOS filters introduced in the previous section. In this section a novel algorithm is introduced for reduction of blocking effects in block-based transform coded images and video frames. The effects of temporal filtering in reduction of motion artifacts is also discussed. The concealment of errors due to transmission impairments is the subject of the next chapter.

6.4.1 De-blocking of Encoded Images and Video Frames

At lower bit rates, the coarse quantization of transform coefficients in block-based image and video compression algorithms usually results in visible distortions in the form of blockiness, as illustrated in Fig. 6.10.

In recent years, a variety of post-processing algorithms have been proposed to remove the blocking effects. The most intuitive solution is to use a low-pass filter to smooth out the boundary effects [10, 107]. Although the low-pass filter can smooth the boundaries of the blocks, it also eliminates the high frequency information corresponding to the interior of the blocks, and thus blurs the real edges.

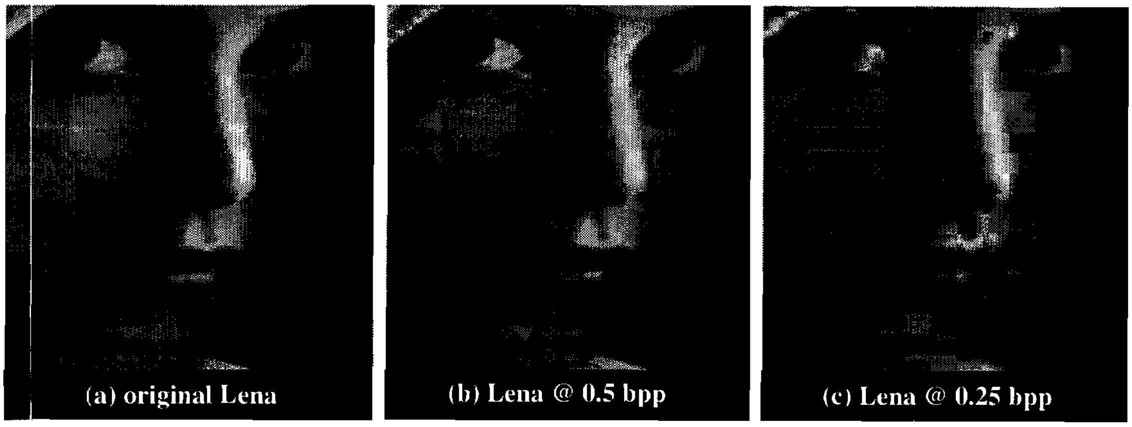
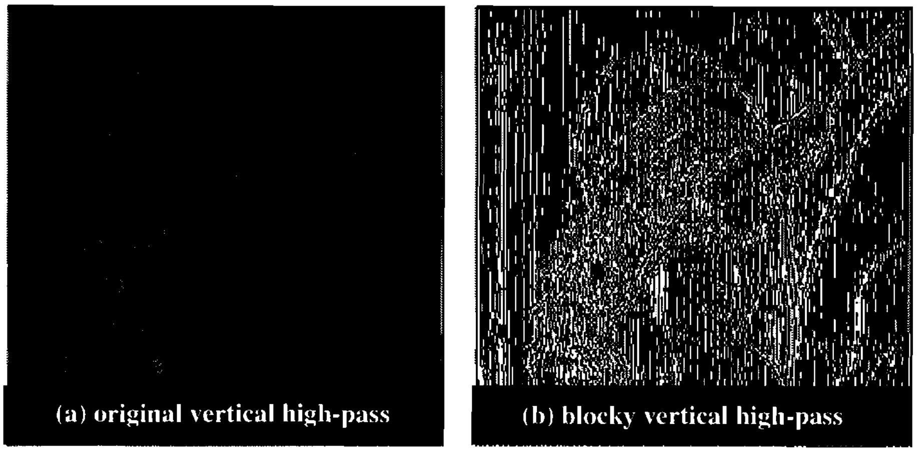

(a) original Lena
(b) Lena @ 0.5 bpp
(c) Lena @ 0.25 bpp

Fig. 6.10. The blocking effects in block-based transform coding algorithms at lower bit rates. (a) Original Lena. (b) Encoded with JPEG at 0.5 bits/pixel. (c) Encoded with JPEG at 0.25 bits/pixel. The visible distortions in form of blockiness is noticeable in (c).

At the cost of computational complexity, better results have been achieved by iterative algorithms such as Projection Onto Convex Sets (POCS) [108, 109], and edge-based spatially adaptive filtering [110]. More recently, wavelet-based post-processing techniques have gained popularity. In [111], a wavelet-based algorithm that uses soft-thresholding of the wavelet coefficients [112] is used to remove the blockiness.

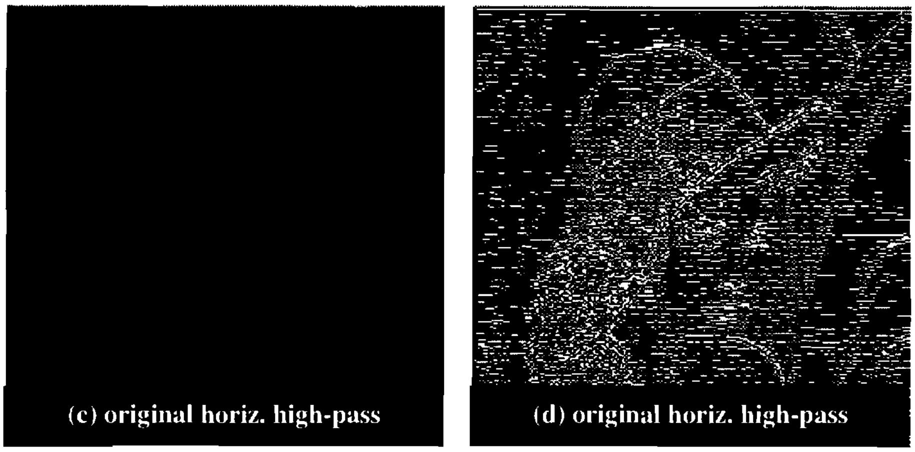
Our approach to de-blocking is motivated by the following facts:

- In a block-based image coding algorithm, the locations of the block boundaries are known. Therefore, the smoothing operation should be restricted to regions around the block boundaries.
- The detail images in the wavelet-based MRA of a blocky image contain vertical and horizontal artifacts in the vertical and horizontal high-pass images, respectively. These artifacts are usually located at the block boundaries, as shown in Fig. 6.11. Therefore, most of these artifacts can be removed by proper filtering and thresholding.
- The block boundaries in the approximation image of the wavelet based MRA should be smoothed with a filtering scheme that does not smear the true edges of the image.
- The de-blocking algorithm should be computationally efficient.



(a) original vertical high-pass

(b) blocky vertical high-pass



(c) original horiz. high-pass

(d) original horiz. high-pass

Fig. 6.11. Vertical and horizontal artifacts in wavelet-based **MRA** of blocky images. (a) The vertical high-pass image of original Lena of Fig. 6.10(a). (b) The vertical high-pass image of blocky Lena of Fig. 6.10(c). (c) The horizontal high-pass image of original Lena of Fig. 6.10(a). (d) The horizontal high-pass image of blocky Lena of Fig. 6.10(a). Figs. (b) and (d) are enhanced for display purposes. The results are shown for level 1, with an orthonormal **MRA,** and based on D4 compactly supported wavelets.

In this algorithm the blocky image is first decomposed into approxiimation and detail subspaces with J-level biorthogonal or orthonormal MRA analysis (in practice, a 2-level decomposition is sufficient). In the next step, the low-pass approximatiion image and the high-pass detail images are processed independently.

Although a low-pass filter can be used to smooth the boundaries of the blocks in the approximation image, it also eliminates some of the high frequency information and thus blurs the edges. In order to overcome this problem, *Spatially Variant Filters* (SVF) can be used [10], Since the location of the block boundaries are known, it is intuitive to smooth the signal along these boundaries, while preserving the pixels that are located well within these blocks. Moreover, different kernels can be used for the pixels along the block boundaries, where the shape and size of the kernel depends on the location of each pixel, as shown in Fig. 6.12(a). In this method a corner pixel is processed with a two dimensional kernel, while pixels along the boundaries are filtered with one dimensional (vertical or horizontal) kernels. In addition, the size of the kernel (one or two dimensional) can vary depending on the distance between the pixel under consideration and the block boundaries based on the local statistics of the image [10]. Linear low-pass SVF filters may blur edges that are close to the block boundaries. To overcome this problem, a variable length GMLOS filter is used to smooth out the boundaries while preserving the true edges.

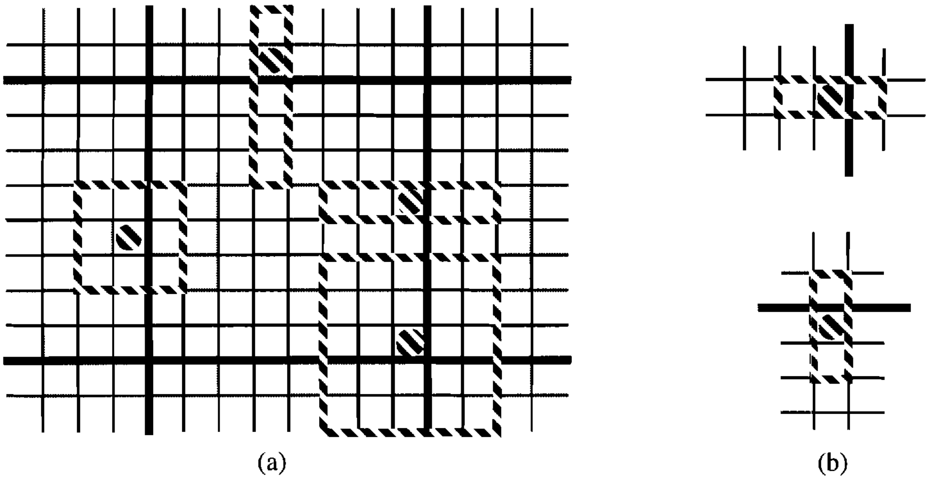


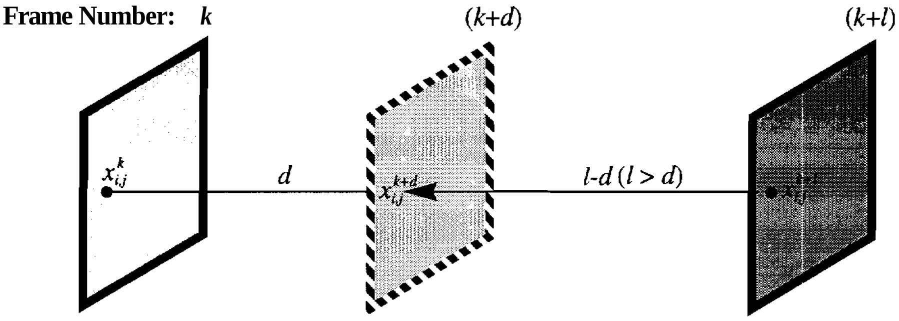
Fig. 6.12. Examples of variable size kernels (window supports) for an SVF filter. (a) SVF window support for filtering the approximation image. (b) 1-D horizontal and vertical support for filtering of block edges in the vertical detail and horizontal detail images, respectively.

As shown earlier (see Fig. 6.11), the detail images in wavelet-based MRA of a blocky image contain vertical and horizontal artifacts in the vertical and horizontal high-pass images, These artifacts are located exactly at the boundary of blocks. In addition, our experimental studies have shown that the wavelet coefficients associated with these artifacts are much smaller than those corresponding to the true edges of the image, and are mostly clustered around zero. In this algorithm, the pixels at the vertical boundaries of blocks in the vertical high-pass image are filtered with a 1-D horizontal GMLOS filter. Similarly, the horizontal boundaries in the horizontal high-pass image are filtered with a 1-D vertical GMLOS filter. The window support for these 1-D filters is shown in Fig 6.12(b). This filtering scheme is performed prior to thresholding the wavelet coefficients in order to strengthen the value of the coefficients for possible real edges at the boundary. The wavelet coefficients in detail images should then be thresholded for elimination of false boundary edges. **A** soft thresholding algorithm similar to [112] can be used to achieve this task. Finally, the filtered and thresholded approximation and detail images are used to reconstruct the image at the original resolution. This algorithm is simple and computationally efficient. It successfully de-blocks the encoded images and its performance is comparable to the more sophisticated and computationally expensive iterative algorithms. Experimental results are provided in section 6.5.

6.4.2 Temporal Filtering of Image Sequences

The frame rate of the video signal at the output of a decoder operating at low bit rates is usually less than 30 frames per second [6, 1]. For example, an H.261 codec (coder1 decoder) operating at 384 kbits/s operates at a rate of about 10-15 frames per second. In order to generate a standard 30 frames video signal, some temporal processing has to be performed at the output of the decoder. The most simple approach is to repeat the decoded frames to compensate for the missing frames that the encoder did not transmit. This creates *motion jerkiness* that is commonly noticed when the codec compresses the video signal at a rate of 15 frames per second or lower [1]. The motion artifacts are also noticeable at the boundary of the blocks due to possible inaccuracies of motion vectors in a block-based motion compensation algorithm [6].

Temporal interpolation and filtering can be effectively used to reduce some of these artifacts. Under this approach, two consecutive decoded frames are used to create the missing frame. Moreover, the consecutive frames can be filtered in the temporal direction to reduce some of the motion artifacts at the boundary of the blocks [10]. Although temporal interpolation at the output of the codec may improve the quality of the video signal by producing smoother motion, it also introduces additional delay and costs [10]. The increase in delay is due to the fact that the post-processor has to wait for two consecutive frames (which are, in general, two or more frames apart) before it begins displaying the output video. The extra cost is due to the additional frame buffer that will be needed to support this process. It has been shown that [10] bilinear interpolation is the optimal choice (in the cost-performance sense) for post-temporal interpolation. The bilinear temporal interpolation is illustrated in Fig. 6.13.



$$x^{k+d}_{i,j} - \mathrm{d}\ x^{k+l}_{i,j} + (l-d)\ x^{k}_{i,j}$$

Fig. 6.13. Post-temporal bilinear interpolation of video frames. The value of the pixel at the interpolation site (i,*j*) is a function of the distance d and the pixel values of its closest neighbors.

The performance of the post-temporal interpolation was found to be highly scene dependent. When the inter-frame motion is small, post-temporal filtering consistently improves the PSNR. However, as the motion content of the scene increases, bilinear interpolation is not as effective. In this case, it is possible to reduce some of the artifacts with uni-directional multistage GMLOS filtering of the consecutive frames in the temporal direction. The filtering scheme is identical to the one introduced in section 6.3.3, with the exception that each subfilter has a support across the consecutive frames in the temporal direction. Experimental results are presented in section 6.5.

6.5 Experimental Results and Conclusions

6.5.1 Noise Removal and Edge Enhancement

The simulations were carried out on various test images with different types of degradations, such as Gaussian, impulsive, and linear blur, as listed in Table 6.1.

Table 6.1
The description of test images.

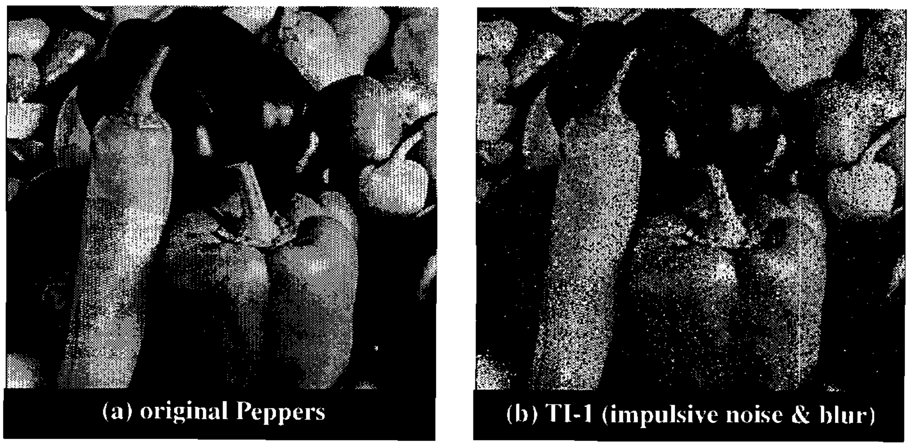
Corrupted Image	Original Image	Description of degradations		
		Gaussian	Impulsive	Blur
TI-1 (Fig. 6.14(b))	Peppers (Fig. 6.14(a): 512x512x8)	None	10%	3x3 mean filter
TI-2 (Fig. 6.15(b))	Einstein (Fig. 6.15(a): 512x512x8)	(0,100)	10%	None
TI-3 (Fig. 6.16(b))	Stream (Fig. 6.16(a): 512x512x8)	None	10%	None
TI-4 (Fig. 6.17(c))	Rods (Fig. 6.17(a): 256x256x8)	None	None	3x3 mean filter

TI-1, which does not contain fine levels of detail, is shown in Fig. 6.14(b). It was created by blurring the original Peppers with a 3x3 mean filter (linear blur), followed by randomly replacing 10% of the pixels with gray scale values of ±200. **A** 3x3 square window was used for all single-stage filters. This type of processing window normally achieves the highest level of noise attenuation, but it has a tendency to obliterate fine details of the image [76, 98]. For multi-stage filtering, 5x5 bidirectional processing windows were used. These filters have been shown to have better noise attenuation properties than uni-directional multi-stage filters, but are not as good at preserving the details [12, 98].

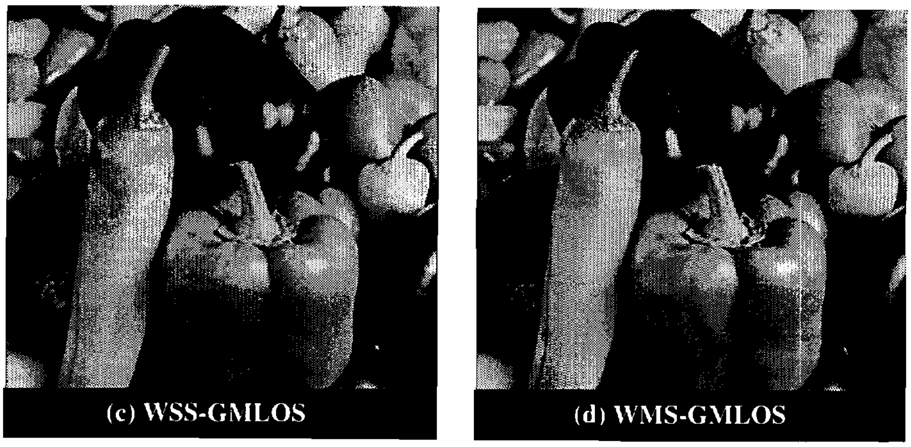
TI-1 was filtered with the Weighted Single-Stage (WSS-GMLOS) and Weighted Multi-Stage GMLOS (WMS-GMLOS) filters. The results are shown in Figs. 6.14(c), and (d), respectively. Both of the GMLOS filters appear to be successful in editing the impulses and enhancing the edges. The corresponding Mean-Absolute-.Error (MAE) and Mean-Squared-Error (MSE) for these filters is listed in Table 6.2. The MAE and MSE measures, are the most frequently used measures of error in image processing applications [71]. For comparison, the MAE and MSE for the nonlinear filters of section 6.3.4, median, and multistage median filters are also listed in Table 6.2[4].

---

[4] *No algorithm were found in the literature for multi-stage LUM, CS, or WMMR filters.*



(a) original Peppers (b) TI-1 (impulsive noise & blur)



(c) WSS-GMLOS (d) WMS-GMLOS

Fig. 6.14. The result of GMLOS filters for TI-1. (a) Original Peppers. (b) Noisy test image 1 (TI-1). (c) Weighted single-stage GMLOS filter. (d) Weighted multi-stage GMLOS filter.

Table 6.2
MSE and MAE for test image 1 (TI-1).

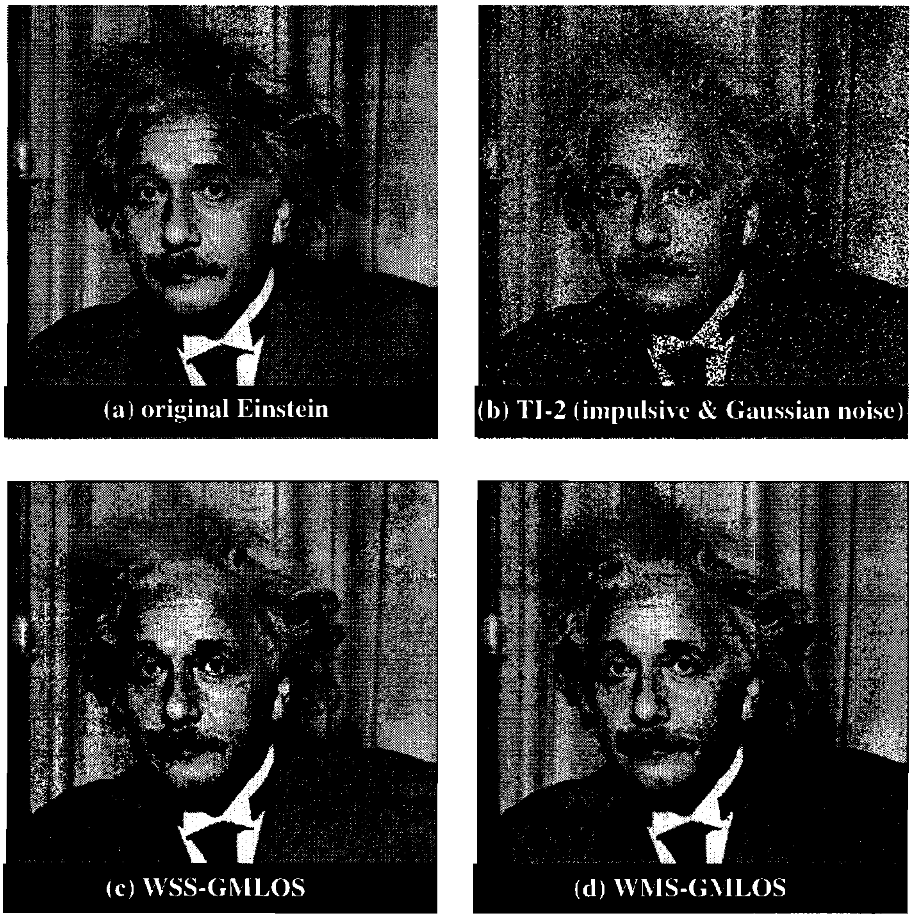
TI-1	**No Filtering**	**WSS-GMLOS 3x3 Square**	**CS k = 4 3x3 Square**	**WMMR-AVE 3x3 Square**	**WMMR-MED 3x3 Square**	**LUM k=2, l=5 3x3 Square**	**GMLOS Bidirectional**	**Median Bidirectional**
MSE	2318.63	242.21	258.27	248.96	257.47	256.25	248.96	263.05
MAE	17.11	5.40	5.81	5.57	5.68	5.70	5.65	

The second test image TI-2, which has more fine details than TI-1, is shown in Fig. 6.15(b). It was created by adding Gaussian noise (zero mean, and variance of 100) to the original Einstein, followed by randomly replacing 10% of the pixels with gray scale values of ±200. A 3x3 square window was used for all the single-stage filters, and 5x5 bidirectional processing windows were used for multi-stage filters. TI-2: was filtered with WSS-GMLOS and WMS-GMLOS filters. The results are shown in Figs. 6.15(c), and (d), respectively. Both of the GMLOS filters appear to be successful in editing the noise while preserving the ***significant*** details of the image. The corresponding MAE and MSE for these filters is listed in Table 6.3.

TI-3, which has the finest level of details among the test images of Table 6.1, is shown in 'Fig. 6.16(b). It was created by randomly replacing 10% of the pixels in the original Stream image, with gray scale values of ±200. A 3x3 X-shaped window was used for all the single-stage filters, because X-shaped filters are more suitable for detail preservation than the square windows. For multi-stage filtering, 5x5 uni-directional processing windows were used, because they are more effective in preserving the details than bidirectional filters. TI-3 was filtered with WSS-GMLOS and WMS-GMLOS filters. The results are shown in Figs. 6.16(c), and (d), respectively. Both of the GMLOS filters appear to be successful in editing the noise. However, WMS-GMLOS performed better than WSS-GMLOS, in terms of preserving details.

TI-4 was used to demonstrate the edge enhancement properties of the nonlinear sharpeners of Section 6.34. It was created by filtering the original Connection Rods image, with a 3x3 mean filter (linear blur). This image was then filtered by single-stage median, GMLOS, CS, WMMR-AVE, WMMR-MED, and LUM filters. The filtered images and their corresponding gradient based edge maps are shown in Fig. 6.17. A subjective evaluation of Fig. 6.17 reveals that the SS-GMLOS has produced an edge map that is more localized, and has more amount of detail than the other nonlinear sharpeners.

For processing TI-1, TI-2, and TI-3, all possible values for the ***tunable*** parameter(s) of CS and LUM filters were used, and the tunable parameter(s) that produced the minimum MSE/MAE was chosen. For TI-4, the tunable parameters for which the edge maps had more detail were chosen.



Fig;. 6.15. The result of GMLOS filters for TI-2. (a) Original Einstein. (b) Noisy test image 2 (TI-2). (c) Weighted single-stage GMLOS filter. (d) Weighted multi-stage GMLOS filter.

Table 6.3
MSE and MAE for test image 2 (TI-2).

TI-2	**No Filtering**	**WSS-GMMS *3x3* Square**	**CS k = 4 3x3 Square**	**WMMR-AVE 3x3 Square**	**WMMR-MED 3x3 Square**	**LUM *k*=3, *l*=5 *3x3* Square**	**WMS-GMLOS 5x5 Bidirectional**	**MS-Median 5x5 Bidirectional**
MSE	2084.82	136.83	128.27	156.42	168.89	140.38	125.32	127.67
MAE	20.00	8.63	7.72	8.55	8.84	8.42	7.12	7.15

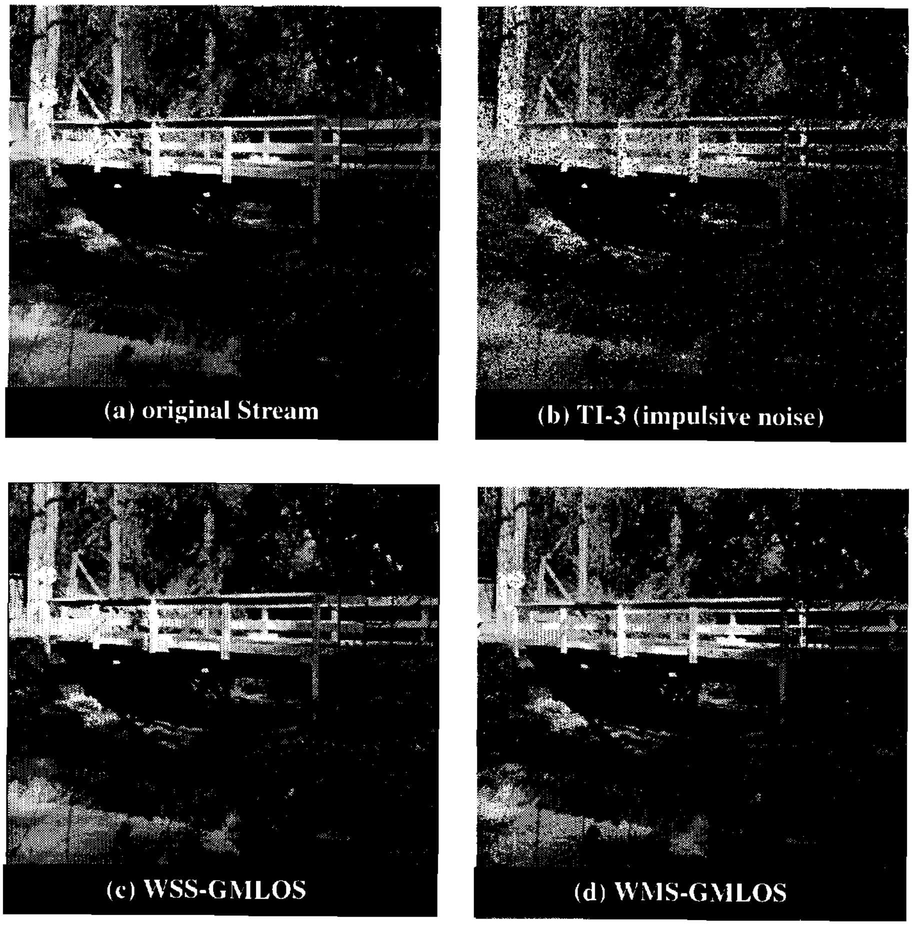
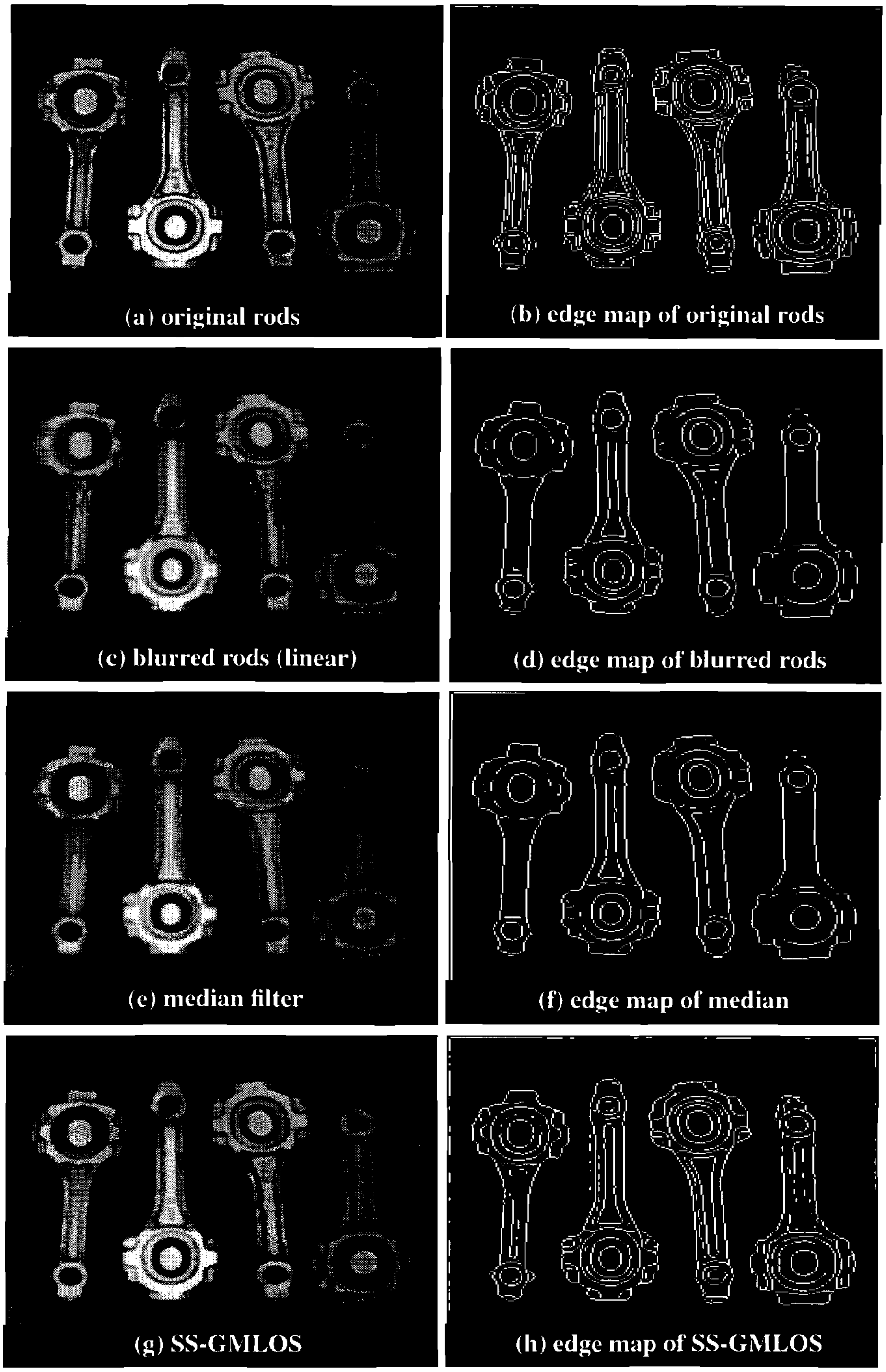


Fig. 6.16. The result of GMLOS filters for TI-3. (a) Original Stream. (b) Noisy test image 3 (TI-3). (c) Weighted single-stage GMLOS filter. (d) Weighted multi-stage GMLOS filter.

Table 6.4
MSE and MAE for test image 3 (TI-3).

TI-3	**No Filtering**	**WSS-GMLOS 3x3 X-shaped**	**CS k = 4 3x3 X-shaped**	**WMMR-AVE 3x3 X-shaped**	**WMMR-MED 3x3 X-shaped**	**LUM k=3, l=5 3x3 X-shaped**	**WMS-GMLOS 5x5 uni-directional**	**MS-Median 5x5 uni-directional**
MSE	1858.19	150.97	154.80	186.44	204.37	218.01	150.25	209.86
MAE	12.16	7.15	7.26	8.22	8.16	8.60	4.20	4.25



(a) original rods

(b) edge map of original rods

(c) blurred rods (linear)

(d) edge map of blurred rods

(e) median filter

(f) edge map of median

(g) SS-GMLOS

(h) edge map of SS-GMLOS

Fig. 6.17. Edge enhancement with nonlinear filters. (a) Original rods (256x256x8). (b) The edge map of (a). (c) Blurred rod (linear blur). (d) The edge map of (c). (e) Median filter. (f) The edge map of (e).

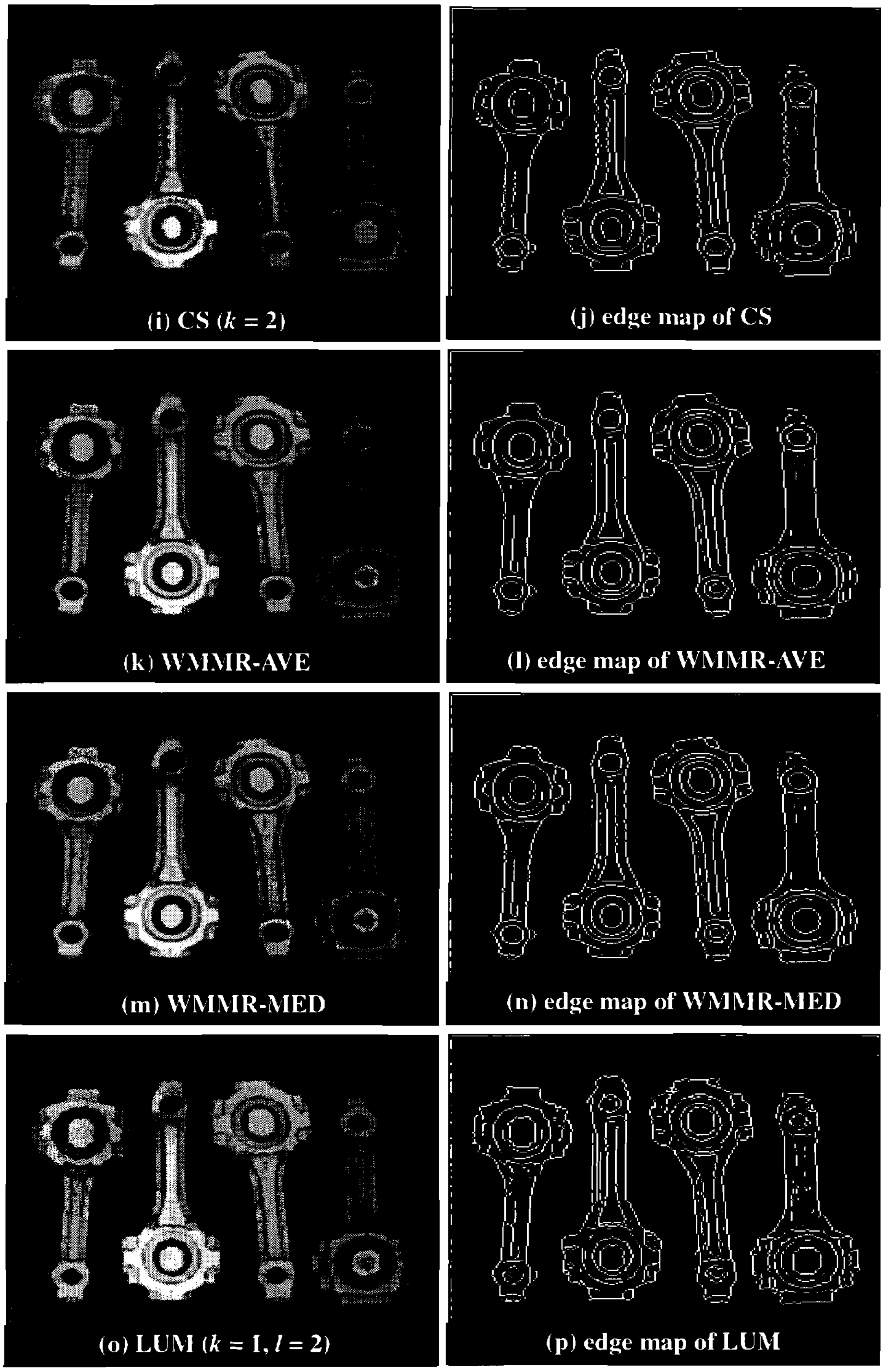


Fig. 6.17. Continued: (g) SS-GMLOS filter. (h) The edge map of (g). (i) CS filter (*k*=2). (j) The edge map of (i). (k) WMMR-AVE. filter (l) The edge map of (k). (m) WMMR-MED filter. (n) The edge map of (m). (o) LUM filter (*k*=1, *l*=2). (p) The edge map of (o).

6.5.2 De-blocking of Block-Based Transformed Coded Images

The proposed de-blocking algorithm of section 6.4.1 was tested on various blocky images. The test image Lena is used here, because of its wide-spread use in the literature. The original zoomed Lena is shown in Fig. 6.10(a). The JPEG compressed Lena at 0.25 bits/pixel is shown in Fig. 6.18(a), and its zoomed version in Fig. 6.18(b). A 2-level MRA was performed on this image by using Daubechies D4 wavelets. The high-pass vertical and horizontal detail images are shown in Figs. 6.18(c), and (d), respectively. The corresponding GMLOS filtered and soft-thresholded detail images are shown in Figs. 6.18(e), and (f), respectively. Finally, the reconstructed de-blocked image and its zoomed version are shown in Figs. 6.18(g), and (h), respectively. The subjective quality of the de-blocked image is good, and it has a PSNR of 30.52 dB. The obtained results are comparable to those reported in [109], as listed in Table 6.5, however the proposed algorithm is computationally efficient, because it is not iterative.

Table 6.5
De-blocking results for Lena test image (PSNR in dB).

Bit Rate	JPEG(PSNR)	Method in [46]	Our algorithm
0.15	26.57	27.58	27.45
0.24	29.35	30.43	30.40

6.5.3 De-blocking and Temporal Filtering of Video Frames

In this experiment the ***mobile-calendar*** and ***table-tennis*** Common Intermediate Format (CIF) [6] image sequences were used. The sequences where 100 frames long. These sequences were compressed at 384 Kbps with an H.261 software codec. The decoded frame rate was 15 frames per second. The decoded frames were first de-blocked, and then bilinear interpolation was used to interpolate the missing frames. The interpolated frames were then filtered with a uni-directional multi-stage GMLOS filter, where each sub-filter had a support of length three (i.e. the pixels from the previous and the next frame were used). The de-blocked frame 50 of both sequences is shown Fig. 6.19. On average, the post-temporal filtering and interpolation improved the PSNR by up to 2 dB. The results are shown in Fig. 6.19.

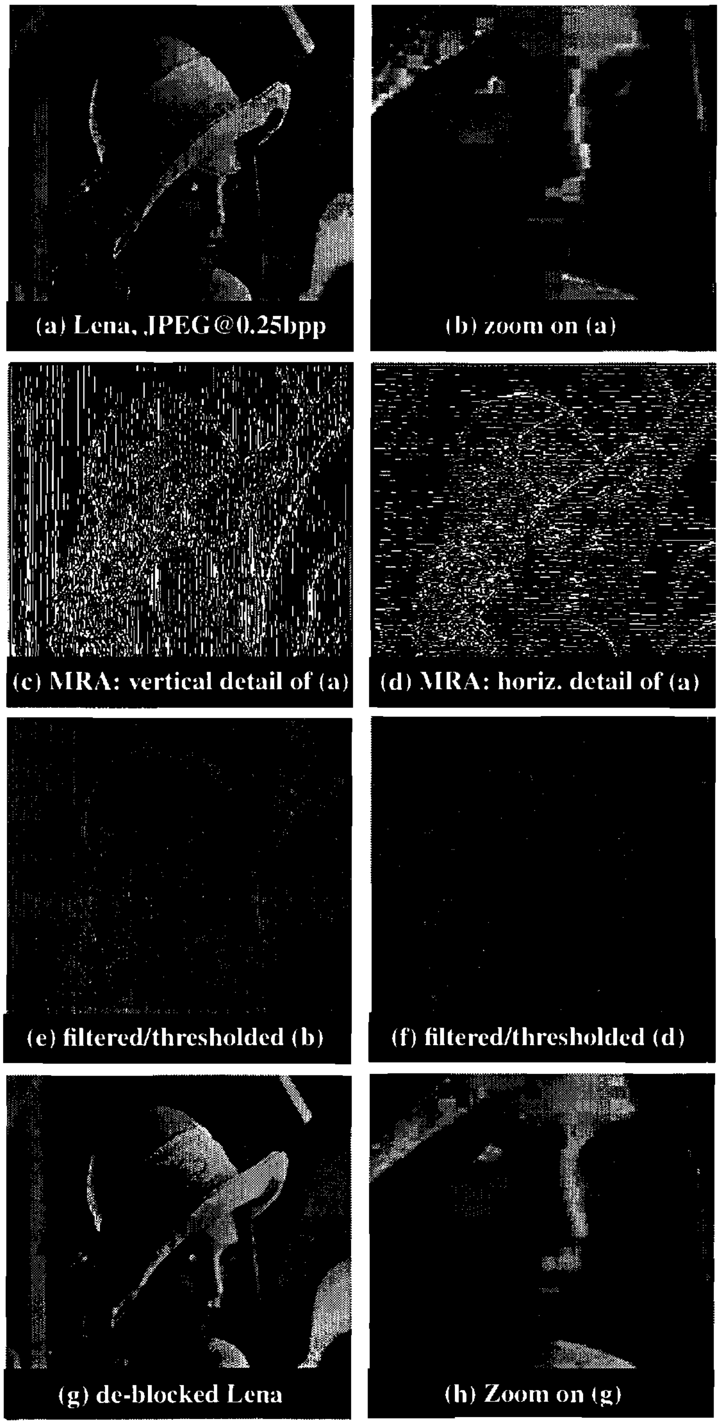
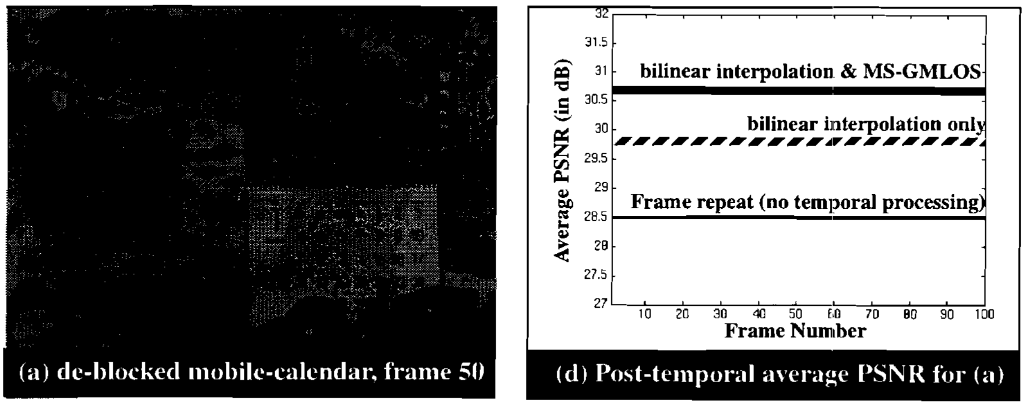
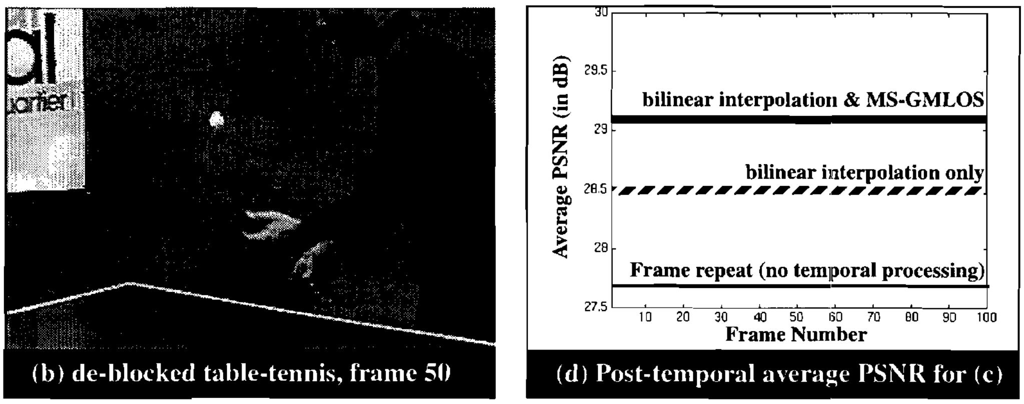

(a) Lena, JPEG@0.25bpp
(b) zoom on (a)
(c) MRA: vertical detail of (a)
(d) MRA: horiz. detail of (a)
(e) filtered/thresholded (b)
(f) filtered/thresholded (d)
(g) de-blocked Lena
(h) Zoom on (g)

Fig. 6.18. De-blocking with GMLOS and wavelet based MRA (Daubechies D4, J = 2). (a) JPEG encoded Lena at 0.25 bits/pixel. (b) Zoomed version of (a). (c) Vertical high-pass image of (a) at resolution J = 1. (d) Horizontal high-pass image of (a) at resolution J =1. (e) (c) processed with GMLOS and soft-thresholding. (f) (d) processed with GMLOS and soft-thresholding. (g) The de-blocked image, PSNR = 30.52 dB. (h) Zoomed version of (g). Figs. (c) and (d) were enhanced for viewing purposes.



(a) de-blocked mobile-calendar, frame 50

(d) Post-temporal average PSNR for (a)



(b) de-blocked table-tennis, frame 50

(d) Post-temporal average PSNR for (c)

Fig. 6.19. Average PSNRs for post-temporal filtering and de-blocking. H.261 encoded at 384 Kbps, and decoded at 15 frames per second. (a) De-blocked frame 50 of mobile-calendar (CIF: 288x352). (b) Average PSNR performance for mobile-calendar with no post-temporal filtering, bilinear interpolation, and bilinear interpolation with MS-GMLOS temporal filtering. (c) De-blocked frame 50 of table-tennis sequences (CIF: 288x352). (d) Average PSNR performance for table-tennis with no post-temporal filtering, bilinear interpolation, and bilinear interpolation with MS-GMLOS temporal filtering.

In this chapter a new class of robust nonlinear filters based on the theory of generalized maximum likelihood estimation and order statistics were introduced. It was shown that this class of filters is not only capable of attenuating noise and preserving details, but also has the ability to sharpen edges. The simulation results on real images have confirmed these claims. In general, when the image does not contain fine levels of detail and some degree of edge enhancement is desired, it is advantageous to use the single-stage GMLOS filter. However, when the goal is to preserve the fine details, the unidirectional multi-stage GMLOS is a better choice. Comparative experimental studies on real images showed that the weighted single-stage GMLOS filter has better edge enhancement and noise smoothing properties than other nonlinear edge sharpeners, and the weighted multi-stage GMLOS filter is better than other single and multi-stage nonlinear filters in preserving details. Moreover, while most of the other filters use heuristic algorithms to obtain their outputs, the GMLOS filter uses a recursive and computationally efficient algorithm which is l_2 norm optimal.

When prior knowledge about the degradation process is available, tunable nonlinear filters such as CS and LUM filters give the designer some degree or degrees of freedom to optimize the performance of the filter. However, the optimal value of the tunable parameters may vary drastically for different types of degradations. Therefore in many practical applications such as satellite or deep space imaging and image sequence filtering, where complete knowledge of the degradation process is not available, or user interrupts are not allowed, non-tunable filters such as GMLOS might be more desirable. A few applications of the GMLOS filter in the pre-processing and post-processing of images and image sequences were also presented. It was shown that when used in a suitable algorithm, this filter can greatly improve the quality of the decoded images.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

7. ERROR CONCEALMENT OF IMAGE AND VIDEO STREAMS OVER THE ATM NETWORKS

7.1 Introduction

In addition to the coding artifacts that usually introduce visible distortions in the forms of ***blockiness, ringing,*** and ***motion jerkiness,*** encoded bit streams are vulnerable to transmission or media impairments that may result in loss of block(s) of data or total loss of synchronization in the modem packet-switched broadband communication networks or wireless communication systems. The effect of cell loss on picture quality mainly depends on the image or video compression algorithm and the packetization technique used for the transmission of visual information. Block-Based Discrete Cosine Transform (BB-DCT) has been adopted by emerging image and video compression standards including JPEG, H.261, MPEG, and the grand alliance HDTV broadcasting [1]. Asynchronous Transfer Mode (ATM) has recently been accepted as the switching protocol standard for the implementation of the Broadband Integrated Services Digital Networks (B-ISDN) [113] and is provisioned to be the switching protocol standard of future broadband multimedia communication systems. Two major drawbacks of ATM networks are the ***jitter delay*** and ***cell loss*** due to channel congestion or buffer overflow which are important issues of concern in image and video communications. In the latter case, isolated or contiguous blocks of spatial or temporal data may be lost, resulting in severe degradations in the subjective quality of the decoded image or video streams.

In ATM networks, simple error recovery techniques such as automatic retransmission request (ARQ) may not be effective. ARQ will further aggravate the channel congestion and cause the system to drop more cells. During recent years, many error concealment and correction techniques have been proposed to make the information loss of the decoded bit streams subjectively imperceptible. These techniques include block interleaving and forward error correction [114, 115], deinterleaving and scrambling [116], layered coding and prioritizing of cells [117, 118, 119], and spatial and temporal post-processing error concealment [120, 121, 122, 123, 124, 125]. These algorithms use one or more of the following classes of techniques:

- *Error correction techniques:* in this class of techniques, corrupted or missing information is detected and replaced by its exact value using the redundant information that has been inserted into the compressed data stream.
- *Error reduction techniques:* in this class of techniques, the data stream is either interleaved, prioritized, or sent in layers to reduce the impact of cell loss on the perceived quality of the decoded image.
- *post-processing concealment techniques:* in this class of techniques, corrupted or missing information is estimated by using the uncorrupted information available in the data stream.

Error correction techniques that require the insertion of extra information into the compressed bit stream do not always offer a feasible solution for the cell loss problem because of the limited bandwidth available. For example, the optional double error correction code of H.261 (BCH (511,493,2)) requires 18 parity bits for each 493 bits of information. If we were to have a 10 error correction BCH code, the number of parity bits should be increased to 90 bits, and the number of information bits decreased to 421. Moreover, in packet based networks such as ATM, the amount of overhead is much higher because of the demand for an FEC with high correction capability in the event of cell loss due to channel congestion or buffer overflow.

Likewise, error reduction techniques that require layered transmission or cell prioritization do not offer a feasible solution for the cell loss problem. This is mainly due to the fact that low error rate transport channels for high priority data are costly, and the safe delivery of high priority information is not always guaranteed in these systems.

The post-processing techniques that use the available uncorrupted data to approximate the value of missing information have gained popularity in recent years because of their effectiveness, low overhead, and low computational complexity. While for still images only spatial reconstruction is required, for video, both spatial and temporal reconstruction should be performed at the decoder to approximate the missing information, as depicted in

Fig. 7.1.

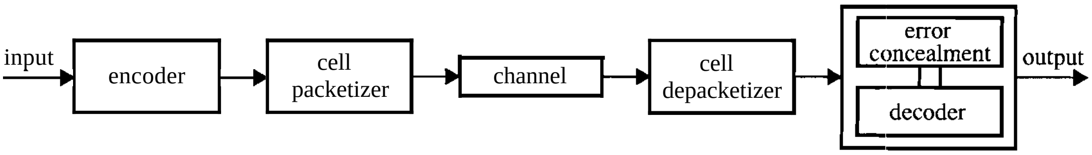


Fig. 7.1. Block diagram of post-processing error concealment scheme.

It is important to note that in post-processing techniques the location of the lost information must be known a priori to the decoder. Since most of the encoders, including the existing image and video standards, only provide a differential address for data blocks, the location information must be provided to the decoder by the cell packing algorithm.

In this chapter, a novel post-processing technique for error concealment in block-based image and video coding systems over ATM networks is presented. The encoded bit streams are packed into ATM cells using a simple error reduction technique such that in the event of cell loss the location of the missing block is known to the decoder, and to guarantee safe transmission of important synchronization and header information. The post-processing in spatial domain is achieved by using the WSS-GMLOS filter [12, 13] in a multi-directional recursive nonlinear filtering scheme [11] with variable size kernels, and the lost temporal informations are reconstructed with GMLOS filtering of the motion vector (MV) components of the neighboring macroblocks.

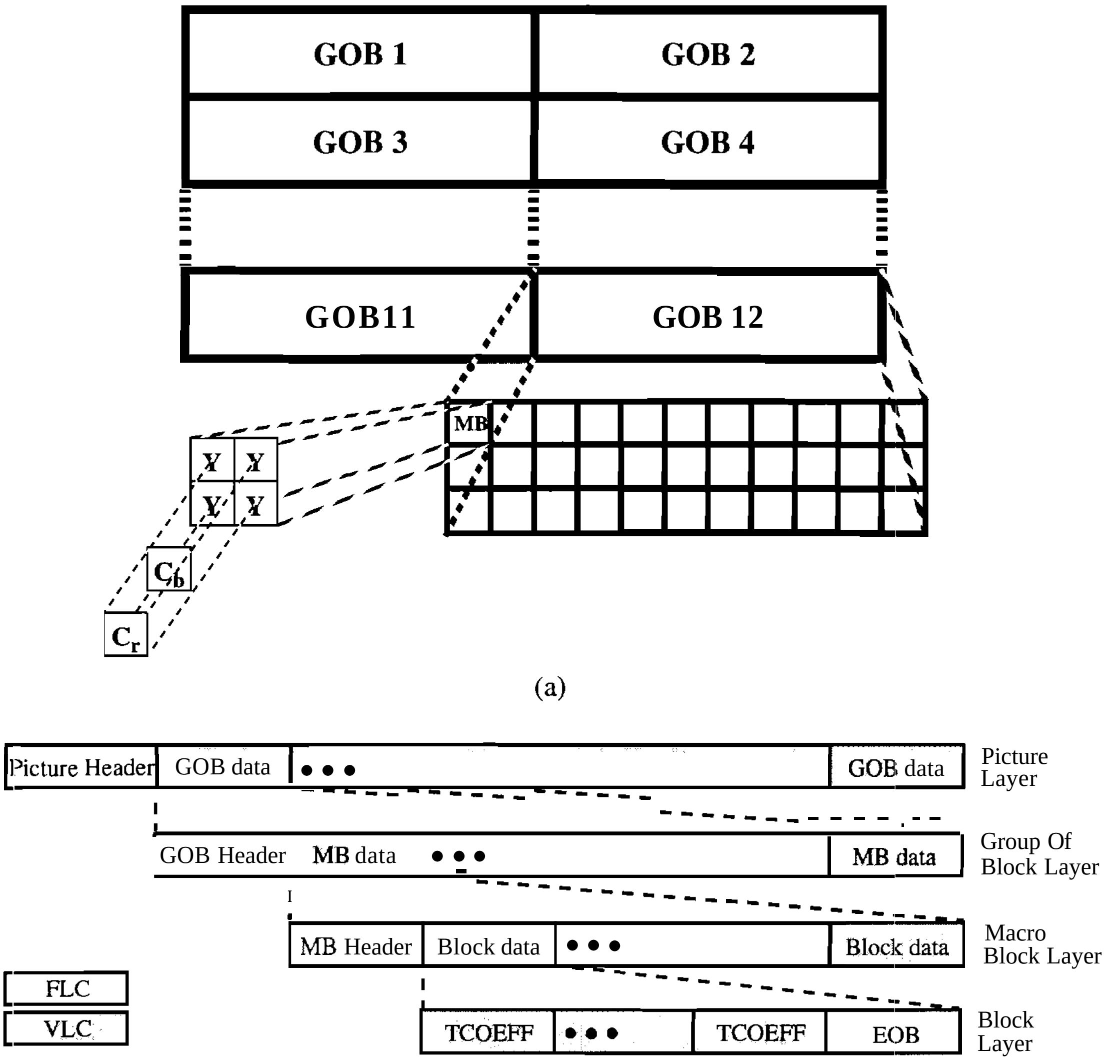
7.2 ATM and ATM Cell Packing

ATM is a low-delay, high-bandwidth switching and multiplexing technology for both public and private networks and is provisioned to become the preferred transport technology for future broadband communication networks [113, 126]. This is mainly because of its high flexibility in handling multimedia data under constant or variable bit rate channel requirements. An ATM network can support a variety of applications, ranging from video on demand to videoconferencing, that may have different requirements in tenns of bit rate, end-to-end delay, jitter delay, and quality of service. Although most aspects of ATM are well defined to form a standard for this new technology, the methodology for image and video transmission is still an open issue for further research.

An ATM network organizes data into fixed-size packets called cells. An ATM cell has a fixed length of 53 bytes which consists of 5 bytes of header and 48 bytes of payload. The header contains the necessary information for routing, payload type, cell loss priority, and cyclic redundancy check for error correction. The payload consists of an adaptation layer and the encoded bit stream, which may contain voice, image, video, or data. The adaptation layer specifies how the data is packed into the payload and can be designed to meet the desired requirements of different applications.

Besides jitter delay which is an important issue of concern in video transmission, an ATM based network may encounter three types of errors; *Type* **I:** bit errors in the payload, which may corrupt the data, *Type* **II:** bit errors in the header information: which may cause cell loss, and *Type* **III:** cell loss due to channel congestion or buffer overflow. These errors may result in severe visual distortions in the decoded images and video frames in the form of missing block(s) of data. The effects of cell defect on picture quality in conjunction with current image and video compression techniques have been studied in [127], and in [114, 115, 116, 117, 118, 119] a variety of prioritizing/interleaving techniques have been proposed to reduce the effect of ATM cell loss. Unfortunately, none of these techniques address the question of how the location of the missing data can be detected at the decoder. The information regarding the location of lost information is crucial to the effective concealment of errors by post-processing techniques.

In block-based compression techniques, the image is first partitioned into blocks of fixed or variable sizes. In variable block size algorithms such as the one introduced in this thesis, the image is segmented into blocks of *NxN* for N = 4, 8, 16, and 32. In the fixed block-based algorithms such as JPEG, H.261, and MPEG [1] the image is segmented into square blocks of 8x8 or 16x16. These blocks are then separately coded and grouped into Macroblocks (MBs), Group of Blocks (GOBs), and slices along with the proper synchronization and header information to form the encoded bit stream prior to transmission. As an example, the Common Intermediate Format (CIF) frame structure and the bit stream hierarchy for the H.261 video compression standard is illustrated in Fig. 7.2.



(a)

(b)

GOB: Group of blocks

MB: Macroblocks

Y: Luminance blocks **(8x8)**

C_b and C,: Chrominance blocks **(4x4)**

FLC: Fixed Length Code

VLC: Variable Length Code

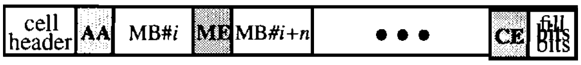
TCOEFF: Transform Coefficients

EOB: End Of Block:

Fig. 7.2. (a) H.261 CIF frame structure (luminance 288x352). (b) H.261 encoded video multiplex bit stream hierarchy.

An empirical study of different image and video compression techniques reveals that it is possible to pack more than one block of image or video data into an ATM cell. For example the JPEG baseline algorithm allocates an average of 4 bytes to each square block of 8x8 in a 512x512 gray scale image that has been encoded at 0.5 bits per pixel (bpp), and the H.261 algorithm allocates an average of 3.5 bytes to each MB for a CIF video clip encoded at a bit rate of 320 Kilo bits per second (Kbps).

In this approach to ATM cell packing, it is proposed to insert the important synchronization and header information into high priority cells and then pack an integral number of blocks or MBs into normal cells by a proper interleaving mechanism [116, 127, 128, 129, 10]. A simple interleaving technique is to insert every n^t encoded MB of alternate odd/even rows into a cell until it is filled. Moreover, we can encode and transmit the even and odd fields of a frame independently. If an integral number of MBs do not fit into a normal ATM cell, fill bits can be used after the cell end marker to pack that cell, as shown in Fig. 7.3. The interleaving step size n is also packed into high priority cells along with header information for guaranteed safe delivery to the decoder.



AA: Absolute Address
MB: Macroblocks
ME: Macroblock End
CE: Cell End

Fig. 7.3. Packing of normal ATM cells with interleaved MBs.

Although this ATM cell packing technique may require extra overhead and buffer size and result in extra delay, it offers the following major advantages: h

- It guarantees the safe arrival of the synchronization and header information.
- It localizes the loss of MBs within a frame.
- It forces the lost MBs to be isolated within a frame.

The first advantage prevents the loss of entire frame(s) during transmission while the last two are essential for the effective reconstruction of missing information via post-processing techniques. A detailed study of the interleaving mechanism, buffer constraints, and delay analysis is the subject of our future research. Error concealment with post-processing techniques is presented in following sections.

7.3 Spatial Error Concealment with Post-Processing Techniques

Once the location of a lost block is known to the decoder, post-processing techniques can be used in the transform or pixel domain to conceal the errors. Let M denote a missing block of size NxN, $\hat{x}_{i,j}$ the estimated value of the missing pixels in M, and $x_{i,j}$ the value of uncorrupted pixels in the neighboring blocks. The missing block M can have up to eight NxN neighbors B^k, k = {n, s, e, w, ne, nw, se, sw}, where the values of k correspond to the blocks at the north, south, east, west, north-east, north-west, south-east, and south-west of M, respectively. **A** variety of post-processing techniques could be used to approximate the missing information by using the available uncorrupted data in the neighboring blocks as explained in the following subsections.

7.3.1 Error Concealment with Replacement Techniques

In this class of techniques the missing block M in the decoded image or intracoded video frame is replaced by one of its available neighboring blocks in the same frame [130]. For video frames the missing block can also be replaced with the block of the previous frame which has the same physical location. This technique works reasonably well if the two adjacent blocks are highly correlated. But if M belongs to a region in the image across the edges or close to the boundary of different objects, the performance of this method is severely degraded and it produces visible distortion in that region. Moreover, when a lost block is replaced by an uncorrelated block from the previous video frame that has a relatively high contrast with the surrounding area, visually distracting flashes can appear in the decoded image sequence.

7.3.2 Error Concealment with Least Square Techniques

In this class of techniques the single-pixel wide boundary pixels of the four nearest neighbors adjacent to M are used to impose smoothing constraints on the reconstructed values inside the missing block [120, 122, 123, 11, 128]. **A** number of different cost functions can be used to impose smoothness constraints, but an appropriate cost function for smoothness measure in image processing applications is the sum of the weighted square differences between each lost pixel and its four nearest neighbors, as shown in Fig. 7.4.

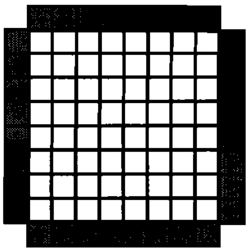


Fig. 7.4. An 8x8 missing block and the single-pixel wide boundaries of its 4.-nearest neighbors (shaded area) used to compute weights in the least squares techniques.

Let x be a vector of length N^2 composed of the samples in M arranged in a lexicographic order, and $\omega^{n}_{i,j}$, $\omega^{s}_{i,j}$, $\omega^{w}_{i,j}$, and $\omega^{e}_{i,}$, the weighting coefficients, then the cost function $J(x)$ is given by

$$J(x) = \frac{1}{2} \sum_{(i,j) \in M} [\omega^{n}_{i,j} (\hat{x}_{i,j} - \hat{x}_{i-1,j})^2 + \omega^{s}_{i,j} (\hat{x}_{i,j} - \hat{x}_{i+1,j})^2 + \omega^{w}_{i,j} (\hat{x}_{i,j} - \hat{x}_{i,j-1})^2 + \omega^{e}_{i,j} (\hat{x}_{i,j} - \hat{x}_{i,j+1})^2] \tag{7.1}$$

where $\hat{x}_{i-s,j-s} = x_{i-s,j-s}$, $s = \{-1,1\}$, for pixels at boundaries of M adjacent to its neighbors. This cost function must be minimized *recursively* for the pixels at the boundaries of M toward its center such that the boundary information from adjacent blocks can be propagated into the interior of the missing block.

The above cost function is the well known least squares cost function and can be written in matrix form as

$$J(x) = \frac{1}{2} x^t Q x - x^t b + c \tag{7.2}$$

where

$$Q = \sum_{m=1}^{4} [Q^t_m Q_m + S_m - Q_m - Q^t_m] \tag{7.3}$$

$$b = \sum_{m=1}^{4} [S_m - Q^t_m S_m] b_m \tag{7.4}$$

and

$$c = \sum_{m=1}^{4} \boldsymbol{b}_m^t \boldsymbol{S}_m \boldsymbol{b}_m \tag{7.5}$$

The matrices **Q,** are lower and upper diagonal matrices with zero entries along the main diagonal such that **Q,** $= \boldsymbol{Q}_2^t$ and $\boldsymbol{Q}_3 = \boldsymbol{Q}_4^t$. The matrices $\boldsymbol{S}_m$ are diagonal matrices with entries $\omega_{i,j}^k$ for the north, south, west, and east directions along the main diagonal. The vectors **b,** contain the values from the adjacent border pixels of M and zeros corresponding to the lost pixels. The proper choice of weighting coefficients guarantees the positive definiteness of **Q** [120] and the optimal least square solution is given by

$$\hat{\boldsymbol{x}}_{opt} = \boldsymbol{Q}^{-1}\boldsymbol{b} \tag{7.6}$$

Alternatively, instead of the above matrix inversion approach, we can use an iterative gradient descent algorithm to find the optimal solution. If we let $\boldsymbol{g}_l = \boldsymbol{Q}\hat{\boldsymbol{x}}_l - \mathbf{b}$ be the l^{th} iteration of the gradient vector of $\boldsymbol{J}(\boldsymbol{x})$, then an iterative solution for $\hat{\boldsymbol{x}}_{opt}$ can be obtained via the steepest descent algorithm [49] by

$$\hat{\boldsymbol{x}}_{l+1} = \hat{\boldsymbol{x}}_l - \frac{\boldsymbol{g}_l^t \boldsymbol{g}_k}{\boldsymbol{g}_l^t \boldsymbol{Q}\, \boldsymbol{g}_l}\, \boldsymbol{g}_l \tag{7.7}$$

It is important to note that in this method in order to guarantee that the estimated pixel values fall within the dynamic range of the original image, they should be truncated to the allowed minimum and maximum values after each iteration.

It is possible to reduce the computational complexity of the least squares concealment of the lost pixels by computing a single weighting coefficient in the north, south, west, and east directions [123]. In this technique the four weighting coefficients are obtained by minimizing the squared errors between the single-pixel wide boundaries of the missing block and its four nearest neighbors, and the reconstructed block is given by

$$\hat{\boldsymbol{M}} = \omega^n \boldsymbol{B}^n + \omega^s \boldsymbol{B}^s + \omega^w \boldsymbol{B}^w + \omega^e \boldsymbol{B}^e \tag{7.8}$$

where ω^n, ω^s, ω^w, and ω^e are the weighting coefficients corresponding to the four neighboring blocks. If all the weighting coefficients are set to 0.25, the reconstructed block is simply the average of its four nearest neighbors.

Least square concealment techniques are computationally efficient and perform well when the neighboring blocks of the lost cells are strongly correlated. However, their performance is degraded when the missing block falls across the boundaries of objects or where multiple edges or details are present in the region.

7.3.3 Error Concealment with Bayesian Techniques

In this class of techniques, the decompressed image is modeled as a Markov Random Field (MRF) or Gauss Markov Random Field (GMRF) [131] stochastic process, and concealment is achieved by using a Maximum A-Posteriori (MAP) estimator [132]. Let X and Z be discrete parameter MRFs representing the decoded N_1 x N_2 image and the decoded image with missing blocks of data, respectively. Let x_i and z_i be the lexicographic ordering of the i^{th} block in X and Z. If there are m blocks of N x N in X and Z then each image can be represented by vectors x and z as $x = [x_1^t\ x_2^t \ldots x_m^t]^t$, and $z = [z_1^t\ z_2^t \ldots z_m^t]^t$. T and Z are related by

$$z = Tx \tag{7.9}$$

where T is a transformation matrix of size $(N_1N_2 - nN^2)$ x N_1N_2 that is constructed by removing N^2 consecutive rows of an identity matrix, and n is the total number of missing blocks. Let $f(x|z)$ denote the conditional probability density function of x given z, then the MAP estimate of x can be computed as

$$\hat{x}_{map} = arg \max_{x|z = Tx} L(x|z) \tag{7.10}$$

where $L(x|z) = ln f(x|z)$ is the log-likelihood function given by Baye's rule as

$$L(x|z) = ln f(x) + ln P(z = Tx|x) - ln f(z) \tag{7.11}$$

the third term in the above equation is independent of x, and the probability of $z = Tx$ given x is equal to one when $z = Tx$ and zero otherwise, thus the MAP estimate of x can be obtained by

$$\hat{x}_{map} = arg \min_{x|z = Tx} [-\ln f(x)] \tag{7.12}$$

Since the original image X is modeled as an MRF, the probability density function of $\boldsymbol{x}$ can be defined as [131]

$$f(\boldsymbol{x}) = \frac{1}{P}\exp\left(\sum_{c \in C} V_c(\boldsymbol{x})\right) \tag{7.13}$$

where P is a normalizing constant known as the partition function, $V_c(\boldsymbol{x})$ is a potential function defined over a collection of connected points *c* called cliques, and *C* is the set of all cliques over which the potential function is defined. Using the probability density function in (7.13) the MAP estimate of $\boldsymbol{x}$ can be computed by

$$= \arg \min_{x|z = Tx}\left[\sum_{c \in C} V_c(\boldsymbol{x})\right] \tag{7.14}$$

The above optimization problem can be solved by means of iterative algorithms such as :simulated annealing [131], conjugate gradient [133], or iterative conditional modes [134]. Although Bayesian techniques have been successfully used for image segmentation and restoration [131], their utilization for error concealment does not offer a clear advantage. Their performance is highly correlated to the choice of potential function and they are computationally expensive. Moreover, on average they do not yield better results than the least square or polynomial interpolation techniques in terms of subjective quality of the concealed images.

7.3.4 Error Concealment with Polynomial Interpolation

In this class of techniques polynomial interpolation algorithms such as spline based interpolation [135] or wavelet based interpolation [136] can be used to approximate the values of the pixels in M. Interpolation with higher order polynomials is computationally expensive and on average does not show a significant improvement over the binomial interpolation. In bilinear interpolation every pixel in M is reconstructed by using the values in the single-pixel wide boundary pixels of the four nearest neighbors adjacent to M. The bilinear interpolation can be written as

$$\hat{x}_{i,j} = \lambda[(1-\mu_1)x_{i,-1} + \mu_1 x_{i,N}] + (1-\lambda)[(1-\mu_2)x_{-1,j} + \mu_2 x_{N,j}] \tag{7.15}$$

where $\mu_1 = d_1$ / N, $\mu_2 = d_2$ / N, and λ is a weighting factor which determines the contributions of the neighboring blocks on either side of M as shown in Fig. 7.5.

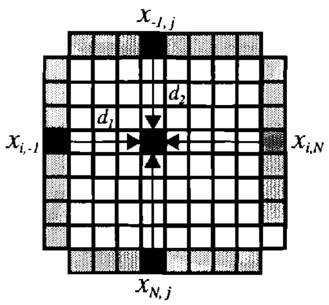


Fig. 7.5. The four closest pixels used in Bilinear interpolation.

In general, this method is computationally simple and results in good reconstruction when the missing block belongs to a relatively homogenous surface or smoothly varying regions. However, its performance is not as good around corner edges or for regions with multiple objects.

7.3.5 Error Concealment with Edge-Based Techniques

In this class of techniques the gradient information [124, 137] or the binary pattern [32] in the neighboring blocks is used to exploit the local geometrical structure of the image by hypothesizing single or multiple edges passing through the missing block. Finally, the reconstruction is achieved by interpolation along the edges. In [137] the magnitude and angular direction of the gradient for each pixel in the surrounding neighbors of a missing block is computed with a Sobel operator. Then each pixel is classified into one of eight quantized edge directions $22.5^{o}i$, $i = 0, 1, \ldots, 7$. Then by using a threshold value, a voting mechanism is employed to determine the direction of possible edges that might pass through the missing block. If the algorithm detects an edge through the missing block, it tries to align the interpolation in the direction of the edge using the linear weighted sum of pixels or Projection Onto Convex Sets (POCS). If multiple edges are detected, the algorithm interpolates the missing block along each direction separately. These multiple images are then mixed together according to some heuristic rule. This method performs well when the missing block can be characterized by a single dominant edge direction or when the surrounding pixels of the missing block contain highly correlated edge information. However, its performance is not as good in regions that are textured or have multiple edges. In addition, the gradient edge operators are very sensitive to noise and may produce false edges, thus causing classification errors. Moreover, this algorithm is computationally expensive, especially when the POCS algorithm is used for spatial interpolation.

In [138] a two-pixel wide binary pattern around the missing block is converted to a binary pattern via thresholding. This binary pattern is then used along with some heuristic rules to hypothesize straight edges through the missing block. This method works better than the former technique when corner or multiple edges are present and is computationally simpler. However, its performance degrades when the size of the missing block is greater than 8x8, because some of the heuristic rules used for hypothesizing straight edges fail to hold for larger blocks of missing data.

7.3.6 Error Concealment with Multi-Directional Recursive Nonlinear Filtering

The majority of post-processing error concealment techniques are based on the assumption of statistical correlation of neighboring MBs, but most of them fail to explicitly exploit the spatial structure of the missing information. Therefore, if the missing block and its neighbors belong to a homogeneous region, then any of the above methods will produce a good approximation for the lost information. However, if the missing block lies on the border of different objects or multiple edges, most of these methods produce faulty estimates. Only in the edge based techniques, structural properties of the neighboring blocks are used to interpolate along hypothetical edges that may pass through the missing block, thus reconstructing the missing information. However, these techniques are either computationally expensive or fail to produce reliable estimates in textured regions or when multiple edges are present, as was explained in the previous subsection. In general, a good interpolation scheme must have the following properties:

- The interpolation should exploit the correlation structure of neighboring blocks.
- The interpolation should be multi-directional.
- The interpolation scheme should be robust.
- The interpolation scheme should be computationally efficient.

In this novel approach, a multi-directional recursive nonlinear filtering (MRNF) scheme with variable filter kernels is used to achieve the above requirements [11, 15]. The reconstruction is started from all directions at the boundaries of the missing block M, and the values of the missing pixels are recursively estimated toward the center of the block. The multi-directional filter kernels or processing windows are chosen to exploit the structural correlation of the missing block with its neighbors while the robust nonlinear WSS-GMLOS filter [12, 13] exploits the statistical correlation of the neighboring MBs.

The support of the processing windows is chosen according to the local statistics of the neighboring pixels within a 3x3 square window based on a bimodal distribution assumption similar to [139]. Two possible filter kernels for multi-directional filtering of the neighboring blocks are shown in Fig. 7.6. The darker squares correspond to the available pixel(s) at the opposite side of the missing MBs and are used to exploit the cross correlation of the interpolating MBs on opposite sides of M.

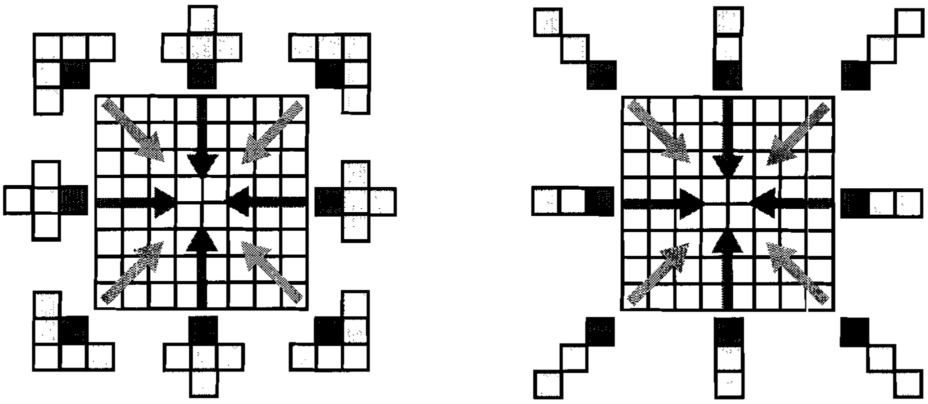


Fig. 7.6. Examples of processing windows or filter kernels for multi-directional recursive nonlinear filtering of the missing blocks. The darker shades correspond to pixels on the opposite sides of the missing block.

Once the processing window W for a missing pixel at the boundary of M is selected, the WSS-GMLOS filter can be used to estimate the value of the missing pixel. Let x be a vector of length n composed of the samples in W arranged in lexicographic order, and $x_R = \{x_{(i)}: i = 1,\ldots,n)$ be the rank ordered vector of samples in x arranged in an ascending order, then the missing pixel can be estimated as

$$\hat{x} = \frac{1}{s} \sum_{x_{(i)} \in W^I} \omega_i x_{(i)} \qquad (7.16)$$

where W^I is the inlier set for W, ω_i is the weight associated with the i^{th} sample of W^I, and $s = \sum \omega_i$. We assume that the members of the inlier set are degraded with a zero mean and uncorrelated additive noise process with variance δ_n^2, such that

$$\omega_i = \frac{1}{\delta_n^2} \qquad (7.17)$$

After reconstruction of the boundary pixels in M with multi-directional WSS-GMLOS filters, the estimation process is continued recursively toward the center of the missing block until all the samples in M are reconstructed. The flowchart of our MRNF algorithm is shown in Fig. 7.7.

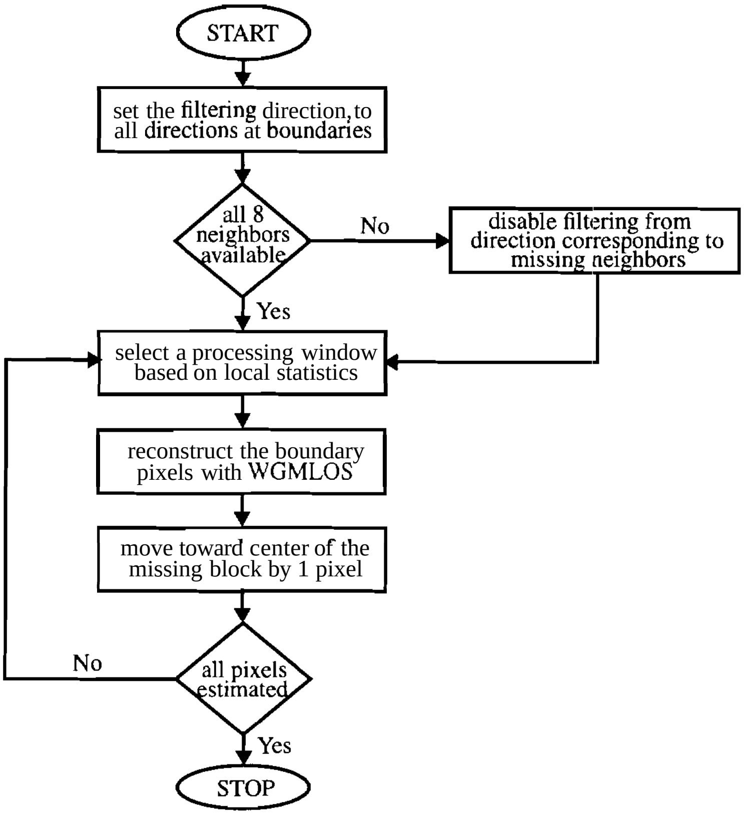

START
set the filtering direction, to all directions at boundaries
all 8 neighbors available
No
disable filtering from direction corresponding to missing neighbors
Yes
select a processing window based on local statistics
reconstruct the boundary pixels with WGMLOS
move toward center of the missing block by 1 pixel
all pixels estimated
No
Yes
STOP

Fig. 7.7. The flowchart of MRNF error concealment algorithm.

The MRNF technique is a computationally efficient algorithm that is capable of exploiting both geometrical and statistical structure of the missing pixels with their immediate neighbors in each direction for effective concealment of errors due to cell loss. Moreover, the robustness of the WSS-GMLOS filter could eliminate the effect of false edge pixels in the process of interpolation.

7.4 Concealment of Temporal Information

In block-based motion compensated video compression techniques a hybrid approach based on the concept of ***intraframe*** and *interframe* coding is being used. In the intraframe mode of operation, spatial redundancy is exploited by segmenting a single frame into many small blocks that are then transformed, quantized, and encoded similar to a still image. In the interframe mode of operation, temporal redundancy is exploited by using motion compensation to generate a prediction of the current video frame from previous and possibly from future frames. The difference between this prediction and the actual frame is then transformed, quantized, and coded along with motion vector information. Finally, the intraframe and interframe information as well as the synchronization and other side information is structured into a sequence of bit streams for storage or transmission. In this class of video compression techniques a single lost block may affect the subjective quality of more than one block in the future frames, even if no temporal information is lost, as illustrated by an error propagation tree in Fig. 7.8. clearly, the loss of temporal information will further degrade the quality of the decoded bit stream. Therefore, both spatial and temporal reconstruction algorithms are needed in video error concealment.

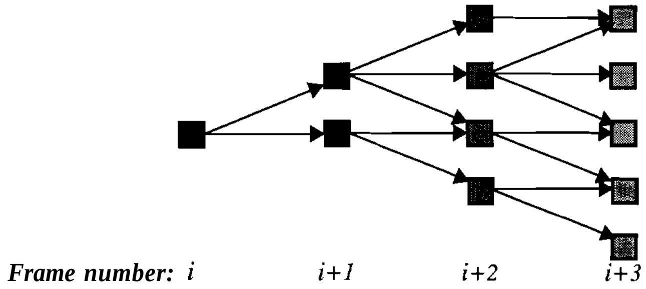


Fig. 7.8. An example of block error propagation into the future frames in motion compensated video compression.

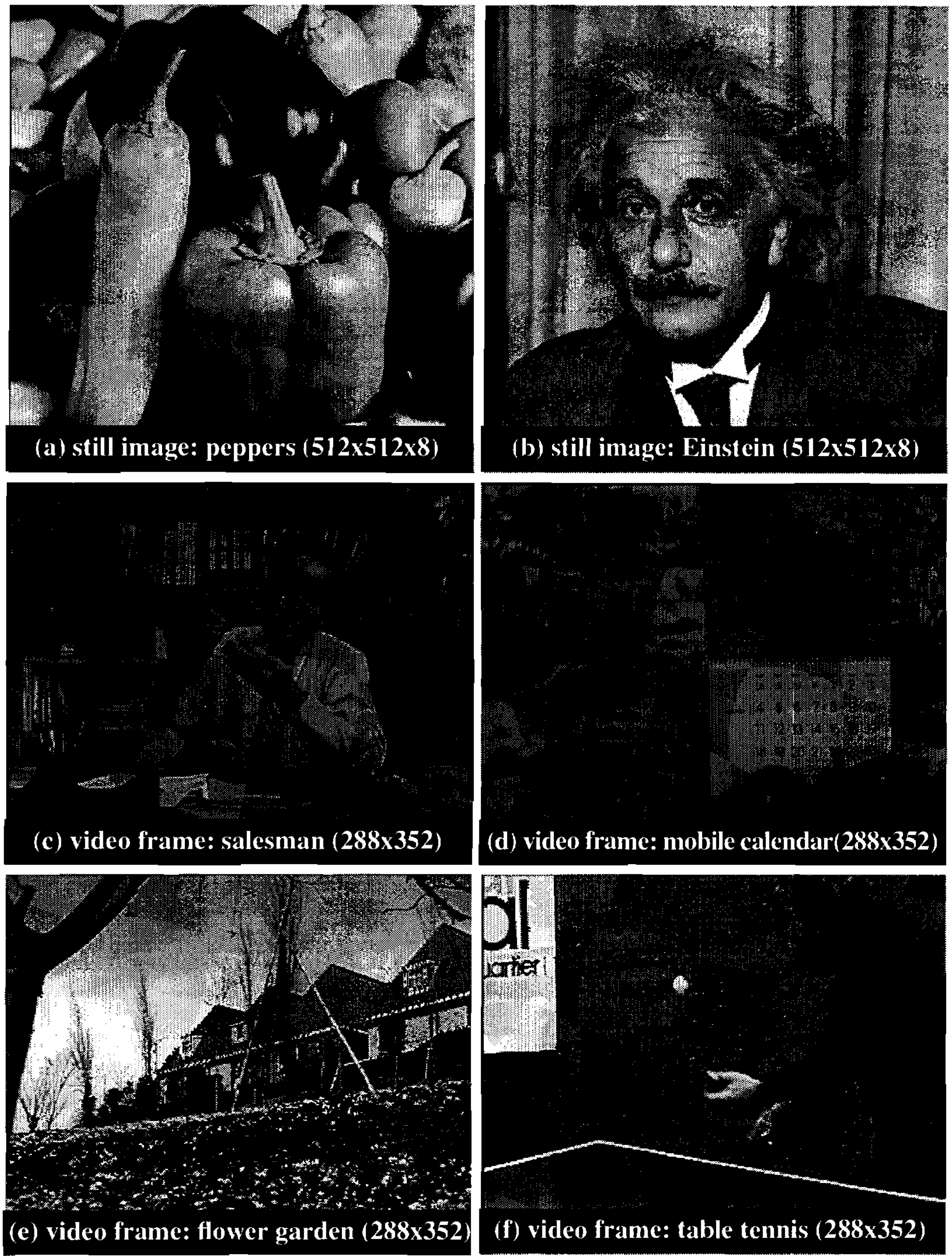
The errors in the intraframe coded frames can be concealed by using the spatial error concealment techniques of the previous subsection. In the interframe mode of operation, the encoded bit stream normally contains predicted and bi-directionally interpolated frames. When temporal information is lost, the decoder has to generate an estimate for the prediction direction (forward, backward, or interpolative) and the missing motion vectors. The experimental results on different test sequences demonstrated that the components of the adjacent motion vectors have intraframe correlations that range between 0.1 to 0.45. Moreover, the interframe correlation between the neighboring motion vectors in temporal direction was much higher in scenes with little motion but considerably smaller in scenes with rapid motion. These correlations suggest that both intraframe and interframe motion vectors could be used to estimate the lost motion vector components.

The simplest temporal reconstruction algorithm is the temporal replacement technique [118, 125]. In this method temporal information from the previous frame is used to replace the lost temporal information. This technique works well in areas with no motion or small motion but fails when moving objects with moderate or high motion are present at the scene. It is possible to achieve a better temporal concealment by using a least squares cost function based on a smooth movement assumption across the successive frames [118, 120, 123, 129, 140]. In this class of techniques, the average or weighted average of motion vectors in the surrounding or possibly previous frame MBs are used to reconstruct the lost temporal information. These techniques work well in areas with no motion, small motion, or moderate motion but fail when multiple moving objects or high motion is present at the scene. In [141] the median filter is used to reconstruct the lost motion vector information. This algorithm performs better than the least squares techniques at the boundaries of moving objects where motion vectors may point in different directions, however its performance is inferior to that of the least squares techniques in regions with small and moderate motion components. A few temporal concealment techniques based on motion compensation at the encoder or receiver are also investigated in [125], but these method are either computationally expensive or the overhead information associated with them may not fit into the available channel bandwidth. A Bayesian approach to the recovery of the motion vector is also proposed in [132]. This method is computationally expensive and its performance is highly sensitive to the potential function used for the Bayesian estimation.

Our experimental results showed that the replacement of lost motion vectors with the output of the GMLOS filter operating on the motion vectors in the neighboring MBs within the current frame yield better subjective image quality than the averaging or Bayesian techniques. This is partly due to the fact that near the boundaries between two objects the corresponding motion vectors may point in different directions, in which case averaging or Bayesian techniques may produce an unreliable estimate for the missing motion vector components. In addition, our algorithm performs better than median based techniques in areas with small or moderate motion vector components. The use of the motion vector components of the previous frame along with the GMLOS filter for reconstructing the missing temporal information of the current frame was also studied. However, this technique produces unreliable estimates for motion vector components near the boundaries of multiple objects and results in visual distortions in the form of high contrast discontinuities along the edges across different objects. The experimental results of using this method on real images and video clips are provided in the following section.

7.5 Experimental Results and Conclusions

The simulations were carried out on various test images and video sequences. The still images were grayscale 512x512, and 8 bits per pixel (bpp). The video sequences were in CIF (luminance 288x352) format [1], with luminance frame size of 288x352, and 100 frames long. Some of the test images are shown in Fig. 7.9. JPEG and QTMP compression algorithms were used for still images. For the video sequences H.261 at a bit rate of 320 Kbps, and MPEG at 1.5 Mbps with an interframe interval of 1 to 10 frannes were used [1]. The ATM cell loss was simulated with an even/odd interleaving scheme and a random loss pattern. The performance of the algorithm was only studied for the luminance component of the test video sequences because the human visual system is less sensitive to errors in the chrominance components. The experimental results confirmed that when a reconstructed video clip is viewed at a full frame rate (30 frames per second), the luminance errors are clearly visible, while the chrominance errors are virtually imperceptible.



(a) still image: peppers (512x512x8)

(b) still image: Einstein (512x512x8)

(c) video frame: salesman (288x352)

(d) video frame: mobile calendar(288x352)

(e) video frame: flower garden (288x352)

(f) video frame: table tennis (288x352)

Fig. 7.9. Test images: some of the still images and video sequences used in our simulations. The still images are 512x512x8 and the video sequences are CIF with luminance:; of size 288x352.

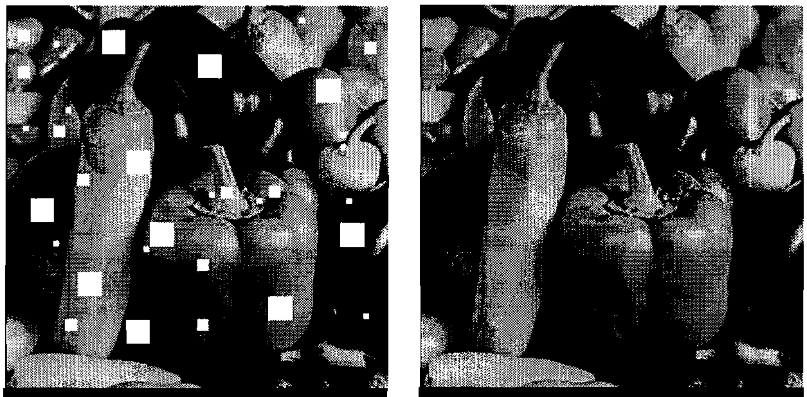
The peak signal-to-noise ratio (PSNR) is not always a reliable measure of perceived quality, thus the performance of the algorithm was measured with both PSNR and subjective image quality.

The test image *Peppers* of Fig. 7.9(a) was encoded at 0.5 bpp (PSNR = 36.15 dB) with the QTMP compression algorithm and packed into ATM cells. The reconstructed image with 10% block loss and no error concealment is shown in Fig. 7.10(a). The test image Einstein of Fig. 7.9 (b) was encoded at 0.5 bpp (PSNR = 29.84 dB) with the JPEG algorithm and packed into ATM cells. The reconstructed images with 5% block loss and 25% block loss and no error concealment are shown in Fig. 7.10(b) and Fig. 7.10(c), respectively. In addition to annoying visual artifacts in the form of missing blocks, the PSNR of the reconstructed images dropped by as much as 15 dB. MNIRF-GMLOS error concealment was used to conceal the errors in these test images, and the results are shown in Figs. 7.10(b), (d), and (f). The PSNR values for bilinear interpolation (BLI), constrained least squares (CLS), Bayesian interpolation (BI), and MNRF-GMLOS concealment algorithm for the test images are listed in Table 7.1.

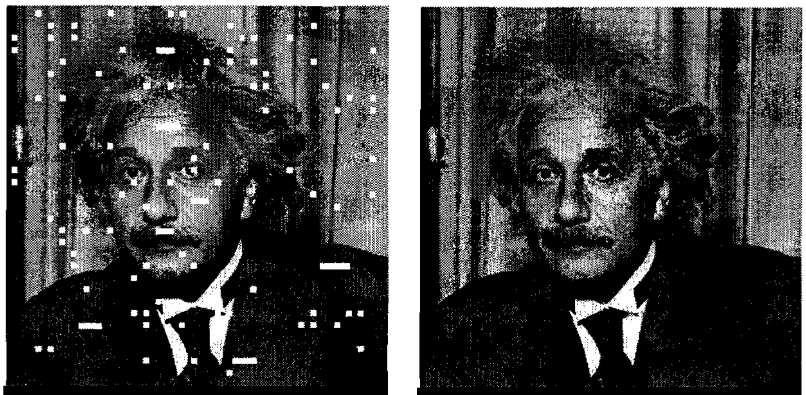
Table 7.1
PSNR (in dB) for different spatial concealment techniques.

Image (% block loss)	**Corrupted Image**	**Spatial Error Concealment Techniques**			
		BLI	**CLS**	**BI**	**MRNF**
peppers (10%)	20.17	34.83	34.85	34.68	34.94
Einstein (5%)	22.37	29.25	29.12	29.15	29.78
Einstein (25%)	13.75	27.69	27.25	27.02	28.65

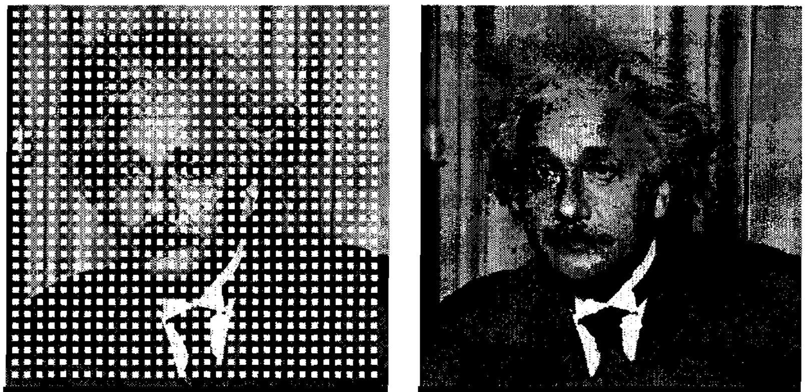
The proposed algorithm was also tested on the *salesman, mobile calendar, flower gal-den*, and *table tennis* image sequences. One hundred frames of each image sequence were encoded by using a H.261 encoder at 320 Kbps and an MPEG encoder at a bit rate of 1.5 Mbps. The encoded bit streams were then packed into ATM cells. The H.261 intracoded frame 128 of the salesman sequence (PSNR = 38.83 dB) is shown in Fig. 7.11(a), and the same frame with 5% MB loss and PSNR of 16.20 dB is shown in Fig 7.11(b).



(a) QTMP@0.5 bpp, block loss:10% (b) MRNF-GMLOS, PSNR=34.94dB



(c) JPEG@0.5 bpp, block loss:5% (d) MRNF-GMLOS, PSNR=29.78dB



(e) JPEG@0.5 bpp, block loss:25% (f) MRNF-GMLOS, PSNR=28.65dB

Fig. 7.10. Concealment of still images: (a) Peppers encoded with QTMP, and 10% block loss, PSNR = 20.17 dB. (b) Concealed peppers with MRNF-GMLOS, PSNR = 34.94 dB. (c) Einstein encoded with JPEG and 5% block loss, PSNR = 22.37 dB. (d) Concealed Einstein with MRNF-GIMLOS, PSNR = 29.78 dB. (c) Einstein encoded with JPEG and 5% block loss, PSNR = 13.75 dB. (d) Concealed Einstein with MRNF-GMLOS, PSNR = 28.65 dB.

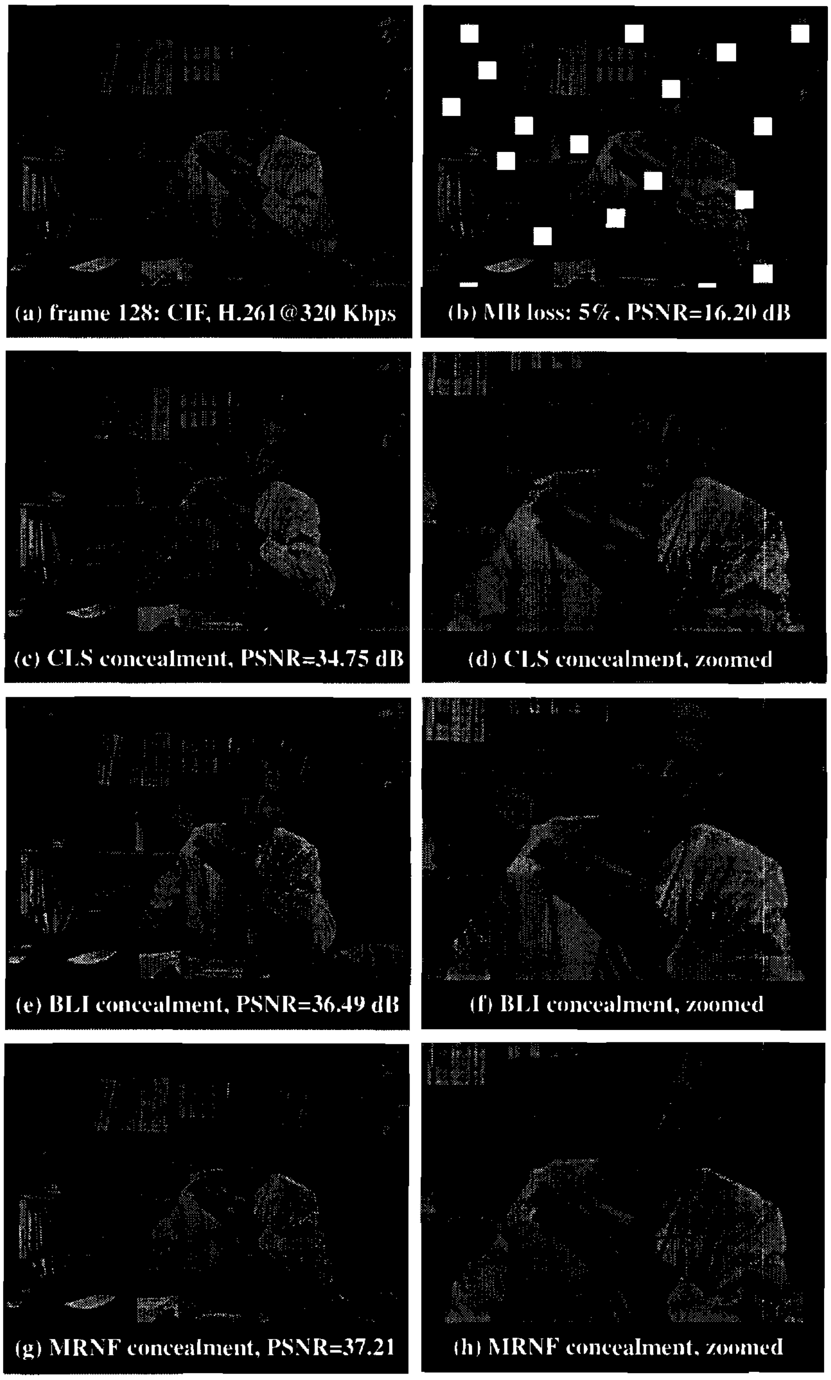

(a) frame 128: CIF, H.261@320 Kbps
(b) MB loss: 5%, PSNR=16.20 dB
(c) CLS concealment, PSNR=34.75 dB
(d) CLS concealment, zoomed
(e) BLI concealment, PSNR=36.49 dB
(f) BLI concealment, zoomed
(g) MRNF concealment, PSNR=37.21
(h) MRNF concealment, zoomed

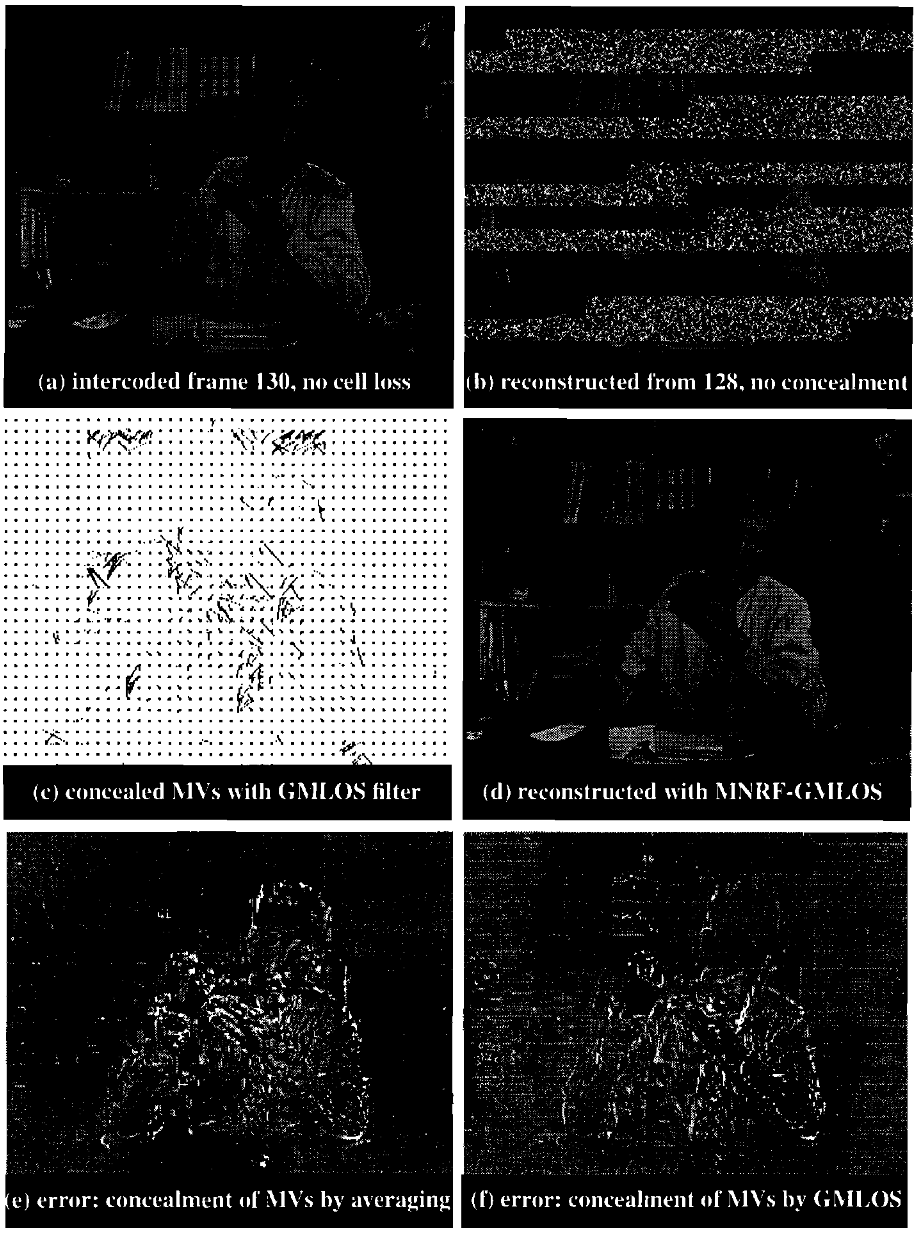
Fig. 7.11. Spatial concealment of video frames: salesman frame 128, H.261 at 320 Kbps, (a) Intracoded, PSNR = 38.83 dB, (b) 5% of the MBs are lost, PSNR = 16.20 dB. (c) Concealed with constrained least squares (CLI), PSNR = 34.75 dB. (d) Zoomed version of (c). (e) Concealed with bilinear interpolation (BLI), PSNR = 36.49 dB. (f) Zoomed version of (e). (g) Concealed with MRNF-GMLOS, PSNR = 37.21 dB. (h) Zoomed version of (g).

The reconstructed versions of corrupted frame 128 when concealed with the CLS, BLI, and MRNF-GMLOS techniques are shown in Figs. 7.11(c), (e), and (g), respectively. In order to evaluate the subjective quality of the concealed frames, their zoomed versions are also included in Fig. 7.11. The PSNRs of the concealed salesman frame 128 for these techniques and the BI concealment technique are listed in Table 7.2. The H.261 intercoded frame 130 of the same sequence (PSNR = 36.95 dB) that was predicted from frame 128 with no errors is shown in Fig. 7.12(a). The reconstructed frame 130 with no concealment and 5% loss of spatial and MV information is shown in Fig. 7.12(b). The concealed motion vectors with inter/intra frame GMLOS filtering of neighboring motion vectors and the reconstructed frame 130 with the MNRF-GMLOS spatio-temporal concealment algorithm is shown in Figs. 7.12(c) and (d), respectively. The difference or error image of frame 130 when lost temporal information is concealed by averaging is shown in Fig. 7.12(e). Finally, the error image of frame 130 when GMLOS is used for temporal concealment is shown in Fig. 7.12(f). The PSNRs of the concealed salesman frame 130 for various spatio-temporal concealment techniques are listed in Table 7.2. Except for the proposed algorithm, the temporal information for frame 130 was concealed by simply averaging (AVE) the available motion vectors in the neighboring macroblocks.

Table 7.2
PSNR (in dB) for different concealment techniques acting on salesman test sequence.

Frame No., Type (% MB loss)	**Corrupted Frame**	**Spatio-Temporal Error Concealment Techniques**			
		BLI-AVE	**CLS-AVE**	**BI-AVE**	**MRNF-GMLOS**
128, intracoded(5%)	16.20	36.49	34.75	35.36	37.21
130, intercoded	14.65	34.76	34.09	34.49	35.17

To evaluate the performance of the ATM cell packing and spatio-,temporal MRNF-GMLOS algorithms for various cell loss rates, the average PSNR of 100 video frames of the four test image sequences were computed at various cell loss rates. The video clips were encoded with a MPEG software *codec* at a bit rate of 1.5 Mbps. The results for the MRNF-GMLOS and CLS-AVE concealment algorithms are shown in Fig. 7.13.



(a) intercoded frame 130, no cell loss

(b) reconstructed from 128, no concealment

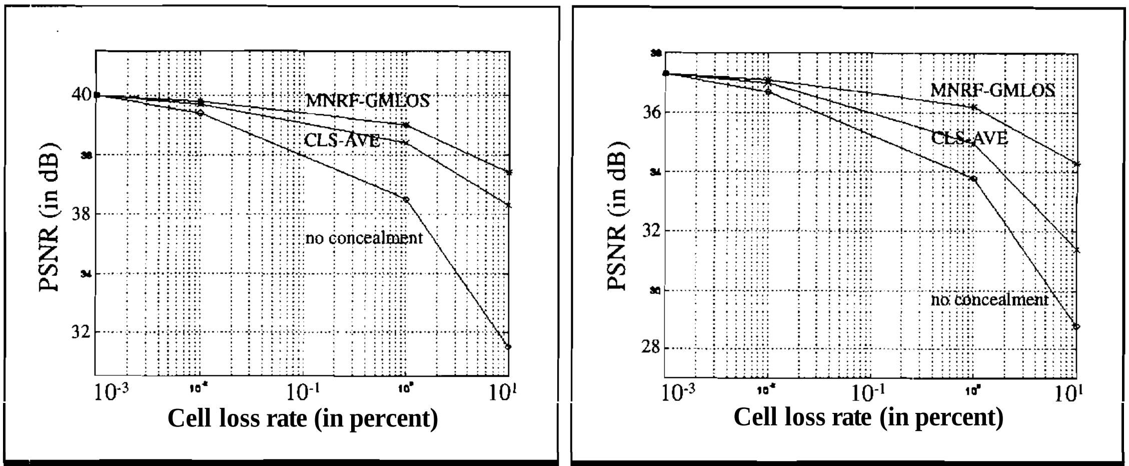
(c) concealed MVs with GMLOS filter

(d) reconstructed with MNRF-GMLOS

(e) error: concealment of MVs by averaging

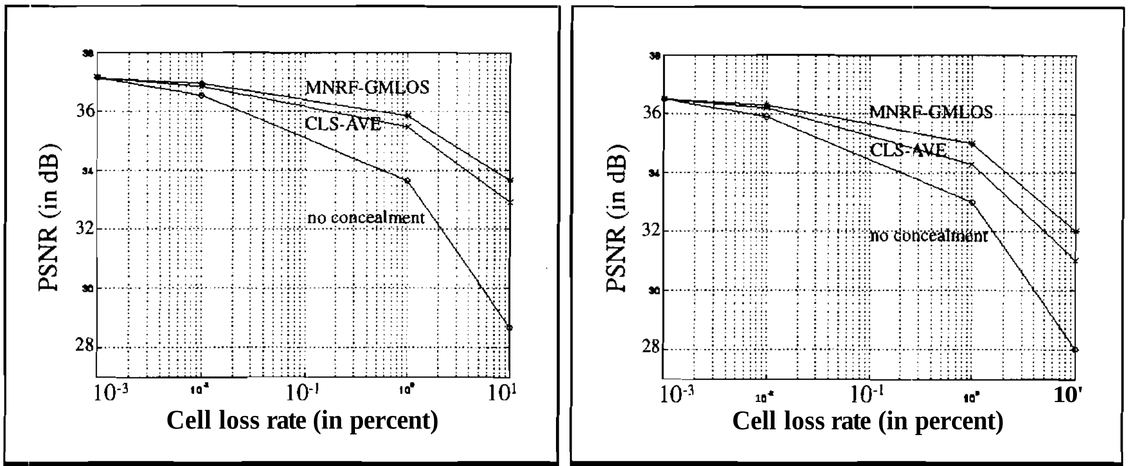
(f) error: concealment of MVs by GMLOS

Fig. 7.12. Spatio-temporal concealment of video frames: salesman frame 130, H.261 at 320 Kbps, (a) Intercoded using frame 128 with no errors, PSNR = 36.95 dB. (b) Reconstructed frame 130, no concealment with lost spatial and MV information. (c) Concealed MVs with inter/intra frame GMLOS filtering of neighboring MVs. (d) Reconstructed frame 130 with the MNRF-GMLOS spatio-temporal concealment algorithm, PSNR = 35.17 dB. (e) Error image when motion vectors are concealed by simple averaging. (f) Error image when the motion vectors are concealed by GMLOS filtering.



(a) salesman: 100 frames, MPEG@1.5Mbps

(b) mobile calendar: 100 frames, MPEG@1.5Mbps



(c) flower garden: 100 frames, MPEG@1.5Mbps

(d) table tennis: 100 frames, MPEG@1.5Mbps

Fig. 7.13. Spatio-temporal concealment coding gain vs. cell loss rate: MPEG ;at1.5 Mbps, PSNR averaged over the 100 frames of the test image sequences, concealed with MRNF-GMLOS and CLS-AVE algorithms. (a) Salesman sequence. (b) Mobile calendar sequence. (c) Flower garden sequence. (d) Table tennis sequence.

The above graphs support the conclusion that even under extreme channel congestion conditions, spatio-temporal post-processing concealment techniques can remarkably improve the quality of the decoded video streams. These techniques are specifically effective when the sequence does not contain large or complicated motion components. For example, as the cell loss rate increases the improvement in the concealed salesman sequence is greater than the other sequences that contain more complicated motion components.

The simulation results for still images, which are shown in Fig. 7.10 and Table 7.1, reveal that for small and isolated block loss patterns, the subjective and quantitative quality of the decoded images concealed with spatial error concealment is excellent. Moreover, most of the post-processing techniques have comparable performances under these conditions. The results of the spatio-temporal concealment of image sequences, which are shown in Figs. 7.11, 7.12, 7.13, and Table 7.2, reveal that these techniques can greatly improve the quality of the decoded video streams in presence of cell loss. It is noteworthy that the performance of intraframe spatial concealment is a function of the sequence content, with errors being mostly noticeable across boundaries of the objects, diagonal lines, corner edges, and highly detailed regions. The performance of interframe temporal concealment is also a function of scene contents, and degrades considerably in scenes with high and complicated motion components. The subjective evaluations showed that in most cases, when both spatial and temporal information are lost, the quality of the concealed sequences is good to acceptable, and as the cell loss rate increases only a few blatantly errant blocks may appear in the reconstructed video streams.

In this chapter a novel approach for error concealment of still images and image sequences in multimedia communication systems was presented. The experimental results demonstrated that the MRNF algorithm with WSS-GMLOS filter for spatial reconstruction, and the GMLOS filter for temporal reconstruction can effectively conceal the missing spatial and temporal information. In addition, the concealment algorithm is computationally efficient and can be implemented in real time with general purpose processors. Our ATM cell packing algorithm is simple and guarantees the safe delivery of important header, synchronization, and block loss location information. Moreover, the incorporated interleaving algorithm for cell packing mostly results in the loss of isolated blocks of data. This is an important factor in the effective concealment of errors via post-processing techniques.

8. CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

Signal expansion with adapted bases is a new field of interest in signal and image processing. In this thesis it was shown that the greedy Segmented Orthogonal Matching Pursuit (SOMP) is an adaptive signal expansion technique that can effectively exploit the sparse structure of signals. It was shown that the SOMP algorithm performs better than the original matching pursuit algorithm in terms of sparsity of representation and speed of convergence.

It was shown that the SOMP algorithm can be used for lossy image compression at low bit rates. Experimental results confirm that this technique is able to compress natural images at low bit rates, and its performance is better than the wavelet based algorithm at very low bit rates, and it is comparable to the best available compression techniques at higher bit rates. It was argued that the computational complexity of the compression algorithm can be justified due to the increasing computational power of digital computers in recent years. It was also shown that this projection based expansion technique can be used to unify the processes of separate prediction and residual coding into a unified framework for video compression applications. Moreover, it was shown that this algorithm performs better than the traditional hybrid block-matching MEMC algorithms at lower bit rates.

For the pre-processing and post-processing of visual information, a new robust nonlinear filter (GMLOS) was introduced. It was shown that this new filter can effectively improve the quality of degraded images and image sequences. It was also shown that this filter is an 1,-optimal order statistic filter and some of its properties were proved. **A** novel algorithm based on wavelet decomposition, variable size kernel GMLOS filters, and soft thresholding for removing blocking effects in block-based transform coding techniques was introduced. It was shown that this algorithm is easy to implement and can effectively remove blocking effects.

A simple algorithm for the packing of visual information into ATM cells was introduced. It was shown that this algorithm is well suited to effective error concealment with post-processing techniques. Finally, a novel error concealment technique was introduced based on the Multi-directional Recursive Nonlinear Filtering (MRNF) of lost information with the GMLOS filter was introduced. Experimental results confirmed that this new technique can effectively conceal errors at low to moderate cell loss rates.

8.2 Future Work

The performance of the SOMP signal expansion algorithm can be improved in a number of ways. For example, it would be interesting to consider a more flexible representation by allowing arbitrary segments, using dynamic programming techniques. Flexible segmentation can be particularly attractive for speech compression applications. The optimal quantization and coding of SOMP coefficients is also an interesting problem for future research. In particular, study on the distribution of the coefficients in the TF-plane can be used for the design of optimal quantizers. Another interesting problem is the design of optimal dictionaries by utilizing techniques from vector quantization literature. The design of adaptive orthogonal matching pursuit filter banks is also another interesting topic for future research. It is also possible to use the results of Chapters 4 and 5 to design a QSOMP video codec for low bit rate applications. Finally, theoretical study of the SOMP algorithm and its relation to the KL transform can be an interesting research topic.

The GMLOS filter can be improved and extended in a number of ways. It would be interesting to let the filter length vary according to local statistics of the processing window. It is also possible to explore a Bayesian cost function to improve the performance of the filter. The algorithm can also be used for design of nonlinear filter banks.

The cell packing algorithm should be studied in more detail, and its delay characteristics should be analyzed under different network interface scenarios. Finally, with the increasing demand for progressive video communication, it would be interesting to extend the error concealment algorithm to a multiresolution framework.

LIST OF REFERENCES

[1] V. Bhaskaran and K. Konstantinides, Image and video compression standards. Kluwer, 1995.

[2] M. Vetterli and J. Kovacevic, Wavelets and *subband* coding, Prentice Hall, 1995.

[3] K. Ramchandran, M. Vetterli, and C. Herley, "Wavelets, subband coding, and best bases," Proceedings of IEEE, vol. 84, no. 4, pp. 541-560, April 1996.

[4] J. Rissanen, "Universal coding, information, prediction, and estimation," IEEE Trams. on Information Theory, vol. 30, pp. 629-636, July 1984.

[5] H. R. Rabiee, R. L. Kashyap, and S. R. Safavian, "Adaptive multiresolution image cocling with matching and basis pursuits," in proceedings of *IEEE ICIP'96*, EPFL Switzerland, Sept. 16-19, 1996.

[6] A. N. Netravali and B. G. Haskell, Digital pictures: representation and compression, Plenum Press, 1989.

[7] A. M. Tekalp, Digital Video Processing, Prentice Hall, 1995

[8] H. R. Rabiee, R. L. Kashyap, and S. R. Safavian, "Multiresolution image coding with segmented orthogonal matching pursuit and wavelet packet dictionaries," Submitted to *IEEE* Trans. on Image Processing, Oct. 1996.

[9] H. R. Rabiee and R. L. Kashyap, "Video compression at low bit rates with segmented orthogonal matching pursuit," to be submitted to IEEE *Trans.* on Circuits and Systemsfor Video Technology, 1996.

[10] H. R. Rabiee and H. Radha, "Pre and post-processing algorithms for video cornpression applications," Technical Memorandum, *AT&T* Bell Labs., Indianapolis, Nov. 1993.

[11] H. R. Rabiee and H. Radha, "Post-processing algorithms for error concealment in video compression applications," Technical Memorandum, *AT&T* Bell Labs., Indianapolis, Dec. 1994.

[12] H. R. Rabiee and R. L. Kashyap, "GMLOS: a new robust nonlinear filter for image processing applications," Twenty Seventh Asilomar *Conf.* on *Signals,* Systems *& Computers*, Pacific Grove, CA, Nov. 1-3, 1993.

[13] H. R. Rabiee and R. L. Kashyap, "GMLOS: A new robust nonlinear filter based on the theory of generalized maximum likelihood estimation and order statistics," Accepted

for publication in IEEE Trans. on Image Processing, Sept. 1996.

[14] H. R. Rabiee and R. L. Kashyap, "De-blocking of block-based transform coded images with wavelet decomposition, GMLOS filtering, and soft-thresholding," to be submitted to IEEE Signal Processing Letters, 1996.

[15] H. R. Rabiee, H. Radha, and R. L. Kashyap, "Error concealment of still image and video streams with multi-directional recursive nonlinear filters," in proceedings of IEEE *ICIP*'96, EPFL Switzerland, Sept. 16-19, 1996.

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

[17] V. R. Algazi, Y. Kato, M. Miyahra, and K. Kotani, "Comparison of image coding techniques with a picture quality scale," in proceedings of SPIE, Applications of Digital *Image* Processing *XV*, vol. 1771, 1992.

[18] A. Gersho and R. M. Gray, *Vector* quantization and signal *compression*, Kluwer, 1992.

[19] D. Gabor, "Theory of communication," Journ. IEE, v01. 93, pp. 429:457, 1946.

[20] S. Mann and Simon Haykin, "Adaptive chirplet transform: an adaptive generalization of wavelet transform," Optical Engineering, vol. 31, pp. 1243-56, 1992.

[21] R. G. Baraniuk and D. L. Jones, "Shear madness: new orthonormal bases and frames using chirp functions," IEEE Trans. on Signal Processing, vol. 41, no. 12, pp. 3543-49, Dec. 1993.

[22] I. Daubechies, Ten lectures on wavelets, Philadelphia, SIAM, 1992.

[23] G. Strang and V. Strela, "Short wavelets and matrix dilation equations," IEEE Trans. on Signal Proccesing, vol. 43, no. 1, pp. 108-115, 1995.

[24] R. Coifman, Y. Meyer and M. V. Wickerhauser, "Wavelet analysis and signal processing," in M. B. Ruskai et al, editor, Wavelets and their applications, pp. 153-178, Jones and Barlett, Boston, 1992.

[25] G. Strang and T. Nguyen, Wavelets and *filter* banks, Wellesley Cambridge Press, MA, 1996.

[26] C. K. Chui, An introduction to wavelets, Academic Press, CA, 1992.

[27] Y. Meyer, Wavelets: algorithms and applications, SIAM, Pennsylvania, 1993.

[28] O.Rioul and M. Vetterli, "Wavelets and signal processing," IEEE ,Signal Processing *Magazine*, vol. 8, pp. 14-38, Oct. 1991.

[29] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE* Trans. on Communications, vol. 31, no. 4, pp. 532-540, April 1983.

[30] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," IEEE Trans. on *PAMI*, vol. 11, no. 7, pp. 674-693, July 1989.

[31] S. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $L_2(\boldsymbol{R})$," *Trans.* American Math. Society, vol. 315, pp. 69-87, Sept. 1989.

[32] S. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE* Trans. on Acoust., Speech, and Signal Processing, vol. 37, no. 12, pp. 2091-2110, Dec. 1989.

[33] O. Rioul, "A discrete-time multiresolution theory," IEEE *Trans.* on Signal *Processing*, vol. 41, no. 8, PP. 2591-2606, Aug. 1993.

[34] P. P. Vaidyanathan, Multirate systems *and filter* banks, Prentice Hall, 1993.

[35] M. Unser, P. Thevenaz, and A. Aldurabi, "Shift-orthogonal wavelet bases using splines," IEEE Signal Processing Letters, vol. 3, pp. 85-88, March 1996.

[36] H. R. Rabiee, R. L. Kashyap, S. R. Safavian, and M. Saeed, "Low bit rate image coding with shift orthogonal wavelet bases," to appear in proceedings of 30th Asilomar *Conference* on *SS&C*, Pacific Grove, CA, Nov. 1996.

[37] S. Mallat and Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries," *IEEE* Trans. on Signal Processing, special issue on Wavelets and *Signal* Processing, vol. 4.1, no. 12, pp. 3397-34.15, Dec. 1993.

[38] I. Daubechies, "Time-frequency localization operators: a geometric phase space approach," IEEE Trans. on Information Theory, vol. 34, no. 4, pp. 605-612, 1988.

[39] R. A. Devore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," IEEE Trans. on Information Theory, vol. 38, no. 2, pp. 719-746, March 1992.

[40] G. Davis, S. Mallat, and S. Avellanda, "Greedy Adaptive approximation," preprint, to appear in *Journal* of Constructive Approximation, 1996.

[41] G. Strang, Linear Algebra and its applications, Third edition., Harcourt Brace Jovanovich, CA, 1988.

[42] K. Gorpchenig, "Acceleration of the frame algorithm," IEEE Trans. on Signal Processing, special issue on Wavelets and Signal Processing, vol. 41, no.12, pp. 3331-3340, Dec. 1993.

[43] S. Chen and D. Donoho, "Basis Pursuit," in Proceedings of SPIE, Application of Wavelets in Signal and Image Processing, July 1995.

[44] R. R. Coifman and M.V. Wickerhauser, "Entropy-Based Algorithms for Best-Basis Selection," IEEE Trans. on Information Theory, vol. 38, no. 2, pp. 713-718, March 1992.

[45] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," IEEE Trans. on Image Processing, vol. 2, no. 2, pp. 160-175, April 1993.

[46] C. Herley, J. Kovacevic, K. Ramchandran, and M. Vetterli, "Tiling of the time frequency domain: construction of arbitrary orthogonal bases and fast tiling algorithms,"

IEEE Trans. on Signal Processing, special issue on Wavelets and *Signal* Processing, vol. 41, no. 12, pp. 3341-3359, Dec. 1993.

[47] R. L. de Queiroz and K. R. Rao, "Time-varying lapped transforms and wavelet packets", IEEE Trans. on Signal Processing, special issue on *Wavelets* and Signal *Processing*, vol. 41, no. 12, pp. 3293-3305, Dec. 1993.

[48] C. Herley and M. vetterli, "Orthogonal time-varying filter banks and wavelet packets," IEEE Trans. on Signal Processing, vol. 42, no. 10, pp. 2650-2663, Oct. 1994.

[49] D.G. Luenberger, Linear and Nonlinear Programming, 2nd Edition. Addison-Wesley Publishing Co., 1989.

[50] J. H. Friedman and W. Stuetzle, "Projection pursuit regression," J. American Statistics Association, vol. 76, pp. 817-823, 1981.

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

[52] L. K. Jones, "On a conjecture of Huber concerning the convergence of projection pursuit regression," The Annals of Statistics, vol. 15, no. 2, pp. 880-882, 1987.

[53] A.E. Jacquin, "Fractal Image Coding: A Review," Proceedings of *IEEE*, vol. 81, Oct. 1993.

[54] M. Kunt, A. Ikonomopoulos and M. Kocher, "Second Generation Image Coding Techniques," Proceedings of IEEE, vol. 73, April 1985.

[55] H. Samet, Application of Spatial Data Structures: Computer Graphics, Image *Processing,* and *GIS*, Addison Wesley, Reading, MA, 1990.

[56] J.M. Shapiro, "Embedded Image Coding Using Zerotress of Wavelet Coefficients," IEEE Trans. on Signal Processing, vol. 41, Dec. 1993.

[57] A. Said and W. Pearlman, "A new fast, ans efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. on C&S for Video Technology, vol. 6, no. 3, 1996.

[58] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint optimization of scalar and tree-structured quantization of wavelet image decomposition," in proceedings of 27th Asilomar Conference on *SS&C,* Pacific Grove, CA, Nov. 1993.

[59] G. Davis, "Self quantized wavelet subtrees: a wavelet based theory for fractal image compression," in proceedings of Data Compression *Conf.* '95, Snowbird, UT, 1995.

[60] S.R. Safavian, H. R. Rabiee, and M. Fardanesh, "Projection and Pursuit Image Compression," Accepted for publication in IEEE Signal Processing Letters, 1996.

[61] S. R. Safavian, H. R. Rabiee, and M. Fardanesh, "Adaptive Multiresolution Image Coding with Projection Pursuit Neural Networks," presented in 29th *Asilomar Conf.* in *SS&C*, Pacific Grove, CA, 1995.

[62] M. Fardanesh, S. R. Safavian, H. R. Rabiee, O. Ersoy, "Multiresulation Image Coding with Quadtree Segmentation, Projection Pursuit, and Neural Networks," submitted to IEEE Trans. on Circuit and Systems, 1996.

[63] H. R. Rabiee, R. L. Kashyap, and H. Radha, "Multiresolution Image Compression with BSP Trees and Multi-Level Block Truncation Coding," ZEEE 2nd *International Conf.* on .Image Processing, Washington D.C., Oct. 1995.

[64] H.R. Rabiee, R.L. Kashyap, and S.R. Safavian, "Mutiresolution Segmentation Based Image Coding With Hierarchical Data Structures," ZEEE *ICASSP'96*, Atlanta, GA, May 1996.

[65] H. Radha, R. Leonardi, M. Vetterli, and B. Naylor, "Binary Space Partitioning (BSP) Tree Representation of Images," Joumal of Visual Communication and Zmage Representation, Sept. 1991.

[66] X. Wu and Y. Fang, "A Segmentation-Based Predictive Multiresolution Image Coder," ZEEE Trans. Zmage Processing, vol. 4, Jan. 1995.

[67] E P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," ZEEE Trans. on Information Theory, vol. 38, no. 2, 1992.

[68] J. Vaisey and A. Gersho, "Image Compression with Variable Block Size Segmentation," ZEEE Trans. Signal Processing, vol. 40, no. 8, Aug. 1992.

[69] R. Neff, A. Zakhor, and M. Vetterli, "Very low bit rate video coding using matching pursuits," in proceeding of SPZE conference on Visual Comm. and Zmage Processing," vol. 2308, no. 1, pp. 47-60, 1994.

[70] T. S. Huang, Ed., Topics in Applied Physics, Two-dimensional Digital Signal *Processing* 11, Springer Verlag, Berlin, 1981.

[71] W. K. Pratt, Digital Zmage Processing, Wiley, New York, 2nd *Edition*, 1991.

[72] R. Chellappa, Ed., Digital Zmage Processing, IEEE Computer Society Press, 1992.

[73] I. Pitas and A. N. Venetsanopoulos, Nonlinear Digital Filters, Kluwer Academic Publishers, 1990.

[74] N. Wiener, Nonlinear Problems in Random Theory, The Technology Press, MIT and Wiley, New York, 1958.

[75] I. Pitas and A. N. Venetsanopoulos, "Order statistics in digital irnage processing," *Proceedings* of the ZEEE, vol. 80, no. 12, Dec. 1992.

[76] H. R. Rabiee and R.L. Kashyap, "GMLOS and a comparative study of nonlinear filters," ZEEE ZCASSP '94, Adelaide, South Australia, April 19-22, 1994.

[77] R. C. Hardie and C. Boncelet, "Gradient-based edge detection using nonlinear edge enhancing prefilters," Preprint, to appear in IEEE Trans. on Zmage Processing, 1995.

[78] A.V. Oppenheim and R.W. Schafer, Discrete Time Signal Processing, Prentice Hall,

1989.

[79] I. Pitas and A. N. Venetsanopoulos, "Nonlinear mean filters in image processing," *IEEE* Trans. on ASSP, vol. 34, no. 6, June 1986.

[80] M. Schetzen, The Volterra and Wiener Theories of Nonlinear Systems, Wiley, 1980.

[81] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, 1982.

[82] J. Serra, Ed., Image Analysis and Mathematical Morphology: Theoretical Advances, vol. 2, Academic Press, 1988.

[83] F. Hampel, E. Ronchetti, P. Rousseevw, and W. Stahel, *Robust* Statistics: An Approach Based on *Influence* Functions, Wiley, New York, 1986.

[84] P.S. Huber, Robust Statistics, Wiley, New York, 1981.

[85] Y. H. Lee and S. A. Kassam, "Generalized median filtering and related nonlinear filtering techniques," IEEE Trans. on ASSP, vol. 33, no. 3, pp. 672-683, 1985.

[86] C. A. Pomalaza-Raez and C. D. McGillem, "An adaptive, nonlinear edge-preserving filter," IEEE Trans. on ASSP, vol. 32, no. 3, pp. 571-576, 1984.

[87] J. L. Hodges Jr. and E. L. Lehmann, "Estimates of location based on rank tests," Ann. Math. Statist., vol. 34, pp. 598-611, 1963.

[88] R. J. Crinon, "The Wilcoxon filter: a robust filtering scheme," in Proc. IEEE ICASSP' 85, pp. 668-971, 1985.

[89] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail preserving filters for image processing," IEEE Trans. on *PAMI*, vol. 9, no. 1, Jan, 1987.

[90] L. S. Davis and A. Rosenfeld, "Noise cleaning by iterative local averaging," IEEE *Trans.* on SMC, vol. 8, no. 9, pp. 705-710, 1978.

[91] H. A. David, Order statistics, Wiley, New York, 1981

[92] J. W. Tukey, Exploratory Data Analysis, Addison-Wesley, Reading, MA, 1970.

[93] G. R. Arce, N. C. Gallagher, and T. A. Nodes, "Median filters: theory for one- and two-dimensional filters," in Advances in Computer Vision and Image Processing, T.S. Huang Ed., JAI Press, 1986.

[94] B. I. Justusson, "Median Filtering: statistical properties," in Two-Dimensional Digital Signal Processing *II*, T.S. Huang Ed., Springer Verlag, 1981.

[95] S. G. Tyan, "Median filtering: deterministic properties," in Two-Dimensional Digital *Signal* Processing *II*, T.S. Huang Ed., Springer Verlag, 1981.

[96] A. C. Bovik, T. S. Huang, and D. C. Munson, "A generalization of median filtering using linear combination of order-statistics," IEEE Trans. on ASSP, vol. 31, no. 6, pp. 1342-1350, Dec. 1983.

[97] R. Bernstein, "Adaptive nonlinear filters for simultaneous removal of different kinds of noise in images," IEEE Trans. on CAS, vol. 34, no. 11, Nov. 1987.

[98] G. R. Arce and R. E. Foster, "Detail-preserving ranked-order based filters for image processing," IEEE Tans. on ASSP, vol. 37, no. 1, Jan. 1989.

[99] H. G. Longbotham and A. C. Bovik, "Theory of order-statistics filters and their relationship to linear fir filters," IEEE Trans. on ASSP, vol. 37, no. 2, Feb. 1989.

[100]E. J. Coyle, J. H. Lin, and M. Gabbouj, "Optimal stack filtering and the estimation and structural approaches to image processing," IEEE Trans. on ASSP, vol. 37, no. 12, pp. 2037-2066, Dec. 1989.

[101]P. Maragos and R. W. Schafer, "Morphological filters part II; their relationship to median, order-statistic, and stack filters," IEEE Trans. on ASSP, vol. 35, no. 8, pp. 1170-1184, 1987.

[102]Y. H. Lee and A. T. Fam, "An edge gradient enhancing adaptive order statistic filter," *IEEE* Trans. on ASSP, vol. 35, no. 5, May 1987.

[103]R. C. Hardie and C. Boncelet, "LUM filters: a class of rank-order-based filters for smoothing and sharpening," IEEE Trans. on Signal Processing, vol. 41, no. 3, pp. 1061-1076, March 1993.

[104]H. Longbotham and D. Eberly, "The WMMR filters: a class of robust edge enhancers," IEEE Trans. on SP, vol. 41, no. 4, April 1993.

[105]D. L. Donoho and P.J. Huber, "The notion of breakdown point," in A Festschriftfor E. L. Lehmann, Wadsworth, 1983.

[106]R. J. Clarke, Transform coding of images, Academic Press, London, 1985.V. Bhaskaran and K. Konstantinides, Image and video compression standards. Kluwer, 1995.

[107]H. C. Reeves and J. S. Lim, "Reconstruction of blocking effects in image coding", Optical Engineering, vol. 23, pp. 34-37, 1984.

[108]A. Zakhor, "Iterative Procedures for reduction of blocking effects in transform image coding," IEEE Trans. on *C&S* for Video Technology, vol. 2, pp. 91-95, March 1992.

[109]Y. Yang, N. P. Galatsanos, A. K. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," IEEE *Trans.* on Image Processing, vol. 4, no. 7, pp. 896-908, July 1995.

[110]I. Linares, R. Mersereau, and M. Smith, "JPEG estimated spectrum adaptive postfiltering using image-adaptive Q-tables and canny edgw detectors," in proceedings of *ISCAS'96*, Atlanta, GA, May 1996.

[111]R.A. Gopinath, M. Lang, H. guo, and J. E. Odegard, "Wavelet-based post-processing of low bit rate transform coded images," in proceedings of IEEE *ICIP'94*, pp. 913-917, Austin, Texas, Nov. 1994.

[112]D. L. Donoho, "Denoising by soft-thresholding," *IEEE Trans, on Information Theory*, vol. 41, no. 3, pp. 613-627, May 1995.

[113]W. Stallings, *ISDN and broadband ISDN with frame relay and ATM, 3rd Edition.* Prentice-Hall, 1995.

[114]P.H. Lee, S.H. Lee, and R. Ansari, "Cell loss detection and recovery," presented at the Third International Workshop on Packet Video, Morristown, New Jersey, March 1990.

[115]H. Hessenmuller, "Video signal transmission in an ATM-based broadband network: treatment of cell losses," presented at the Third International Workshop on Packet Video, Morristown, New Jersey, March 1990.

[116]A.S. Tom, C.L. Yeh, and F. Chu, *"Packet video for cell loss protection using deinterleaving and scrambling,"* in proceedings of *IEEE ICASSP'91*, Toronto, Canada, May 1991.

[117]T. Houdoin and J. Cochennec, "The transfer of variable bit rate in ATM networks: the need for a comprehensive approach," presented at the Third International Workshop on Packet Video, Morristown, New Jersey, March 1990.

[118]M. Ghanbari and V. Seferidis, "cell-loss concealment in ATM video codecs," *IEEE Trans. on CSVT*, vol. 3, no. 3, pp. 238-247, June 1993.

[119]P. Pancha and M.E. Zarki, "MPEG coding for variable bit rate video transmission," *IEEE Communications Magazine,* vol. 32, no. 5, pp. 54-66, May 1994.

[120]Y. Wang and Q.-F. Zhu, "signal loss recovery in DCT-based image and video codecs," in proceedings of *SPIE Conf. on Visual Communication and Image Processing,* vol. 1605, pp. 667-678, Nov. 1991.

[121]H. Sun and J. Zdepski, "Adaptive error concealment algorithm for MPEG compressed video," in proceedings of *SPIE Conf. on Communications and Image Processing*," vol. 1818, 1992.

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

[123]S.S. Hemami and T.H.Y. Meng, "Spatial and temporal video reconstruction for non-layered transmission," in proceedings of *Fifth International Workshop on Packet Video, VISICOM'93.* 1993.

[124]W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *IEEE Trans. on Consumer Electronics,* vol. 39, no. 3, pp. 455-460, Aug. 1993.

[125]M. Kharatichvili and P. Kauff, "Concealment techniques for data-reduced HDTV recording," *Signal Processing and Image Communication,* no. 7, pp. 173-182, 1995.

[126]D. Minoli, *Video dialtone technology, digital video over ADSL, HGC, FTTC, & ATM.* McGraw-Hill Inc., 1995.

[127]W. Verbiest, L. Pinno, and B. Voeten, "The impact of the ATM concept on video coding," *IEEE Journal of Selected Areas in Communications*, vol. 6, no. 9, pp. 1623-1632, Dec. 1988.

[128]P. Salama, N. Shroff, E.J. Coyle, and E.J. Delp, "Error concealment techniques for encoded video streams," in proceedings of *IEEE ICIP'95*, Washington DC, Oct. 23-26, 1995.

[129]Q. Zhu, Y. Wang, and L. Shaw, "Coding and cell-loss recovery in DCT-based packet video," *IEEE Trans. on CSVT*, vol. 3, no. 3, pp. 248-258, June 1993.land, Sept. 16-19, 1996.

[130]L.T. Chia, D.J. Parish, and J.W.R. Griffiths, "On the treatment of video cell loss in the transmission of motion-JPEG and JPEG images," *Computer & Graphics*, vol. 18, no. 1, pp. 11-19, 1994.

[131]S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721-741, Nov. 1984.

[132]P. Salama, N. Shroff, and E.J. Delp, "A bayesian approach to error concealment in encoded video streams," in proceedings of *IEEE ICIP'96*, EPFL Switzerland, Sept. 16-19, 1996.

[133]T. Symchony, R. Chellappa and Z. Lichtenstein, "Relaxation algorithm for MAP estimation of gray-level images with multiplicative noise," *IEEE Trans. on Information Theory*, vol. 36, pp. 608-613, May 1990.

[134]J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society*, series B, vol. 48, no. 3, pp. 259-302, 1986.

[135]M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I - theory," *IEEE Trans. on Signal Processing*, vol. 41, no. 2, pp. 821-833, Feb. 1993.

[136]M. Unser, "Efficient dyadic wavelet transformation of images using interpolation filters," in proceedings of *IEEE ICASSP'93*, pp. V149-V152, Minnesota, 1993.

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

[138]W. Zeng and B. Liu, "Geometric-structure-based directional filtering for error concealment in image/video transmission," in proceedings *SPIE Wireless Data Transmission, Photonics East'95*, vol. 2601, Philadelphia, Oct. 1995.

[139]V. Algazi, G. Ford, and R. Potharlanka, "Directional interpolation of images based on visual properties and rank order filtering," in proceedings of *IEEE ICASSP'91*, pp. 3005-3008, 1991.

[140]M Wada, "Selective recovery of video packet loss using error concealment," *IEEE*

Journal of Selected Areas in Communication, vol. 7, no. 5, pp. 807-814, June 1989.

[141]P. Haskell and D. Messerschmitt, "Resynchronization of motion compensated video affected by ATM cell loss," in proceedings of IEEE *ICASSP'92*, pp. III-545-III548, 1992.

APPENDIX: PROOF OF THEOREMS

Proof of Theorem 3.3

Assuming the atom φ_{γ_p} is selected by the OMP algorithm at iteration p < N, then there exists two possibilities:

1. $\langle r^{(p)}\boldsymbol{x}, \varphi_{\gamma_p}\rangle = 0$, for $p <$ N.
2. $\langle r^{(p)}\boldsymbol{x}, \varphi_{\gamma_p}\rangle \neq 0$, for p < N.

If statement 1 is true, then the algorithm has converged in less than N iteration. If statement 2 is true, it has to be shown that the set of orthogonalized atoms $\{\vartheta_k\}$, where $0 \leq \mathrm{k} < \mathrm{N}$, form an orthogonal basis for $\mathcal{H}$.

Clearly, $r^{(p)}\boldsymbol{x}$ is orthogonal to $\{\varphi_{\gamma_k}\}$ for $0 \leq p < \mathrm{k}$. Since $\langle r^{(p)}\boldsymbol{x}, \varphi_{\gamma_p}\rangle \neq 0$, by assumption the atoms in the set $\{\varphi_{\gamma_k}\}$ should be linearly independent. In finite dimensional spaces this implies that the set $\{\varphi_{\gamma_k}\}$ for $0 \leq \mathrm{k} < \mathrm{N}$ form a basis for $\mathcal{H}$. Therefore, the orthogonalized set of atoms $\{\vartheta_k\}$ obtained from this set, form an orthogonal basis for $\mathcal{H}$. This means that the OMP converges in less than or equal to N iterations.

□

• Proof of Theorem 6.1

Let $\{z_i : i=1,\ldots,n\}$ be the set of input data points within the processing window W, and let $W^{NC} = \{z_{NCk}$: k = 1,...,$L)$ be any non-contiguous subset of W with size L, $L < n$. Let the members of W^{NC} be ordered, such that

$$z_{NC(1)} < z_{NC(3)} < z_{NC(4)} < \ldots < z_{NC(L+1)} \tag{A.1}$$

Note the absence of $z_{NC(2)}$ in the above list. Since W^{NC} is not contiguous (*Definition* 6.3), there must exist a member, say $z_{NC(2)}$, such that $z_{NC(2)} \in W$, $z_{NC(2)} \notin W^{NC}$, and $z_{NC(1)} < z_{NC(2)} < z_{NC(L+1)}$. It has been assumed, without loss of generality, that

$z_{NC(1)} < z_{NC(2)} < z_{NC(3)}$. Two subsets W^1 and W^{L+1}, which are ***less contiguous*** than W^{NC}, can be constructed by adding the member $z_{NC(2)}$, and deleting one of the two extreme values, $z_{NC(1)}$ or $z_{NC(L+1)}$. By ***less contiguous*** it is meant that if there exist m members in W^{NC} that are not in W, but have values within the extremes $z_{NC(1)}$ and $z_{NC(L+1)}$, then W^1 or W^{L+1} have only (m-1) such numbers.

These sets can be described as $W^{L+1}=\{z_{NC(i)}:i=1,...,L\}$, $W^1=\{z_{NC(i)}:i=2,...,L+1\}$, and $W^2=\{z_{NC(i)}:i=1,3,4,...,L+1\}$. The superscript ***i*** in the subset W^i means $z_{NC(i)}$ is not present in that subset. Now, the objective function in ***(6.6)***can be written as

$$\mathbf{J}_i = \mathbf{J}_1(W^i) = \left(\sum_{k=1}^{L+1} z^2_{NC(k)} - z^2_{NC(i)}\right) - \frac{1}{L}\left(\sum_{k=1}^{L+1} z_{NC(k)} - z_{NC(i)}\right)^2$$

Now, it will be proven that either $\mathbf{J}_{L+1} < \mathbf{J}_2$ or $\mathbf{J}_1 < \mathbf{J}_2$. Let

$$\bar{z} = \sum_{k=1}^{L+1} z_{NC(k)}$$

then

$$\begin{aligned}
\mathbf{J}_2 - \mathbf{J}_{L+1} &= z^2_{NC(L+1)} - z^2_{NC(2)} + \frac{1}{L}[(\bar{z} - z_{NC(L+1)})^2 - (\bar{z} - z_{NC(2)})^2] \\
&= z^2_{NC(L+1)} - z^2_{NC(2)} + \frac{1}{L}[(z_{NC(2)} - z_{NC(L+1)})(2\bar{z} - z_{NC(2)} - z_{NC(L+1)})] \\
&= (z_{NC(L+1)} - z_{NC(2)})\left[(z_{NC(L+1)} + z_{NC(2)})\left(1 + \frac{1}{L}\right) - \frac{2\bar{z}}{L}\right]
\end{aligned}$$

$$\mathbf{J}_2 - \mathbf{J}_{L+1} = 2(z_{NC(L+1)} - z_{NC(2)})\left[\frac{(z_{NC(L+1)} + z_{NC(2)})}{2} - \frac{\bar{z}}{L\left(1 + \frac{1}{L}\right)}\right] \tag{A.2}$$

there are only two possibilities

- ***Case*** 1

$$\frac{(z_{NC(L+1)} + z_{NC(2)})}{2} \geq \frac{\bar{z}}{L\left(1 + \frac{1}{L}\right)} \tag{A.3}$$

- *Case* 2:

$$\frac{(z_{NC(L+1)} + z_{NC(2)})}{2} \leq \frac{\bar{z}}{L\left(1 + \frac{1}{L}\right)} \tag{A.4}$$

If **(A.3)** is true, then $(\mathbf{J}_2 - \mathbf{J}_{L+1})\ 2\, 0$, since $(z_{NC(L+1)} - z_{NC(2)}) > 0$. If (A.4) is true, then similar to (A.2), the following expression can be derived

$$\mathbf{J}_2 - \mathbf{J}_1 = 2(z_{NC(1)} - z_{NC(2)})\left[\frac{(z_{NC(1)} + z_{NC(2)})}{2} - \frac{\bar{z}}{L\left(1 + \frac{1}{L}\right)}\right]$$

By definition, $(z_{NC(2)} - z_{NC(1)}) \geq 0$, and by (A.4)

$$\frac{\bar{z}}{L\left(1 + \frac{1}{L}\right)} - \frac{(z_{NC(1)} + z_{NC(2)})}{2} \geq \frac{(z_{NC(L+1)} + z_{NC(2)})}{2} - \frac{(z_{NC(1)} + z_{NC(2)})}{2}$$

hence, It can be conclude that $(\mathbf{J}_2 - \mathbf{J}_1)\ 2\, 0$.

□