# Distributed Image and Video Coding

# based on
# Compressed Sensing

A thesis presented in partial fulfilment of the
requirements for the degree of

Doctor of Philosophy
in
Computer Engineering
at
Massey University
New Zealand.

**Muhammad Yousuf Baig**

2013

# Abstract

Conventional methods for encoding of images and videos is a complex process with high computational demands. They are designed for application scenarios where the signals concerned are encoded once and played back many times. However, new applications such as wireless video sensor networks demand low cost and low power cameras with limited computing resources. The focus of this thesis is on such image and video coding systems where the computational burden is shifted from the encoder to the decoder.

Three separate coding schemes have been developed – two for videos and one for images. Together they form a framework for distributed coding which is based on the theory of compressed sensing and distributed coding. Compressed sensing is a relatively new theory for the acquisition of sparse signals that allows the sampling rate to be much lower than the Nyquist limit. Distributed coding is based on the theorem by Slepian and Wolf, and Wyner and Ziv. It allows different correlated parts of a signal to be encoded independently without loss of coding efficiency. The decoding of these separately encoded parts are then decoded jointly in order to exploit the correlation between them. The main characteristics of the coding scheme proposed in this thesis are: (1) they do not require the use of traditional codecs; (2) only compressed sensing measurements are used for encoding and decoding; (3) no motion estimation and compensation are involved for videos.

The first proposed coding scheme is for the encoding of whole video frames. The compressed sensing measurement of individual frames are separately encoded. These frames are divided into key and non-key frames with the key frames encoded at a higher rate than non-key ones. While the key frames are decoded independently, the non-key ones are decoded with the help of side information generated from the measurements of the key frames. The most important part of the decoder is a simple, yet effective, side information generation method which requires only minimal computation. The side information generated is simply added to the measurements of the non-key frames for use with any compressed sensing reconstruction algorithm. The other two coding schemes are block-based coding methods. Each image or frame is divided into non-overlapping image blocks in a similar way it is done in some existing coding standards. The coding of the blocks are performed in a distributed manner by classifying them into key blocks and non-key blocks. An adaptive encoding strategy based on block similarity is also developed. Experimental analyses using publicly available test images and videos show that the performances of the simpler codecs proposed are better than other existing compressed sensing based codecs. The video codecs also out-perform conventional distributed video codec in terms of simplicity, compression ratio and decoding complexity.

The basis of these coding methods is on the correlation of frames or blocks. This correlation is established through experimental analyses. These analyses also showed that the minimum square error between any pair of them can be effectively used as a measure of correlation. In conjunction with the development of the codecs, a quantization scheme that is tailored to the statistics of CS measurements has also been proposed. This scheme yields better results than a uniform quantizer and those used for JPEG. The quantizer is also robust against different statistics of individual images. Separate experimental evaluations also show that structurally random matrices are the best sensing matrices for acquiring images and the sparse reconstruction by separable approximation (SpaRSA) algorithm produces the best reconstructed image quality.

# Acknowledgments

I am profoundly grateful to my supervisor Associate Professor Edmund Lai for his constant support and guidance throughout my PhD study. There were many occasion in this journey when things were not going in a right direction but Edmund's encouragements and endless patience has helped me to finish this thesis. Edmund's knowledge and inspiring research insights were always fruitful and without him, this endeavour could have never been completed.

I am also thankful to my co supervisor Dr Amal Punchihewa for his valuable comments, suggestions and feedback throughout my research. This PhD thesis could not be completed without a financial support and scholarship from Higher Education Commission, Pakistan. I am thankful for their support and appreciate their effort for educational reforms. I am also thankful to Mehran University for granting me study leave to pursue PhD study.

I want to express my gratitude to my parents for their prayers and support. I am thankful to my wife Sana for her love, understanding and endless support despite all the difficulties. Finally love to my children, Maryam and Taha who always brings smiles back in a long and hard day

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **AVC** | Advanced Video Coding |
| **BCH Codes** | Bose Chaudhuri Hocquenghem Codes |
| **BCQP** | Bounded Constraint Quadratic Program |
| **BCS** | Bayesian Compressive Sensing |
| **BM** | Block Matching |
| **BP** | Basis Pursuit |
| **BPDN** | Basis Pursuit Denoising |
| **CoSaMP** | Compressive Sampling Matching Pursuit |
| **CRC** | Cyclic Redundancy Check |
| **CS** | Compressed Sensing |
| **DCT** | Discrete Cosine Transform |
| **DCVS** | distributed compressed video coding |
| **DISCOS** | Distributed Compressed Video Sensing |
| **DISCOVER** | DistributedCoding forVideo Services |
| **DSC** | Distributed Source Coding |
| **DVC** | Distributed Video Coding |
| **Ffmpeg** | Fast Foraward MPEG |
| **GOB** | Group of Blocks |
| **GOP** | Group of Picture |
| **GPSR** | Gradient Projection for Sparse Reconstruction |
| **ISAR** | Inverse Synthetic Aperture Radar |
| **ITU-T** | International Telecommunication Union |
| **JPEG** | Joint Picture Expert Group |
| **k-t FOCUSS** | Focal Under Determined System Solver in k-t Space |
| **LASSO** | Least Aboslute Shrinkage and Selecion Operator |
| **LDPC** | Low Density Parity Check |
| **LDPCA** | Low Density Parity Check Accumulate |
| **LIMAT** | Lifting-based Invertible Motion Adaptive Transform |
| **LP** | Linear Programming |
| **MC** | Motion Compensation |
| **MCFI** | Motion Compensated Frame Interpolation |
| **MCI** | Motion Compensated Interpolation |
| **ME** | Motion Estimation |
| **MP** | Matching Pursuit |
| **MPEG** | Motion Picture Expert Group |
| **MR** | Measurement Rate |
| **MRI** | Magnetic Resonance Imaging |
| **MSE** | Mean Square Error |
| **NESTA** | Nesterov's Algorithm |

| | |
|---|---|
| **NP** | Non-deterministic Polynomial-time |
| **OMP** | Orthogonal Matching Pursuit |
| **PCA** | Principle Component Analysis |
| **PFT** | Partial Fourier Transform |
| **PL** | Projected Landweber |
| **PRISM** | Power-efficient, Robust, hIgn compression Syndrome based Multimedia coding |
| **PSNR** | Peak Signal to Noise Ratio |
| **QCIF** | Quad Common Interchange Format |
| **RD** | Rate Distortion |
| **RDO** | Rate Distortion Optimization |
| **RIP** | Restricted Isometry Property |
| **SBHE** | Scrambled Block Hadamard Ensemble |
| **SFE** | Scrambled Fourier Ensemble |
| **SI** | Side Information |
| **SpaRSA** | Sparse Reconstruction by Seperable Approximation |
| **SPL** | Smoothed Projected Landweber |
| **SRM** | Structurally Random Matrices |
| **SSIM** | Structure Similarity Index |
| **STD** | Standard Deviation |
| **StOMP** | Stagewise Orthogonal Matching Pursuit |
| **SVC** | Scalable Video Coding |
| **SVD** | Singular Value Decomposition |
| **TV** | Total Variation |
| **TwIST** | Two-Step Iterative Shrinkage Thresholding |
| **TWR** | Through-the-wall Radar |
| **WT** | Wavelet Transform |
| **WZ** | Wyner Ziv |

# Chapter 1

# Introduction

Image and video compression and their related applications have advanced substantially in recent years. Major coding standards such as Joint Photographic Experts Group (JPEG) [1], Moving Picture Experts Group (MPEG) [2] and H.26x [3] are well developed and widely deployed. The techniques employed in these standards exploit the spatial and/or temporal redundancies in the signal for data compression very effectively. However, this causes the encoding process to be typically 5 to 10 times computationally more complex than the decoding process [4]. This kind of computationally asymmetrical systems is suited for applications where the video is encoded once and decoded many times based on a single camera (sensor). The research described in this thesis is concerned with a new approach to image and video coding where the computational burden is shifted from the encoder to the decoder. This approach is more suitable for modern video applications where a network of cameras is deployed.

## 1.1 Background and Motivation

Conventional video coding systems are developed for applications which typically require the compressed video sequences to be played back many times. Since compression only needs to be performed once while decompression (playback) is performed many times, it is desirable that the decoding/decompression process can be done as simply and quickly as possible. Therefore, essentially all current image and video compression schemes, such as the various JPEG and MPEG standards as well as H.264 [2, 3], involve a complex encoder and a simple decoder. In order that video encoding can be performed in real time at frame rates of 30 frames per second or more, the encoding process has to be performed by specially designed hardware, increasing the cost of cameras.

In the past ten years, there has been much research and development in sensor networks where a large number of sensors are deployed [5–8]. For some applications such as video surveillance and sports broadcasting, these sensors are in fact video cameras. For such systems, there is a need to re-evaluate conventional strategies for video coding. If the encoders are made simpler, then the cost of the cameras can be reduced. If tens or hundreds of cameras are involved, the total system cost of this proposed setup could be substantially lower compared with deploying current camera systems. Apart from economics, the cameras used in (wireless) video sensor networks are required to have low power consumption which in turn limit the amount of computing resources. These requirements make it difficult for traditional coding methods to provide an acceptable solution due to its inherit encoding complexity. Thus there is a need for new image and video coding techniques that shifts the complexity from the encoder to the decoder.

In the past few years, two related new areas of signal acquisition and compression have arisen. The first one is Distributed Video Coding (DVC) [9]. The distributed approach allows any redundancy or correlation between video frames to be exploited at the decoder rather than the encoder, thus reducing the complexity of the encoder. At the same time, a new theory called Compressed Sensing (CS) has been developed [10–12]. It provides a completely new approach to data acquisition. In essence, CS tells us that for signals which possess some "sparsity" properties, the sampling rate required to reconstruct these signals with good quality can be much lower than the lower bound specified by Shannon's sampling theorem. Since image and video signals contain substantial amounts of redundancy, they are sparse signals and CS can potentially be applied. There have been some studies reported recently on the application of CS to video signals in the literature [13–18]. Although they are mostly preliminary exploratory research, they point to the possibility of using CS as a simple encoder for video signals. The simplicity of the encoding process is traded off by a more complex, iterative decoding process. The reconstruction process of CS is usually formulated as an optimization problem which potentially allows one to tailor the objective function and constraints to the specific application. The concept of CS fits in well with DVC. The use of CS combined with DVC can provide the data rate reduction required while providing flexibility at the decoder.

## 1.2 Scope and Objectives

The aim of this thesis is to explore ways by which DVC and CS can be used together in new image and video coding architectures with the characteristics of simple low-power encoders coupled and flexible decoders that can be tailored to a specific application. The encoders must rely only on CS data acquisition techniques; no traditional codec should be required. For video coding, the simplicity required also dictates that the computationally expensive process of motion estimation or compensation should not be used. At the same time, there is no assumption that a feedback channel exist between the decoder and the encoder. This is because in many applications, such a feedback channel is not physically possible.

Since CS-based cameras are still in early stages of research and development, the images and videos that are used for experimentation are acquired in the traditional way. The CS measurements of these data are computed from the original pixel data. However, none of the compression systems proposed in this thesis shall assume that the pixel data are available to the encoders. This make them truly CS-based systems.

Specific objectives of the research are as follows:

- Evaluate the effectiveness of different CS sensing matrices and reconstruction algorithms for image compression.

- Study the effects of quantization on the reconstruction of image and video signals acquired using CS and hence develop effective quantization schemes. Current quantization schemes are primarily based on characteristics of discrete cosine transform and wavelet transforms which may no longer be applicable.

- Study the relationship between CS measurements of adjacent frames in a videos sequence. Investigate how this relationship can be used to improve the reconstruction quality and compression ratio for video coding.

- Study the relationship between CS measurements of neighbouring blocks in an image. Investigate how this relationship can be used to improve the reconstruction quality and compression ratio for Intra image and Inter-frame compression.

- Develop an effective CS-based image coder decoder (codec) using a distributed coding approach.

- Develop an effective CS-based video coder decoder (codec) using a distributed coding approach.

## 1.3 Original Contributions

The main contributions of this thesis are summarized as follows:

- Different classes of CS sensing matrices and reconstruction algorithms have been characterized, in terms of acquisition time, reconstruction time and reconstruction quality, specifically for image compression.

- A new quantization scheme that is suitable for quantizing CS measurements is proposed. This scheme is based on the statistical properties of CS measurements and has been shown to be robust against individual image statistics. It has also been shown that this scheme performs better than both JPEG and uniform quantization.

- A distributed inter-frame compressed video sensing codec is proposed. Unlike most architectures found in the literature, it does not require a feedback channel or motion estimation. The proposed codec only uses CS measurements to generate side information needed for decoding non-key frames. The new side information generation scheme is based on the correlation of CS measurements between video frames. It is a simple and effective scheme that requires minimal computation at the decoder. It also does not require key frames to be decoded first. Experimental results show that this codec performs well against other CS-based codecs and comparable with other conventional distributed codecs which have much higher complexity.

- For block-based processing, the impact of block size on CS reconstruction in image coding is studied. It has been found that a larger block size provides better results at lower measurement rates but in general as measurement rate increases the difference is not significant.

- A new distributed CS image codec with adaptive and non-adaptive block encoding is proposed. The proposed adaptive block classification based on adjacent block similarity is able to reduce measurement rate. Intra block correlation is used to generate side information to help decode non-key blocks. This side information generation scheme is used to improve the reconstruction quality of blocks in both adaptive and non-adaptive encoding.

- A related distributed block-based video codec with adaptive block encoding is proposed. This codec exploits both intra-frame and inter-frame block correlation to generate side information. It is shown through experimental comparison that its performance is better than many other CS-based video codecs.

## 1.4 Organization of the Thesis

The rest of this thesis is structured as follows.

In Chapter 2, an overview of the essential elements of the theory of compressed sensing is presented. A review of the literature on CS-based Image and Video coding is presented in Chapter 3. The performances of different types of sensing matrices and reconstruction algorithms for CS are evaluated specifically for full image compression in Chapter 4. A new quantization scheme for quantizing CS measurement is also proposed in this chapter. It is used in the codecs proposed in subsequent chapters. In Chapter 5, a distributed Inter-frame video codec based on CS is presented. It consists of a simple yet effective side information generation scheme that exploits the correlation of the CS measurements between frames. A block-based coding approach is adopted in Chapters 6 and 7 for image and video coding respectively. Both adaptive and non-adaptive measurement rate assignment schemes are explored. Their performances are evaluated against other CS-based and conventional distributed techniques reviewed in Chapter 3. Chapter 8 concludes the thesis and suggests some avenues for future work.

# Chapter 2

# An Overview of Compressed Sensing

Compressed Sensing (CS), also known as Compressive Sampling, is a theory that is first developed by Emmanuel Candes, together with Justin Romberg and Terry Tao [10–12,19]. This theory provides a way for sampling continuous signals at a rate much lower than the Nyquist rate given by Shannon's sampling theorem. It is applicable to signals that are *sparse* in some domain. This implies that signals that are highly compressible, such as images and videos, can potentially take advantage of CS. It is well known that such signals posses structures or features which could be exploited for efficient encoding. CS provides techniques and methods which can be used to acquire those significant features efficiently without loosing significant information. The signal processing research community has been attracted by the idea of CS and a large number of papers have been published both in the theory and applications of CS in the last few years [20]. This chapter provides a brief survey of the theory and techniques of CS, together with some potential applications.

## 2.1 Key Elements of Compressed Sensing

Shannon's uniform sampling theorem [21,22] provides a lower bound on the rate by which an analogue signal needs to be sampled in order that the sampled signal fully represents the original. If a signal $f(x)$ contains no frequencies higher than $\omega_{max}$ radians per second, then it can completely determined by samples that are spaced $T = \pi/\omega_{max}$ seconds apart.

$f(x)$ can be reconstructed perfectly using these samples $f(nT)$ by

$$f(x) = \sum_{k \in Z} f(kT) sinc(x/T - k) \tag{2.1}$$

The uniform samples $f(nT)$ of $f(x)$ may be interpreted as coefficients of some basis functions obtained by appropriate shifting and rescaling of the sinc function.

For high bandwidth signals such as video, the amount of data generated based on a sampling rate of at least twice the bandwidth is very high. Fortunately, only a relatively small number of the discrete cosine transform (DCT) or wavelet transform (WT) of these signals have significant magnitudes. Therefore, if those coefficients that have magnitudes smaller than a certain threshold are discarded, then the amount of data that represents the original signal can be reduced. This is the basic idea exploited by all existing lossy signal compression techniques.

Another way of putting it is that these signals are *sparse* in the DCT and WT domains. Sparsity implies that the information rate of the signal is much lower than is suggested by its bandwidth. For such signals, instead of acquiring the raw data, performing an orthogonal transform and then discarding insignificant coefficients, CS theory suggests that essential information content can be acquired directly, through a measurement process that is incoherent with the signal. Subsequently, a high quality signal can be reconstructed from these measurements using appropriate optimization techniques. These key elements of CS are illustrated in Figure 2.1.

### 2.1.1 Sparsity

Sparsity is important in CS as it determines how efficient one can acquire signals non-adaptively. The most common definition of sparsity in the CS literature is as follows. Let $f \in R^n$ be a vector which represents a signal that can be expanded in an orthonormal basis $\Psi = [\psi_1, \psi_2 \ldots \psi_n]$ as

$$f(t) = \sum_{i=1}^{n} x_i \psi_i \tag{2.2}$$

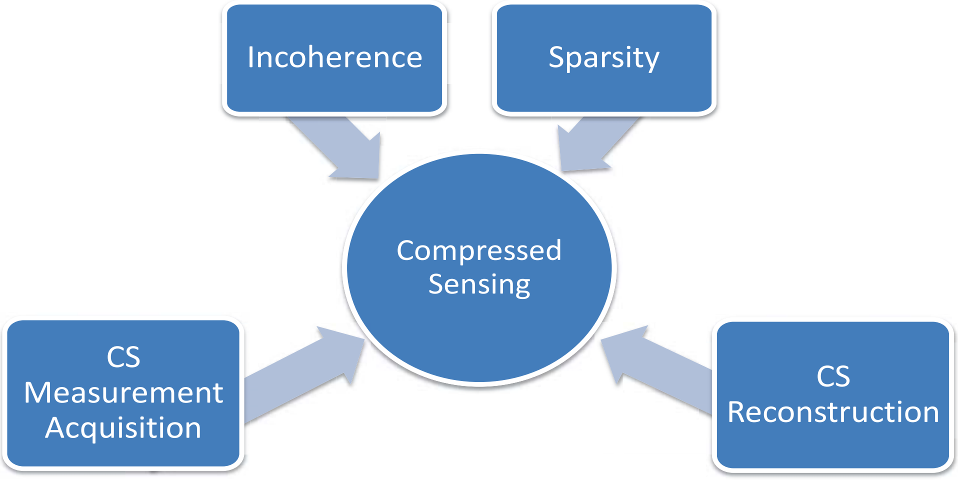


Figure 2.1: Key Elements of Compressed Sensing

where $x_i = \langle f, \psi_i \rangle$ are the coefficients of this expansion. In matrix form, (2.2) becomes

$$\mathbf{f} = \Psi \mathbf{x} \tag{2.3}$$

If all but a few of the coefficients $x_i$ are zero, then $f$ is said to be sparse in a strict sense. In this case, let $S$ denotes the number of non-zero coefficients with $S \ll n$. Then $f$ is said to be $S$-sparse. If $x$ is $S$-sparse and can be approximated by $x_S$, then the error $||f - f_S||_{l_2}$ is small since $||f - f_S||_{l_2} = ||x - x_S||_{l_2}$ for an orthonormal basis $\Psi$.

In practice, most compressible signals have only a few significant coefficients while the rest have relatively small magnitudes. If we set these small coefficients to zero in the way that it is done in lossy compression, then we have a sparse signal.

### 2.1.2 Incoherence

Consider two different orthonormal bases, $\Phi$ and $\Psi$, both of $R^n$. The coherence between these two bases is defined in [19] by

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq k,j \leq n} |\langle \phi_k, \psi_j \rangle| \tag{2.4}$$

This gives the largest correlation between any two elements of the two bases. It can be shown that

$$\mu(\Phi, \Psi) \in \left[1, \sqrt{n}\right] \tag{2.5}$$

In CS, a signal that is sparse in $\Psi$ has to be acquired by a sensing (or sampling) domain $\Phi$ which is incoherent with $\Psi$. A signal is more compressible if it has high sparsity in $\Psi$ that is less coherent to $\Phi$. Interestingly, random matrices are largely incoherent with any fixed basis [19].

### 2.1.3 CS Measurement Acquisition

In CS, the ability of a sensing technique to effectively retain the features of the underlying signal is very important. The sensing strategy should provide sufficient number of CS measurements in a non-adaptive manner that are suitable for near perfect reconstruction.

Let $\mathbf{f} = \{f[1], \ldots, f[N]\}$ be a vector of $N$ real-valued samples of a signal. If the representation of $\mathbf{f}$ in a transform domain by $\mathbf{x}$, then

$$\mathbf{f} = \Psi\mathbf{x} = \sum_{i=1}^{N} x_i \psi_i \tag{2.6}$$

where $\Psi = [\psi_1, \psi_2 \ldots \psi_N]$ is the transform basis matrix and $\mathbf{x} = [x_1, \ldots, x_N]$ is an $N$-vector of coefficients with $x_i =< \mathbf{f}, \psi_i >$.

Suppose a general linear measurement process computes $M < N$ inner products $y_j = \langle \mathbf{f}, \phi_j \rangle$ between $\mathbf{f}$ and a collection of vectors $\{\phi_j\}_1^M$. Let $\Phi$ denote the $M \times N$ matrix with the measurement vectors $\phi_j$ as rows. Then the measurements $\mathbf{y} = [y_1, \ldots, y_M]$ is given by

$$\mathbf{y} = \Phi\mathbf{f} = \Phi\Psi\mathbf{x} = \Theta\mathbf{x} \tag{2.7}$$

by using (2.6) with $\Theta = \Phi\Psi$. These measurements are non-adaptive and does not depend on the structure of the signal if $\Phi$ is fixed [12]. The minimum number of measurements needed to reconstruct the original signal depends on the matrices $\Phi$ and $\Psi$. The following theorem specifies this relationship [23].

**Theorem 1.** *Let $f \in R^N$ has a discrete coefficient sequence $x$ in the basis $\Psi$. Let $x$ be $S$-sparse. Select $M$ measurements in the $\Phi$ domain uniformly at random. Then if*

$$M \geq C \cdot \mu^2 (\Phi, \Psi) \cdot S \log N \tag{2.8}$$

*for some positive constant $C$, then with high probability, $x$ can be reconstructed using the following convex optimization program:*

$$\min_{\tilde{x}} \|\tilde{x}\|_{l_1} \text{ subject to } y_k = \langle \phi_k, \Psi\tilde{x} \rangle, \forall k \in J \tag{2.9}$$

*where $J$ denotes the index set of the $M$ randomly chosen measurements.*

This is an important result and provides the requirement for successful reconstruction. It has the following three implications [19]:

(i) The smaller the coherence between the sensing and basis matrices, the fewer the number of measurements needed.

(ii) There will be no information loss by measuring any set of $M$ coefficients.

(iii) The signal $f$ can be exactly recovered without assuming any knowledge of the non-zero coordinates of $x$ or their amplitudes.

### Restricted Isometry Property

There are certain properties a sensing matrix should possess. The most important one is the Restricted Isometry Property (RIP) [24]. For each integer, $S = 1, 2, \cdots$ the isometry constant $\delta_S$ of a matrix $\Phi$ is defined as the smallest number such that

$$(1 - \delta_S)||x||_{l_2}^2 \leq (1 + \delta_S)||x||_{l_2}^2 \tag{2.10}$$

holds true for all $S$-sparse vectors $f$. Matrix $\Phi$ is said to obey RIP of order $S$ if $\delta_S$ is not too close to one. Owing to this property, $\Phi$ preserves the Euclidean length of $S$-sparse signals. This implies that $S$-sparse vectors cannot be in the null space of $\Phi$. Suppose $S$-sparse signals are acquired with $\Phi$ and $\delta_{2S}$ is sufficiently less than one. Then all pairwise distances between $S$-sparse signals are well preserved in the measurement space bound given by [19]

$$(1 - \delta_{2S})||x_1 - x_2||_{l_2}^2 \leq ||\Phi x_1 - \Phi f_2||_{l_2}^2 \leq (1 + \delta_{2S})||f_1 - f_2||_{l_2}^2 \tag{2.11}$$

RIP is a very important property for estimating the suitability of a sensing matrix for effective CS encoding. Unfortunately RIP is NP hard to verify for a specific sensing matrix. There are alternative conditions proposed for estimating the suitability of sensing matrices. The authors in [25] introduced a simple extension of the RIP, called RIP-p. It uses the $lp$-norm instead of the usual $l_2$-norm. Another one is the model-based RIP proposed in [26] which concerns all $x$ restricted to a specific model. If the model is the set of $S - sparse$ signals, then it is reduced to the usual RIP.

Designing an effective sensing matrix $\Phi$ is an important research problem in CS. A good sensing matrix should be capable of maintaining the information of the original signal as well as allowing a small number of measurements for a successful reconstruction. Therefore it should be incoherent to a variety of sparsifying matrices. It should allows for fast and efficient computation and is hardware implementation friendly. There has been many proposals for efficient sensing matrices in the CS literature. In Section 4.1, a classification of these sensing matrices is made and their relative performances are evaluated.

### 2.1.4 CS Reconstruction

The reconstruction problem involves taking the $M$ measurements $\mathbf{y}$ to reconstruct the length-$N$ signal $\mathbf{x}$ that is $S$-sparse, given the random measurement matrix $\Phi$ and the basis matrix $\Psi$. Since $M < N$, this is an ill-conditioned problem.

The conventional approach to solving ill-conditioned problems of this kind is to minimize the $l_2$ norm. In this case, it is expressed as

$$\hat{x} = argmin \left\| x' \right\|_2 \text{ such that } \Theta x' = y \tag{2.12}$$

However, it has been proven that this $l_2$ minimization can only produce a non-sparse $\hat{x}$ [12]. The reason is that the $l_2$ norm measures the energy of the signal and signal sparsity properties could not be incorporated in this measure.

The $l_0$ norm counts the number of non-zero entries and therefore allows us to specify the sparsity requirement. The optimization problem using this norm can be stated as

$$\hat{x} = argmin \left\| x' \right\|_0 \text{ such that } \Theta x' = y \tag{2.13}$$

There is a high probability of obtaining a solution using only $M = S + 1$ independent and identically distributed (i.i.d.) Gaussian measurements [19]. However, the solution produced is numerically unstable [12].

It turns out that optimization based on the $l_1$ norm is able to exactly recover $S$-sparse signals with high probability using only $M \geq cSlog(N/S)$ i.i.d. Gaussian measurements [10, 11]. The convex optimization problem is given by

$$\hat{s} = argmin \left\| x' \right\|_1 \text{ such that } \Theta x' = y \tag{2.14}$$

which can be reduced to a linear program. Algorithms based on Basis Pursuit (BP) [27] can be used to solve this problem with a computational complexity of $O(N^3)$ [10]. BP is a quadratically constrained $l_1$-minimization problem:

$$\min_x \parallel x \parallel_{l_1} \text{ subject to } \parallel y - \theta x \parallel_{l_2} \leq \epsilon \tag{2.15}$$

where $y$ is the noisy CS measurements with noise $\epsilon$. This is the preferred CS reconstruction formulation as the estimate of noise $\epsilon$ may be known or can be computed.

A second approach to solving the CS reconstruction problem is to formulate the BP problem as a second order cone program:

$$\min_x \frac{1}{2}||y-\theta x||_{l_2}^2 + \lambda ||x||_{l_1} \tag{2.16}$$

This formulation is also known as Basis Pursuit Denoising (BPDN) [28]. It is tractable due to its bounded convex optimization nature. The term $\lambda||x||_{l_1}$, which is also known as regularization, can be interpreted as a maximum *a posteriori* estimate in a Bayesian setting. BPDN is very popular in signal and image processing applications.

The third approach is known as the Least Absolute Shrinkage and Selection Operator (Lasso) [29]. It involves the minimization of an $l_2$ norm subject to $l_1$ norm constraints:

$$\min_x ||y-\theta x||_{l_2} \text{ subject to } ||x||_{l_1} \leq \tau \tag{2.17}$$

For block-based image and video processing applications, instead of seeking sparsity in the transform domain, a Total Variation (TV) norm can be used. A total variation of the image can be expressed as

$$||X||_{TV} = \sum_{i,j} \sqrt{(x_{i+1,j}-x_{i,j})^2 + (x_{i,j+1}-x_{i,j})^2} \tag{2.18}$$

The TV norm uses sparse approximation of the image gradient and is very useful in avoiding high frequencies artefacts in the image.

There are a number of different algorithms developed to solve these three CS reconstruction problems. Linear Programming (LP) techniques [10, 11], greedy algorithms [30–33], gradient-based algorithms [34, 35], iterative shrinkage algorithms [36, 37], etc. In Section 4.2, an empirical evaluation of the performances of various reconstruction algorithms for image reconstruction is provided.

## 2.2 Potential Applications

Even though CS theory has only been developed for a few years, it has already been applied to a number of different applications. Some of them are outlined below.

### Magnetic resonance imaging

Magnetic Resonance Imaging (MRI) is one of the most interesting applications of CS. With MRI, a patient goes through a scanning process by lying still inside MRI machine for a substantial amount of time. The quality of the images obtained is determined by how still the patient remains during that time. MRI acquires specific grid points in 2 or 3-dimensional $k$-space and an inverse transform is used to reconstruct the image. The medical image is compressible and therefore, by using CS, a small subset of $k$-space samples is sufficient to reconstruct the image. This leads to a reduction of the acquisition time, indirectly leading to higher quality images obtained. Detailed information on compressed sensing MRI is available in [38, 39] and the references therein.

### Imaging

In typical digital cameras, the image sensors are large and expensive. By using CS, the number of measurements acquired can be made much smaller than the number of pixels needed, resulting in smaller and cheaper cameras. An example is the single pixel camera [40] developed by a group at Rice University in the USA. The idea of this camera is to trade spatial resolution for temporal resolution. The image sensor consists of only a single pixel but it takes a large number of measurements over a short period of time. An array of micro-mirrors is placed in front of a conventional optical lens system to focus on the single-pixel sensor. Each individual micro-mirror can be made to reflect light towards or away from the sensor. The orientation is randomly changed between measurements. In this way, each measurement is a random combination of pixels in the entire scene. The full image is then reconstructed from these measurements based on CS reconstruction techniques.

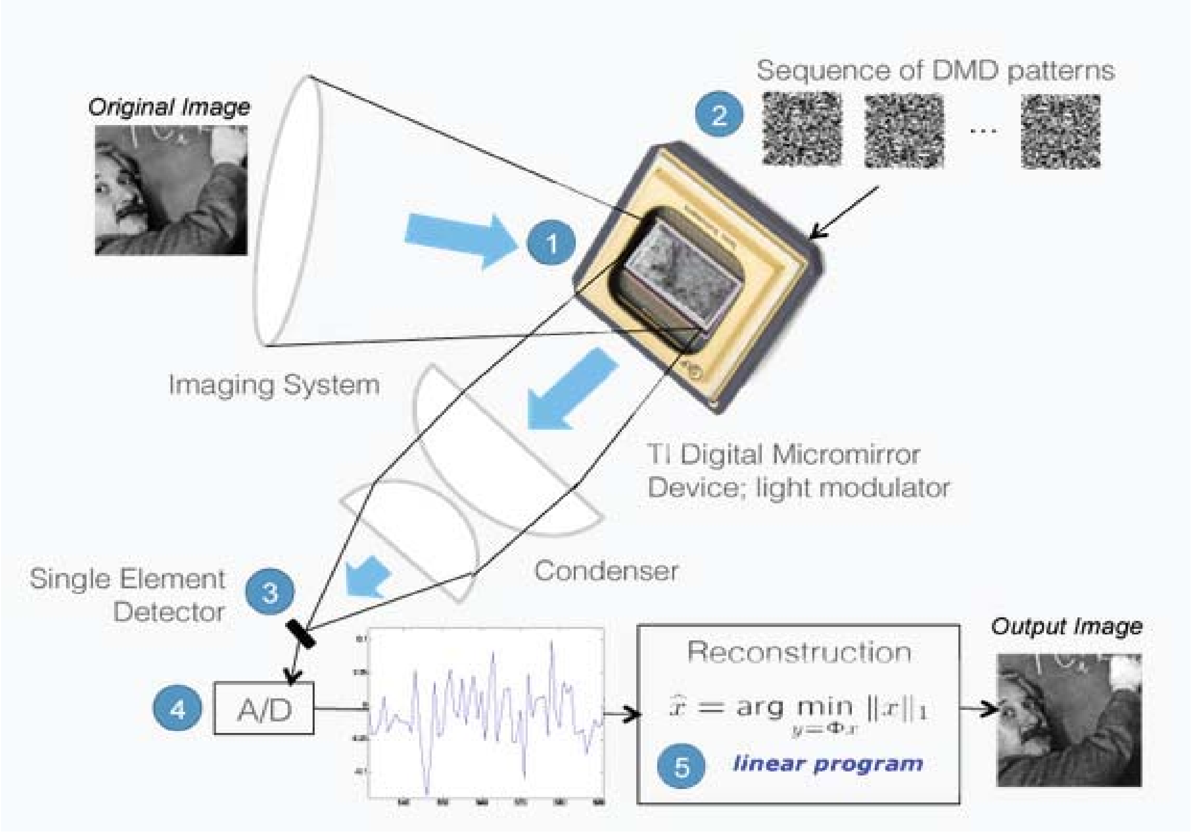


Figure 2.2: CS Camera, [Image Courtesy, InView Corporation [41]]

Recently, InViewCorp has announced a commercial CS camera based on the single-pixel camera concept [41]. Figure 2.2 shows the architecture of the camera. The image is focused onto a Digital Micro-mirror Device (DMD). For each measurement, one-half of the mirrors are directed towards the single photo-detector diode. A reconstruction algorithm is used to reconstruct the full image. Further details on CS hardware developed by InViewCorp is available on their website [41] and in research papers [40, 42].

Another example of a CS based imaging system is the Massachusetts Institute of Technology (MIT) random lens imaging device [43].

### Seismic Data Acquisition

Seismic data acquisition is an expensive and time consuming process. The process starts with an explosion on the surface of the earth. The waves resulting from this explosion will be reflected from the boundaries of layers. An array of geo-phones capture these reflections. This process is repeated many times and every time an explosion is made at a different location. By collecting all these reflections, information of the layers of the earth can then be estimated. By making use of CS, the number of explosions can be reduced, making the process faster and cheaper. Details can be found in [44] and the references therein.

### Hyperspectral Imaging

In hyperspectral imaging, several images acquired at different wavelengths. The data collected consists of a three-dimensional data-cube with two spatial dimensions and one spectral dimension. Each data-cube entry is called a voxel. A pixel's spectral signature is obtained by stacking its voxels in the spectral dimension. These spectral signatures are then used to identify the type of object being imaged. Thus they are typically used in satellite imaging for the identification of objects on the ground. Hyperspectral images are typically compressible (sparse) and therefore CS can potentially be applicable. In [45], a hyperspectral imaging system is proposed that uses a binary aperture mask with a multi-pixel array. In [46], another architecture for hyperspectral imaging that combines multiplexing in the spatial and spectral domain is proposed. InView Corporation [41] also developed a hardware for hyperspectral imaging based on CS theory.

### Radar

Through-the-wall radar (TWR) and inverse synthetic aperture radar (ISAR) are two radar imaging applications that can potentially benefit from CS. Both applications require probing targets using radar signals with large bandwidths. Acquisition and processing a large number of data samples are required to obtain high resolution images. Typically the targets of interest are few (sparse) and have larger cross-section than clutter objects. In [47], the authors shows that when there are only a few targets, using CS based techniques, higher resolution than traditional radar can be obtained. In [48], CS techniques are applied to indoor radar imaging.

### Other applications

The applications of CS are not just limited to the examples mentioned above. CS has found applications in other fields including analog to digital converter design, electron tomography, machine learning, computational biology, and astronomy. More related literature can be found at the Rice University website [49].

## 2.3 Summary

In this chapter, an overview of Compressed Sensing is presented. CS provides a way for signals to be sampled at a much lower rate than suggested by the uniform sampling theorem. This is achieved by exploiting the sparsity of the signal and incoherence between sampling and sparsifying domain. These concepts will be used in the research work presented in this thesis. A number of potential applications of CS, especially related to imaging, have also been described. With the availability of commercial CS hardware, it is expected that more and more practical imaging application will be developed in coming years.

# Chapter 3

# A Review of Compressed Sensing Image and Video Coding

One of the main applications of CS is in imaging as mentioned in Section 2.2. In this chapter, a more in-depth review of current research in using CS for image and video coding, which is the focus of this thesis, is presented. Section 3.1 discusses CS based image coding techniques and Section 3.2 discusses CS based video coding techniques. Some conventional distributed video coding techniques are also reviewed in Section 3.3 for completeness. Some of the experimental results presented in subsequent chapters will be compared with those obtained from these conventional techniques.

## 3.1 Compressed Sensing Image Coding

Image coding techniques based on CS can be broadly classified in two categories – block-based and full image. There are two specific types of block-based techniques that are of interest – Distributed and Multi-scale. An overview of these techniques are presented in this section.

### 3.1.1 Full Image Coding

In full image coding, the CS measurements of the whole image is acquired using a suitable sensing matrix $\Phi$. Due to the large dimension of image data, the size of $\Phi$ is also large.

Most researchers do not apply $\Phi$ directly to the image data. Instead, a sparsity transform is first applied to the image data. CS measurements are then obtained by applying $\Phi$ to the transform coefficients.

The first application of CS to image coding is found in [50]. The authors use a percentage (less than 100%) of CS measurements for image recovery. The images recovered from $3M$ to $5M$ CS measurements were found to have the same quality as the ideal $M$-term wavelet approximation. The main difference is that the CS measurements are obtained randomly without considering the image structure or the magnitude of its transform coefficients. A random Fourier matrix is used for sensing. A similar approach is taken by [33,51] to show the effectiveness of their reconstruction algorithms. The main disadvantage of this approach is that a transform has to be computed, which increases the computational burden on the encoder.

A more efficient method known as Structurally Random Matrices (SRM) is proposed in [52]. First, the image is pre-randomized by random permutation. Then a block-based sensing matrix is applied to the randomized data. Finally, the CS measurements are obtained by randomly down sampling the data. SRM is later extended to Scrambled Block Hadamard Ensemble (SBHE) as a sensing matrix in [53]. This method can be used with different sparse signals quite effectively as all it needs is pre-randomize the signal before applying a CS sensing process.

### 3.1.2 Block Based Coding

Computational burden and memory requirements at the encoder can be reduced by using block based image coding. Block-based coding has been used extensively in traditional image and video coding methods. A block-based image coding method based on CS was first proposed in [54]. The image is divided into non-overlapping blocks and each image block is sampled independently using an appropriate sensing matrix. The image is reconstructed using a variant of the Projected Landweber (PL) reconstruction using hard thresholding and iterative Wiener filtering. This framework was later extended by [55] using directional transforms. At the encoder, an image is divided into non-overlapping blocks and then each block is independently sampled using random Gaussian matrices. To reconstruct the image at the decoder, it start with an initial solution given by

$$x^{(0)} = \Phi^T y \tag{3.1}$$

where $\Phi$ is the sensing matrix and $y$ contains the CS measurements of all the blocks. Then a Wiener filter is applied to $x^{(0)}$. For each block, the smoothed projected Landweber (SPL) transform is applied. A sparsifying transform is then applied to the output of the SPL step. Wiener filtering is incorporated into the basic PL framework to reduce blocking artefacts. A specific stopping criteria based on the $l_2$ norm is checked at the end of each iterative step. Different directional sparsifying transforms are used in the experiments in [55].

Neither of the above block-based schemes consider the fact that different blocks may have different compressibility. Each image block is encoded at the same measurement rate. In general, the sparsity of individual block is different. Smoother image blocks have a higher degree of sparsity, while texture and edges have low level of sparsity.

The approach to block-based CS coding in [56] utilizes the coefficient structure of the transform. At the encoder, an image is divided into blocks and then a weighting matrix is applied to the sensing matrix to provide different emphasis on different coefficients. A $2D$ DCT matrix is used as the sparsifying matrix and the sensing matrix is a random Gaussian matrix. The weighting matrix is derived from the JPEG quantization table by taking the inverse of the table entries and adjusting their amplitudes. This scheme is not very practical as it requires the creation of a weighting matrix which will be different for each block. Although the authors have suggested that the weighting matrix for other blocks can be obtained by sub-sampling or interpolating an initial weighting matrix, the implications are not discussed. Further, a DCT transform is required, which increases the computational burden at the encoder.

In [57], the idea of randomly permuting the transform coefficients before CS measurements acquisition has been put forward. In an image, different blocks contains different amount of information. Smoother blocks have less detailed information and thus naturally have higher sparsity while textures and edges will have more information and thus have low sparsity. To balance the sparsity between image blocks, a random permutation is applied to all transform coefficients. First, a $2D$ DCT is applied to each image block. The transform coefficients of all the block are then randomly permuted. They are then reassembled before block-based CS sampling is performed. At the decoder, inverse random permutation is performed after CS reconstruction of the coefficients. The block image is subsequently recovered by inverse DCT. This approach still requires sparsity transformation. Also, if an image is already transformed at the encoder using a DCT transform, then there is no real benefit in using CS. The encoding delay is high because DCT has to be applied to all blocks first before random permutation can be applied, which negates the benefit of block-based encoding.

### 3.1.3 Multi-scale Coding

Apart from DCT, wavelet transform is the most common transform used in image processing and coding. There is an established statistical model of wavelet coefficients that could be exploited for efficient encoding and decoding. Thus wavelet transform can also be used as a sparsifying transform for CS encoding. The benefit with this approach is that CS sampling can be adapted to the structure of the wavelet decomposition. This technique is generally called multi-scale CS encoding. It was first proposed in [58] where CS encoding was applied to measure the fine scale properties of the signal. Conventional linear measurements and reconstruction were used to obtain the coarse-scale properties of the signal.

A similar approach is adopted in [59] where an image is separated into dense and sparse components. The dense component is encoded by linear measurements while the sparse component is encoded by CS. This encoding scheme is illustrated in Figure 3.1. To obtain dense ($I_D$) and sparse ($I_S$) component, a three level wavelet transform is applied.

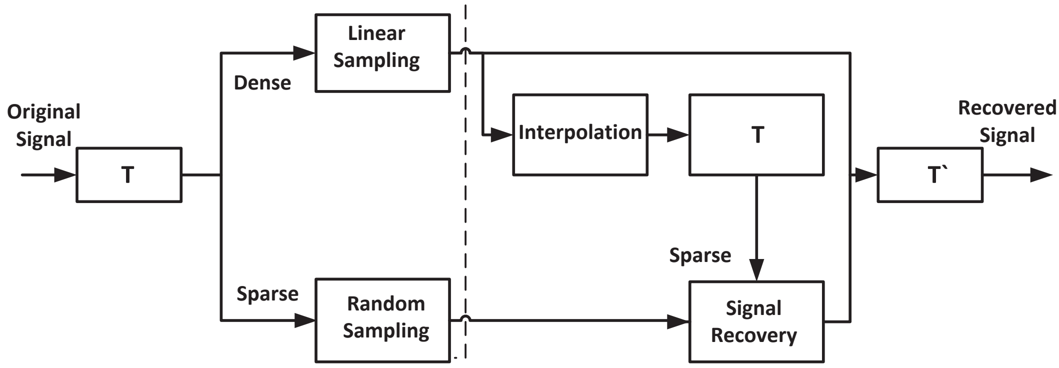


Figure 3.1: Multi Scale Image Coding Scheme [59].

Thus an image $I$ can be expanded as

$$I = \sum_{k} \alpha_{1j_0,k} W_{j_0,k} + \sum_{j=j_1}^{j_2} \sum_{k} \alpha_{2j,k} W_{j,k} \tag{3.2}$$

where $W_{j_0,k}$ and $W_{j,k}$ are wavelets at different scales. The first term on the right-hand-side of (3.2) is the coarsest scale $j_0$ which is considered the dense component $I_D$. The second term is the sparse component $I_S$ where $j_1$ is the next scale and $j_2$ is the finest scale. The CS measurements of $I_S$ is obtained using a Gaussian random matrix $\Phi$. In order to reduced the large memory requirements of a dense random Gaussian matrix, $I_S$ is regrouped into blocks by scales and then sampled on a block-by-block basis. At the decoder, the dense component is simply reconstructed by the inverse wavelet transform. To recover $I_S$, a $2D$ piecewise autoregressive model is used to predict $I$, denoted by $\hat{I}$. This prediction effectively serves as side information. The reconstructed dense component $\hat{I_D}$ is used as the starting point for reconstructing $I_S$ using CS reconstruction.

Another multi-scale CS image coding is presented in [60]. Block-based CS is deployed independently within each subband of each decomposition level of the wavelet transform of the image. If wavelet transform produces $L$ levels of wavelet decomposition, each subband $s$ at level $L$ is divided into $B_L \times B_L$ blocks and sampled using a sensing matrix $\Phi$. The baseband is always sampled fully. As different levels have different sparsity, they are sampled at different rates. To perform reconstruction, an initial solution $x^{(0)} = \Omega^{-1}\hat{x}^{(0)}$ is obtained by applying the inverse block-based wavelet transform. Then Weiner filtering is applied to enhance smoothness. After this, wavelet transform is applied again. For each block of each subband in each decomposition level, the Landweber step is applied using block-based $\Phi$. A stopping criteria is defined to terminate the algorithm.

A multi-scale CS encoding technique that is based on the Bayesian perspective was proposed in [61]. With Bayesian Compressive Sensing (BCS), the CS problem is formulated as a linear regression problem with a constraint (prior) that the underlying signal is sparse. BCS has previously been used for signals and images that are sparse in a wavelet basis [62]. A quadtree structure of the wavelet coefficients for an image [61] is used, with each wavelet coefficient generally serving as a parent for four children coefficients. The wavelet coefficients at the coarsest scale serve as root nodes of the quadtree, with the finest scale of coefficients constituting the leaf nodes. If a wavelet coefficient at a particular scale is negligible, then its children are also generally (but not always) negligible. In CS, wavelet coefficients are not directly observed but projections of these coefficients are observed. In [62], a hidden markov tree is employed within the CS inversion, explicitly imposing the belief that if a given coefficient is negligible, then its children coefficients are likely to be so. Other related Bayesian approaches in a multi-scale framework can be found in [63, 64].

### 3.1.4 Distributed Coding

Distributed coding is a lossy coding technique which uses side information at the decoder to improve the quality of reconstruction. This is based on the seminal work of Slepian and Wolf [65], which is later extended by Wyner and Ziv [66, 67]. Slepian-Wolf coding is a lossless distributed coding technique. Two statistically dependent sources $X$ and $Y$ are independently encoded and jointly decoded. Slepian and Wolf proved that the minimum rate to encode these two correlated sources is the same as that for joint encoding where the encoder has full knowledge of both $X$ and $Y$. With an arbitrarily small

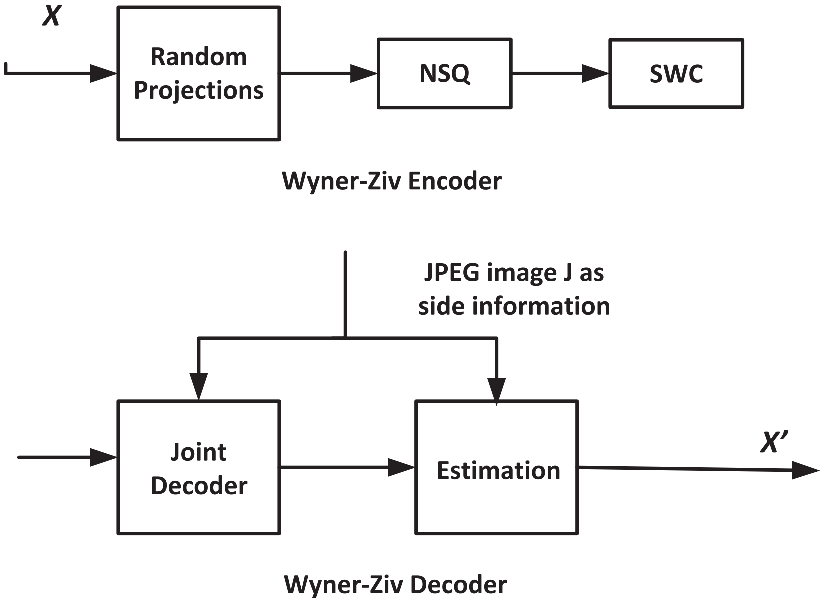


Figure 3.2: Wyner-Ziv Image Coding Scheme [68].

error probability, they can be perfectly reconstructed at the decoder. Wyner and Ziv extended the Slepian-Wolf theorem to lossy compression with side information (SI) at the decoder. Source $X$ is encoded without access to $Y$. At the decoder, side information $Y$ is available and is used to obtain a reconstruction $\hat{X}$ of $X$. This way of coding is asymmetric as $Y$ is independently encoded and decoded while $X$ is independently encoded but conditionally decoded. Wyner and Ziv concluded that there is a rate loss incurred when the SI is not available at the encoder. Their theorem states that when two jointly Gaussian i.i.d. memoryless random sources are independently encoded and the mean-squared error distortion measure is used, there is no coding efficiency loss compared to joint encoding even if the coding process is lossy. The Slepian-Wolf and Wyner-Ziv theorems suggest that two statistically correlated signals can be compressed in a distributed manner without loss of coding efficiency.

In [68], the authors used Wyner-Ziv (WZ) coding for distributed image compression based on CS. The poposed WZ image coding scheme is shown in Figure 3.2. First, random

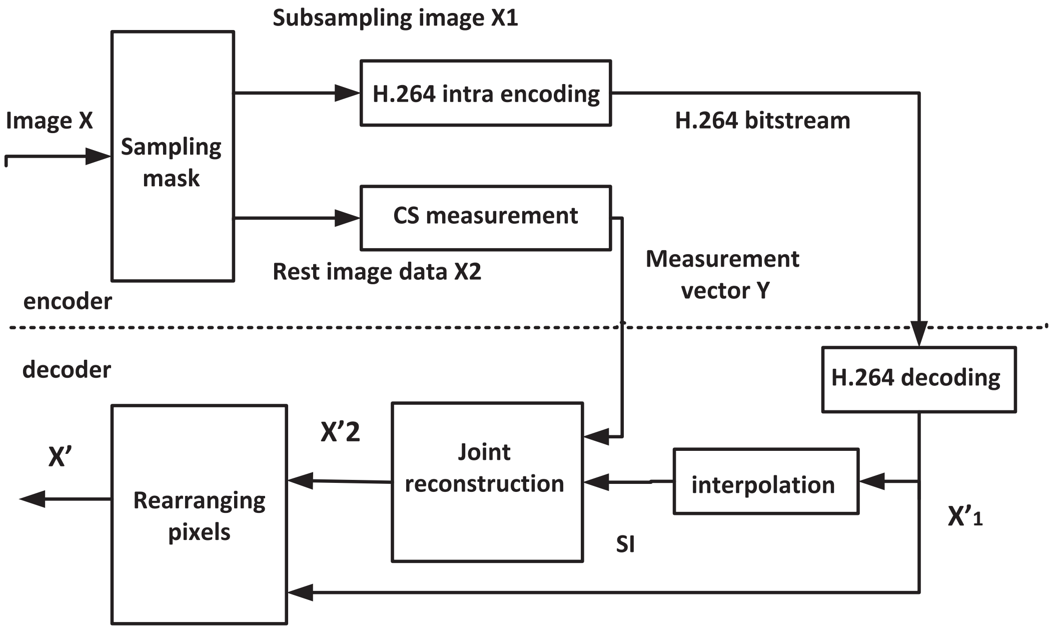


Figure 3.3: Wyner-Ziv Image Coding Architecture [69].

projections of the image are obtained by using a scrambled fast Fourier transform. Then a nested scalar quantization is applied to each measurement. The quantizer consists of a coarse coset channel coder nested in a fine uniform scalar quantizer. A JPEG decoded image is used at the decoder, which is also randomly measured using CS and served as the side information to recover the image. The requirement for a JPEG decoded image as the side information is not very practical. If the JPEG decoded image is available at the decoder, then there is no need for the CS decoded one. Encoding the same image twice will also increase the computational burden at the encoder.

Another distributed approach proposed in [69] first creates two down-sampled images $X_1$ and $X_2$ from the original image $X$. The codec architecture is shown in Figure 3.3. $X_1$ is encoded by H.264 intra-frame coding and $X_2$ is encoded using CS. At the decoder, the H.264 encoded $X_1$ is decoded using intra-frame decoding to obtain $\hat{X}_1$. The CS encoded image $X_2$ is reconstructed with the help of side information generated from an interpolation of $\hat{X}_1$. The main disadvantage of this approach is that it still requires a conventional H.264 codec. That defeats the purpose of having a simple encoder using CS.

None of the above distributed approaches make use of the fact that individual image blocks have different sparsity. One approach that takes advantage of this can be found in [70]. Images blocks are classified either as flat or non-flat. Flat blocks are ones with low frequency contents that can be reconstructed using a lower sampling rate. Non-flat blocks contain texture and edges and require a higher sampling rate for successful reconstruction. The mean $\mu_i$ and variance $\sigma_i^2$ of each block $i$ are computed using the following:

$$\mu_i = \frac{1}{n_1 \times n_2} \sum_{m_1=1}^{n_1} \sum_{m_2=1}^{n_2} b_{m_1 m_2} \tag{3.3}$$

$$\sigma_i^2 = \frac{1}{n_1 \times n_2} \sum_{m_1=1}^{n_1} \sum_{m_2=1}^{n_2} (b_{m_1 m_2} - \mu_i)^2 \tag{3.4}$$

where $\mu_i$ and $\sigma_i^2$ is the mean and variance of each block. $n_1 \times n_2$ is the block size of each image block and $b_{m_1 m_2}$ is the CS measurements of each image block. Define $t_i$ as

$$t_i = \begin{cases} 1, & \sigma_i^2 \geq \lambda\sigma^2 \\ 0, & \sigma_i^2 < \lambda\sigma^2 \end{cases} \tag{3.5}$$

Here, where $\sigma^2$ is the variance of the whole image. If $t_i = 1$, the block $i$ is considered non-flat. Otherwise, the block is flat. The value of $\lambda$ controls the percentage of non-flat blocks. To recover the full image, individual blocks are reconstructed from their CS measurements using the OMP algorithm [31]. Blocking artefacts between flat and non-flat blocks are reduced by mean filtering. After mean filtering, the adjacent columns and rows are added to each block and total variation (TV) minimization is performed. As a result, the computational complexity at the decoder is quite high. Furthermore, in a real CS image acquisition system, the original image data are not available and so this method is not applicable.

A somewhat related approach is recently proposed in [71]. Its architecture is shown in Figure 3.4. An image block is classified as either compressible or incompressible using

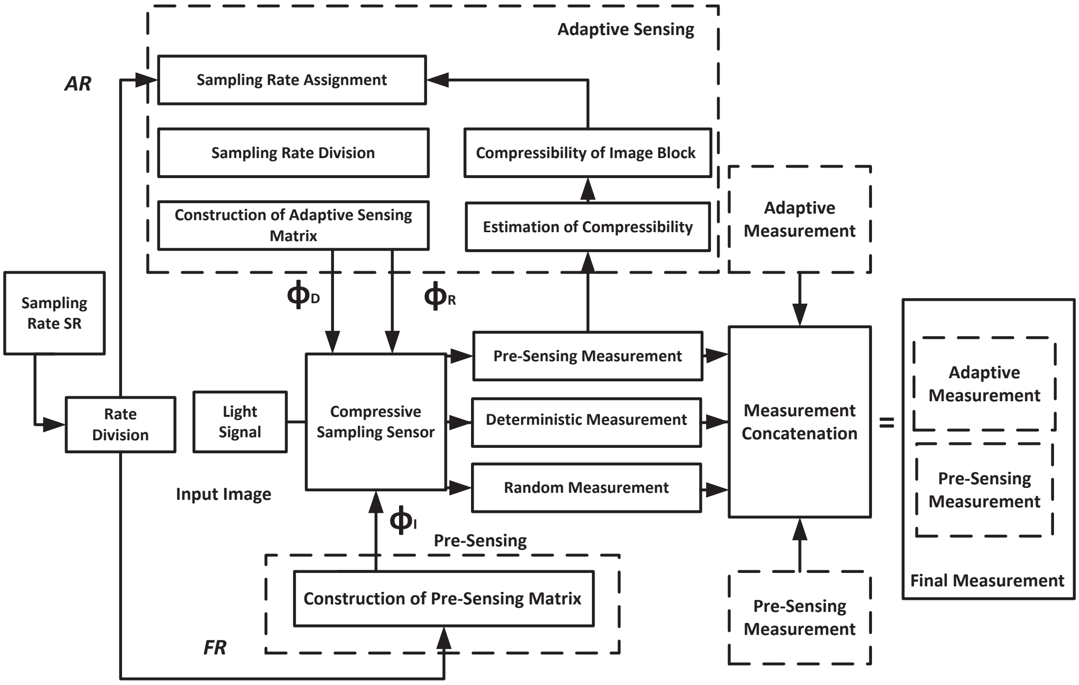


Figure 3.4: Architecture of Adaptive Distributed Image Sensing [71].

the standard deviation (STD) of its CS measurements. The blocks having bigger STD are classified as incompressible, and the ones with lower STD are compressible. If the total adaptive sampling rate is $S$, then the sampling rate $AS_i$ of block $i$ is obtained by

$$AS_i = \frac{P_i}{P} \times S \tag{3.6}$$

where

$$\begin{aligned} P_i &= STD(x_i)/(\sum_{i=1}^{N} STD(x_i + C)) \\ P &= \sum_{i=1}^{N} P_i \end{aligned}$$

and $C$ is a constant to adjust the sampling rate between image blocks. After adaptively assigning measurement rate to each image block, further CS is performed using deterministic and random sensing. Principle Component Analysis is used for deterministic sensing and the sensing matrix is trained off-line by a sample set. Three CS sensing matrices formed the full sensing matrix for an image block. This approach is better than that in [70] since it uses the CS measurements for computing the classification parameter rather than the original image data. The disadvantage is that the need to pre-sense each image block. Also, training is required for deterministic sensing which limits the practical usage of the this scheme.

For effective and simple CS based image coding, it is desirable to use a block-based strategy which exploits the statistical structure of the image data. In general, the distributed approach which applies different measurement rates for different blocks provides much better results. However, it is not easy to estimate the sparsity or compressibility of an image block without applying a sparsifying transform. A new distributed CS image coding method is presented in Chapter 6 that overcomes some of these problems.

## 3.2 Compressed Sensing Video Coding

Research into the use of CS in video coding has started only recently but a number of different methods have already been proposed in the literature. These methods differ in various ways. Some encode a whole video frame while others use a block-based approach. Some use only CS encoding [14, 16, 18, 72–76] while others are hybrid approaches combining conventional coding (MPEG/H.264/AVC) and CS coding [13, 15, 17, 77]. Some methods involve motion estimation or motion compensation (ME/MC) [14, 74–76, 78] while others do not. Some methods make use of a feedback channel from the decoder to the encoder to improve coding efficiency [13, 73], while others use an adaptive scheme to achieve the same [79, 80]. A brief review of the main schemes is presented here.

### 3.2.1 3D Transform Coding

CS based 3D transform encoding methods typically use a 3D transform on a group of video frames or applied at the decoder to estimate the correlation between frames. The first ever use of 3D transform based CS in video processing is proposed in [18]. Their approach is based on the single pixel camera [42]. The image is assumed to be changing slowly enough that a group of one-pixel snapshots constitutes a single frame. 3D wavelets are used as a sparsity-inducing basis for measurement acquisition. A group of frames is jointly reconstructed using a matching pursuit algorithm. The correlation between video frames is exploited purely by the sparsity of the transform. A similar approach is used in [81] to reconstructs a dynamic MRI volume using a temporal Fourier transform coupled optionally with a spatial wavelet transforms as 3D sparsity basis.

In [16], 3D transform is used in conjunction with motion estimation at the decoder. The encoder only takes random CS measurements independently for each frame with no additional compression. A multi-scale framework has been proposed for reconstruction which iterates between motion estimation and sparsity-based reconstruction of the frames. It is centred around the lifting-based invertible motion adaptive transform (LIMAT) method for standard video compression [82]. LIMAT uses second generation wavelets to build a fully invertible transform. To incorporate temporal redundancy, LIMAT adaptively apply motion-compensated lifting steps. Let $k$-th frame of an $n$ frame video sequence be denoted by $x_k$. The lifting transform partitions the video into even frames $x_{2k}$ and odd frames $x_{2k+1}$ and attempts to predict the odd frames from the even ones using a forward motion compensation operator. Motion between a pair of frames is estimated by block matching (BM). The BM algorithm divides the reference frame into non-overlapping blocks. For each block in the reference frame the most similar block of equal size in the destination frame is found and the relative location is stored as a motion vector. This approach contrasts with the one in [18] where the reconstruction of a frame depends only on the individual frames sparsity without taking into account any temporal motion. Due to the multi-scale framework, the method is computationally intensive since motion is estimated from the reconstructed frames before CS reconstruction.

Another approach utilizes 3D wavelets and noiselets to apply CS on multiple frames instead of a single frame [83]. A video is divided into 3D image volumes and their CS measurements are obtained by two sub-processes. First, a 3D wavelet transform

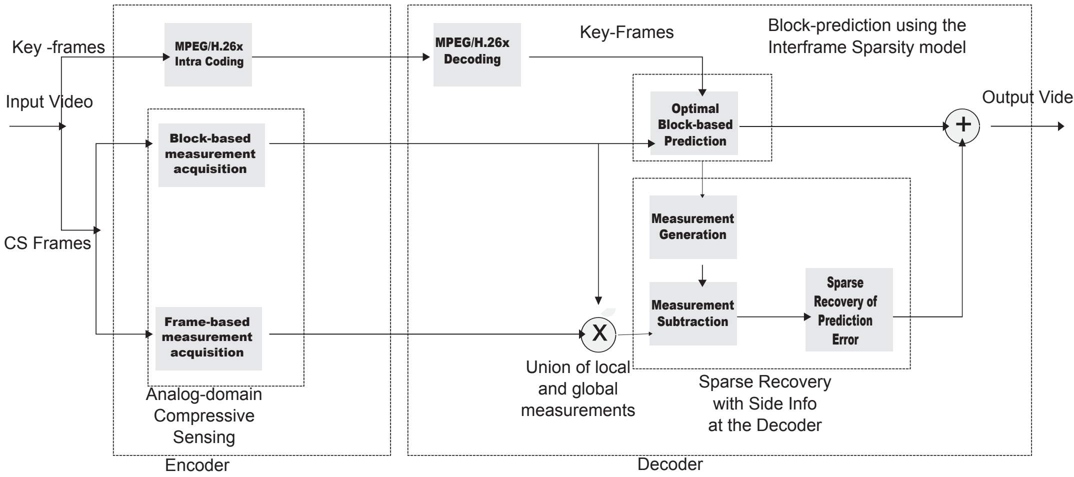

Key -frames
Input Video
CS Frames
MPEG/H.26x Intra Coding
Block-based measurement acquisition
Frame-based measurement acquisition
Analog-domain Compressive Sensing
Encoder
MPEG/H.26x Decoding
Key-Frames
Block-prediction using the Interframe Sparsity model
Optimal Block-based Prediction
Output Video
Measurement Generation
Measurement Subtraction
Sparse Recovery of Prediction Error
X
Union of local and global measurements
Sparse Recovery with Side Info at the Decoder
Decoder

Figure 3.5: Architecture of Distributed Compressed Video Sensing [13].

decomposes a volume into multiple decomposition levels. The coarsest scale is maintained (sampled fully) to form the first set of measurements $y_1$. For the other scales, a 3D noiselet transform is applied which randomly sub-sample the coefficients to form the second set of measurements $y_2$. At the decoder, $y_1$ is inverse transformed to reconstruct the coarse image. Then inverse noiselet transform is applied on $y_2$ to obtain the detailed part of the image. Then TV minimization is performed on the combined reconstructed image. This coding method is very computationally expensive and therefore is not very practical for a resource constraint encoder.

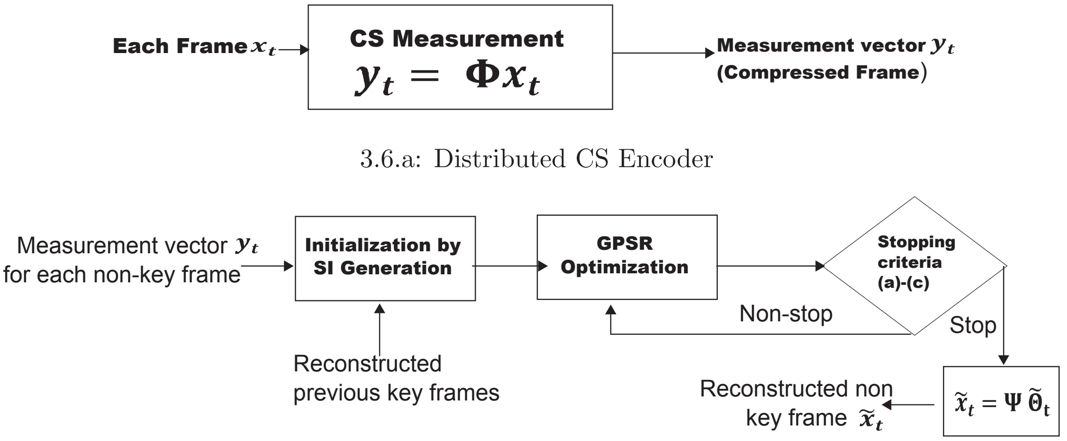
### 3.2.2 Distributed Coding

Distributed coding is based on the distributed source coding theory of Slepian and Wolf [65], and Wyner and Ziv [67]. Source statistics is only exploited at the decoder, not at the encoder as it is done conventionally. Although distributed coding is explicitly used for Wyner-Ziv coding, in CS it is generally used in the context of shifting the encoding burden to decoder. Therefore, some CS video techniques which uses the distributed approach may not apply Wyner-Ziv coding at all.

In [13], a framework which is called Distributed Compressed Video Sensing (DISCOS) is introduced. Figure 3.5 shows the architecture of the DISCOS encoder and decoder.

Video frames are grouped into group of pictures (GOP) consisting of a key frame and a number of non-key frames. For a GOP size of 3, first frame is considered as a key frame while remaining two are considered as non-key frames. Key frames are encoded using traditional MPEG/H.264 encoding. For non-key frames, both local block-based and global frame-based CS measurements are acquired. In this way, more efficient frame-based processing are supplemented by block CS processing so that temporal block motion can be estimated. At the decoder, key frames are decoded using a conventional MPEG/H.264 decoder. For the decoding of non-key frames, the block-based measurements of a CS frame along with the two neighbouring key frames are used for generating sparsity-constraint block prediction. The temporal correlation between frames is efficiently exploited through the inter-frame sparsity model, which assumes that a block can be sparsely represented by a linear combination of few temporal neighbouring blocks. This prediction scheme is more powerful than conventional block-matching as it enables a block to be adaptively predicted from an optimal number of neighbouring blocks, given its compressed measurements. The block-based prediction frame is then used as the side information to recover the input frame from its measurements. The measurement vector of the prediction frame is subtracted from that of the input frame to form a new measurement vector of the prediction error, which is sparse if the prediction is sufficiently accurate. Thus, the prediction error can be faithfully recovered. The reconstructed frame can then be simply obtained from the sum of the prediction error and the prediction frame. This scheme is also related with residual approaches discussed later. The main disadvantage of this scheme is that it still requires a traditional codec for the key frames. Computational burden is also high because it requires both block-based and frame-based processing.

In a similar way, DVC and CS are combined in [14] to simultaneously capture and compress video data. The encoder and decoder designs are shown in Figure 3.6. The main difference from DISCOS is that both key and non-key frames are CS sampled and no conventional MPEG/H.26x codec is required. Key frames are encoded with a higher measurement rate than non-key frames. CS samples of both key frames and non-key frames are obtained using the SBHE sensing matrix [53]. At the decoder, the


Each Frame $x_t$
CS Measurement
$y_t = \Phi x_t$
Measurement vector $y_t$
(Compressed Frame)
Measurement vector $y_t$ for each non-key frame
Initialization by SI Generation
GPSR Optimization
Stopping criteria (a)-(c)
Non-stop
Stop
Reconstructed previous key frames
Reconstructed non key frame $\tilde{x}_t$
$\tilde{x}_t = \Psi \tilde{\theta}_t$

3.6.a: Distributed CS Encoder

3.6.b: Distributed CS Decoder

Figure 3.6: Distributed CS Encoder/ Decoder [14]

key frames are reconstructed using the standard GPSR algorithm [34]. For the non-key frames, in order to compensate for lower measurement rates, side information is first generated to aid in the reconstruction. Side information is generated from motion compensated interpolation from the neighbouring key frames. In order to incorporate side information, GPSR is modified with a special initialization procedure and stopping criteria are incorporated. The convergence speed of the modified GPSR has been shown to be faster and the reconstructed video quality is better than using original GPSR algorithm.

The authors in [84] proposed a novel Wyner-Ziv coding of video using CS. The method consists of a joint sparse model of block-based CS measurements, quantization and entropy coding. Video frames are divided into key and WZ (non-key) frames. Key frames are coded using H.264 intra coding. WZ frames are divided into blocks. Two block sizes of 32 × 32 and 16 × 16 pixels have been used. Let a WZ frame be divided into $B_n$ blocks and there are a total of $T_n$ CS measurements for this frame. Then for a given WZ block $B_{wz}(i)$ with variance $\sigma(i)$, the number of measurement $M_{cs}(i)$ for the block can be approximated as

$$M_{cs}(i) \approx \text{round}\left[\frac{\sigma(i)}{\sum\limits_{i=1}^{B_n}\sigma(j)}T_n\right] \tag{3.7}$$

If $M_{cs}(i)$ is zero, then the encoder is in SKIP mode otherwise it is in CODE mode. The CS measurements will be discarded in SKIP mode. At the decoder, key frames are decoded as H.264 decoding. For a WZ frame, side information is calculated using motion compensated frame interpolation (MCFI) of decoded key-frames and then reconstructed using joint sparse model with SI and CS data. This approach is similar to the adaptive approach (which will be discussed later) to improve coding efficiency of WZ frames. The problem with this approach is that all the blocks needs to be sampled first before the total variance can be computed, which limits the benefits of bock-based encoding. It is also not clear whether the variance of the CS measurements or that of the pixel values is used. If pixel variance is used then it is impractical because in general, CS acquisition does not have access to pixel values. On the other hand, if CS variance is used then there is no need to throw away CS measurements in SKIP mode which has already been acquired.

Another distributed Wyner-Ziv coding system is proposed in [85] which assumes the existence of a feedback channel from the decoder to the encoder. Key frames are encoded and decoded using conventional Intra frame coding. For WZ frames, Wyner-Ziv coding is performed using Slepian-Wolf coding and syndrome bits are generated. To recover the WZ frames at the decoder, side information is generated from the reconstructed key frames using extrapolation. More bits can be requested to improve the reconstruction quality using a feedback channel. This scheme is not applicable to situations where a feedback channel does not exist.

The authors in [76] have proposed a distributed video coding architecture using CS. Wynzer-Ziv coding principles are used at both the encoder and decoder. At the encoder, video frames are sampled using CS. The measurements of the key frames are first quantized using an $M$-level uniform quantizer. $M$ bit-planes are then encoded using a rate compatible LDPCA encoder. At the decoder, the key frames are reconstructed using the GPSR algorithm. To decode each bit-plane, the buffer sends a small part of accumulated syndromes of the current bit-plane to the decoder. Side Information is generated from adjacent key frames using motion compensated interpolation. The LDPCA decoder requests more accumulated syndrome from the buffer using a feedback channel in case of bit-plane decoding error. After all bit-planes have been decoded successfully, the decoded measurements are then de-quantized and used for key frame recovery. The soft input information for the LDPCA decoder for key frames is based on previous decoded key frames using a Gaussian model. The coding of non-key frame is similar to that of key frames but the soft input information of the LDPC decoder is derived from the measurements of the side information. Since this scheme requires a feedback channel, it cannot be applied when it does not exist. Moreover, due to iterative decoding of LDPCA, the system delay will be longer as first the system will decode LDPCA codes and then decoding of CS measurements will take place.

### 3.2.3 Dictionary-based Coding

In dictionary-based approaches, a dictionary is formed from reconstructed frames at the decoder to better reconstruct non-key frames. A dictionary based distributed approach to CVS has been reported in [17]. Key frames are encoded and decoded using conventional MPEG/H.264 techniques. Non-key frames are divided into non-overlapping blocks of $n \times n$ pixels. Each block is then compressively sampled and quantized. At the decoder, key frames are MPEG/H.264 decoded while the non-key frames are dequantized and recovered using a CS reconstruction algorithm with the aid of a dictionary as side information. The dictionary is constructed from the decoded key frames. It's architecture is shown in Figure 3.7.

Two different coding modes are defined. The first one is called the SKIP mode. This mode is used when a block in a current non-key frame does not change much from the co-located decoded key frame. Such a block is skipped for decoding. The decision to enter SKIP mode is dependent on the mean squared error (MSE) between decoded key frame block and current CS frame block, which is done using a feedback channel. If the MSE is smaller than some threshold, the same decoded block is simply copied to

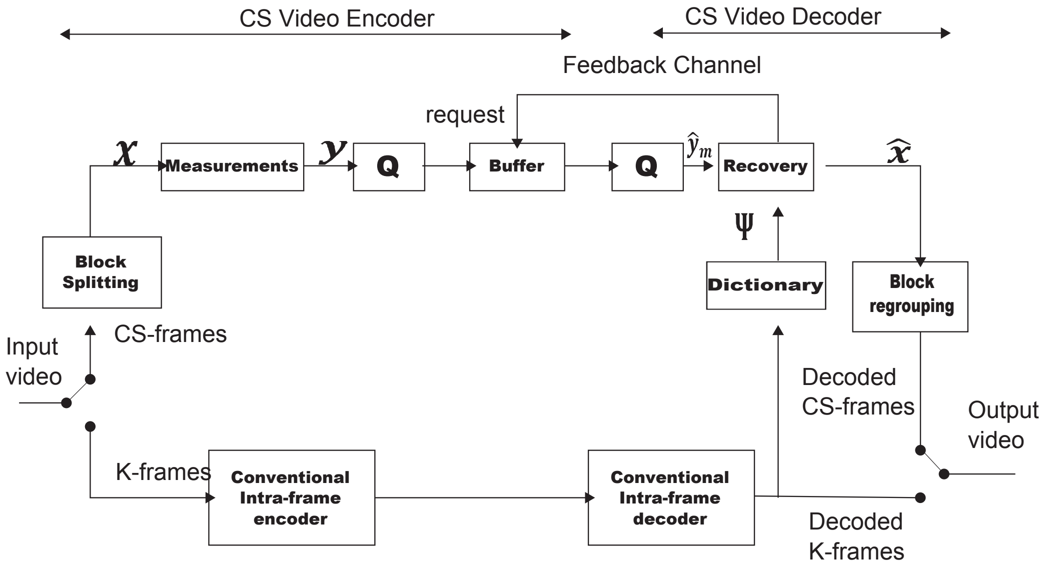


Figure 3.7: CS based Video Coder [17].

current frame and hence the decoding complexity is minimal. The other coding mode is called the SINGLE mode. CS measurements for a block are compared with the CS measurements in a dictionary using the MSE criterion. If it is below some pre-determined threshold, then the block is marked as a decoded block. A dictionary is created from the set of spatially neighbouring blocks of previous decoded neighbouring key frames. A feedback channel is used to communicate with the encoder that this block has been decoded and no more measurements are required. For blocks that are not encoded by either SKIP or SINGLE mode, normal CS reconstruction is performed. The problem with this approach is that a feedback channel is required. Also, to decide whether a block is to be coded in any mode, decoded blocks in the key frame needs to be CS encoded so that the Minimum Mean Square criterion using CS measurements can be computed. It is an extra computation at the decoder. It is also not clear how the threshold for block selection can be determined. Another dictionary based approach is presented in [72]. The authors proposed the idea of using an adaptive dictionary. The dictionary is learned from a set of blocks globally extracted from the previous reconstructed neighbouring frames together with the side information generated from them. Frame are divided into key frames and CS frames. Frame-based CS measurements are acquired for key frames. For CS frames, block-based CS measurements are acquired. At the decoder, the reconstruction of a frame or a block is performed using the sparse reconstruction by separable approximation (SpaRSA) algorithm [36]. Adjacent frames in the same scene of a video are similar. Therefore a frame can be predicted by its side information which can be generated from the interpolation of its neighbouring reconstructed frames. The side information $I_t$ for a CS frame $x_t$ is generated from the motion-compensated interpolation of its previous $x_{t-1}$ and next reconstructed key frames $x_{t+1}$. To learn the dictionary from $x_{t-j}, I_t$ and $x_{t+j}$, for each block, 9 training patches from the nearest 8 blocks overlapping this block and this block itself are extracted. After that, the K-SVD algorithm [86] is applied to the 9 training patches to learn the dictionary $D_t$. $D_t$ is an over complete dictionary. By using the learned dictionary, each block $b_{ti}$ in $x_t$ can be sparsely represented as a sparse coefficient vector $a_{ti} \in R^{P\times 1}$. This learned dictionary provides sparser representation for the frame than using the fixed basis dictionary. This work has been extended to incorporate dynamic measurement rate allocation by incorporating a feedback channel [73]. Dictionary learning using training patches can significantly slow down the reconstruction process.

### 3.2.4 Residue-based Coding

The idea of residue-based approaches is similar to encoding motion information at the encoder for a conventional codec. Instead of encoding motion, for a non-key frame only its residue is encoded. An detailed explanation for CS based prediction is provided in [74]. In order to generate a small residue, a prediction for the original frame $x$ must be as close to the MC prediction as possible. Mathematically it can be expressed as the following optimization problem:

$$\tilde{x} = \arg \min_{p \in P(x_{ref})} ||x - p||_2^2 \tag{3.8}$$

where $P(x_{ref})$ is the set of all ME/MC predictions that can be formed from reference frame $x_{ref}$. As the original $x$ is not available at the decoder in CS, an approximation of it is obtained by using an initial reconstruction $\hat{x}$. Thus the above optimization now becomes

$$\tilde{x} = \arg \min_{p \in P(x_{ref})} ||\hat{x} - p||_2^2 \tag{3.9}$$

Another approach is to use the measurements $y$ and obtain the CS measurements of the MC frame. In this case, the problem becomes

$$\tilde{x} = \arg \min_{p \in P(x_{ref})} ||y - \Phi p||_2^2 \tag{3.10}$$

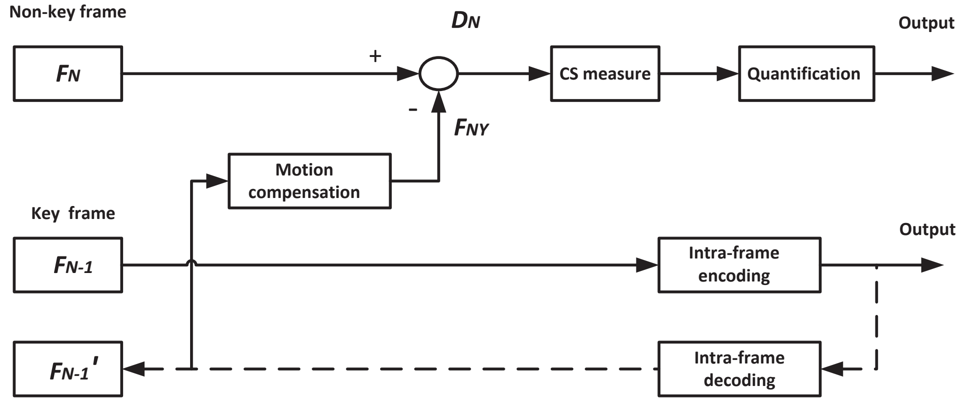
This approach is known as the single hypothesis approach. For better prediction, an number of predictions from all the hypotheses can be obtained instead of using a single prediction.

$$\tilde{w_{t,i}} = \arg \min_{w} ||y_{t,i} - \Phi H_{t,i} w||_2^2 \tag{3.11}$$

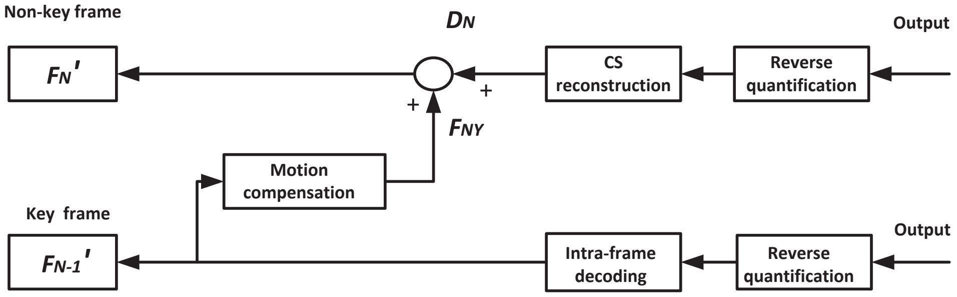
where $i$ is the block index and $t$ is the temporal frame index. $H_{t,i}$ is a matrix of dimensionality $B^2 \times K$ consisting of all possible blocks in reference frame. In this context, $\hat{w}_{t,i}$ is a column vector which represents a linear combination of the columns of $H_{t,i}$. This optimization problem can be solve by $l_1$ minimization as is done in [13,17]. Alternatively, it can be solved using a Tikhonov regularization [87] as proposed in [88].

A different residual CS video coding scheme is proposed in [89]. The architecture is shown in Figure 3.8. At the encoder, before applying CS encoding to the key frames, motion compensation is performed between two key frames to provide prediction for the non-key frames in-between. This prediction is then subtracted from the non-key frames to create a residue which is then CS encoded. At the decoder, after decoding the key frames using intra-frame decoding, motion estimation is performed to predict the non-key frames. This prediction is used in the CS reconstruction of the non-key frames to improve their quality. Since the method requires motion compensation at both encoder and decoder, computational demand is high at both ends.

A better residual technique is proposed in [75] which incorporates ME/MC at the decoder rather than at the encoder. The authors proposed an iterative process of calculating the residue and then improving it using backward and forward motion estimation.



3.8.a: Residual CS Encoder



3.8.b: Residual CS Decoder

Figure 3.8: Residual CS Encoder/ Decoder [89]

A GOP consists of a key frame and some non-key frames. Each frame is CS sampled in a block by block basis. At the decoder, each key frame is reconstructed from its block-based CS measurements. For a non-key frame $x$, an initial reconstruction is performed using its block-based CS measurements to yield $\hat{x}$. Then, full search block based ME is performed using the reconstructed key frame and $\hat{x}$. This motion compensated prediction forms the field of motion vectors and produces a motion compensated frame $\hat{x}_{mc}$. Then for each block $j$ in the non-key frame, compute the residue $y_{r_j}$ by

$$y_{r_j} = y_j - \Phi_B \hat{x}_{mc_j} \tag{3.12}$$

where $\Phi_B$ is the block sensing matrix. If the MC is accurate, then the reconstructed residual frame $\hat{x}_r$ is sparser than original $x$. This $\hat{x}_r$ is then added to $\hat{x}_{mc}$ to form a new

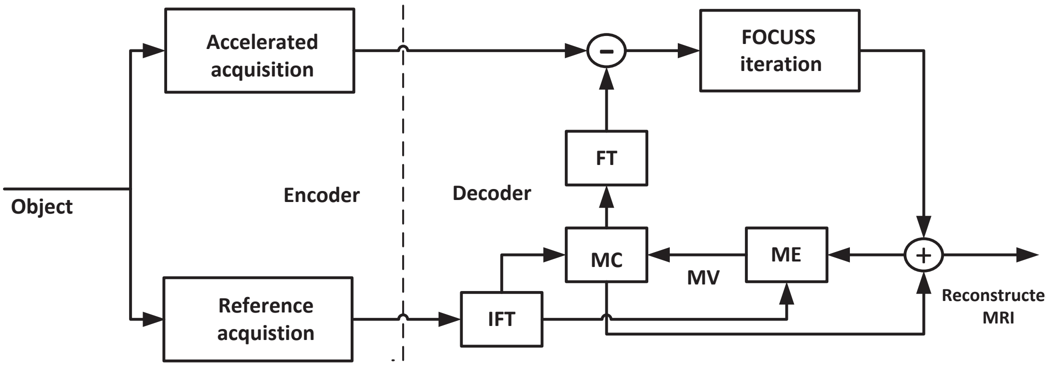


Figure 3.9: *k*-*t* FOCUSS with ME/MC [78].

approximation $\hat{x}$. This approximation is of better quality than the initial approximation created by direct CS reconstruction. This process can be further improved by iteratively repeating this ME. Instead of using a single key frame for ME, backward and forward motion estimation can be performed on a GOP for more accurate results. The advantage in this approach is that all the computation burden is at the decoder, thus keeping the encoder simple. This method has been extended by using a multihypothesis approach [88]. Instead of using a single prediction, multiple distinct predictions are created and then combined to form a composite prediction which provides better results.

Another residual reconstruction technique is specific to dynamic Magnetic Resonance Imaging (MRI) videos. The strategy is to perform residual reconstruction from a prediction of the current frame. A general assumption specific to MRI videos is that the content changes very slowly between adjacent frames in comparison for normal videos. In [90, 91], sparsity of the current reconstructed MRI frame is estimated. Then it is used to predict the sparsity of subsequent frames. Estimation can be performed on the basis of least squares [91] or based on Kalman filtering [90]. The approach used in [92] is based on "Modified-CS" [93]. It uses an $l_1$ solution outside the currently known sparsity pattern and then estimate the changes from frame to frame. The main difference between these approaches and the common video approaches is that they do not perform explicit ME/MC to improve the residual reconstruction. Instead, kalman filtering, least squares or $l_1$ optimization are used to track the changes. They are more suitable for videos with slow object motions like dynamic MRI.

An approach that explicitly use ME/MC, called focal under determined system solver in *k*-*t* space (*k*-*t* FOCUSS), is proposed in [78, 94] for MRI videos. Its architecture is shown in Figure 3.9. It incorporated ME/MC to improve the reconstruction quality of non-key frames by residual and motion estimation prediction from available key frames. Full sampling is used for key frames. A larger GOP size is adopted as motion is very slow in MRI videos.

### 3.2.5 Adaptive Coding

In CS based video codecs, usually a small number of pre-determined measurement rates for each frame or block is use. For example, one measurement rate may be used for key frames and another for non-key frames. However, if the sparsity of the video frames or blocks can be determined, then the correct number of CS measurements can be used to represent them. This is the idea of the adaptive approach.

The architecture of an adaptive block based block-based CS video codec is shown in Figure 3.10 [15]. With this scheme, each video frame is split into $B$ non-overlapping blocks each of size $n \times n = N$ pixels. A reference frame (or key frame) is sampled encoded using conventional methods like MPEG, H.264. For a certain GOP size, first frame is considered as a key frame while remaining frames are considered as non-reference frames. For non-reference frames, after sampling, a compressed sensing test is applied to identify sparse blocks. The $B_s$ sparse blocks are each compressively sampled using an i.i.d. Gaussian measurement matrix and an inverse DCT sensing matrix. The remaining $B - B_s$ blocks are sampled in the traditional way. Signal recovery is performed by the OMP algorithm. In reconstructing compressively sampled blocks, all sampled coefficients with an absolute value less than some constant $C$ are set to zero. Theoretically, if there are $N - K$ non significant DCT coefficients, then at least $M = K + 1$ samples are needed for signal reconstruction [19]. Therefore the threshold is set to be $T < N - K$. The choice

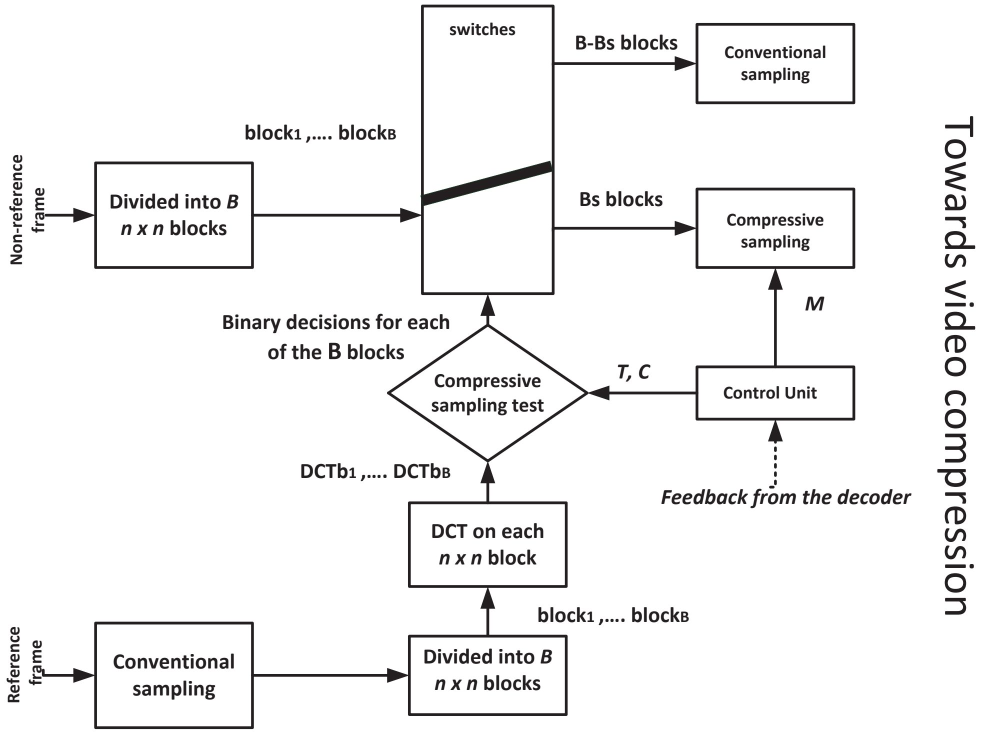


Figure 3.10: Block based Video Codec [15].

of values for $M, T$, and $C$ depends on the video sequence and the size of the blocks. It has been shown experimentally that up to 50% of saving in video acquisition is possible with good reconstruction quality. The disadvantage with this approach is that a conventional codec is required for reference frames. Furthermore, it is difficult to determine a suitable global threshold.

The adaptive CS based video scheme proposed in [80] used a similar idea of weighting CS measurements as in [56]. A support set of transform coefficients are obtained by reconstructing the first frame. Then a weighting matrix is computed which is used to adaptively locate the significant transform coefficients. As adjacent frames are highly correlated, it is very likely that their transform coefficients will also be located at the same locations. The weighting matrix is then used to adaptively sample the next frame. This approach requires CS reconstruction at the encoder which significantly increases its

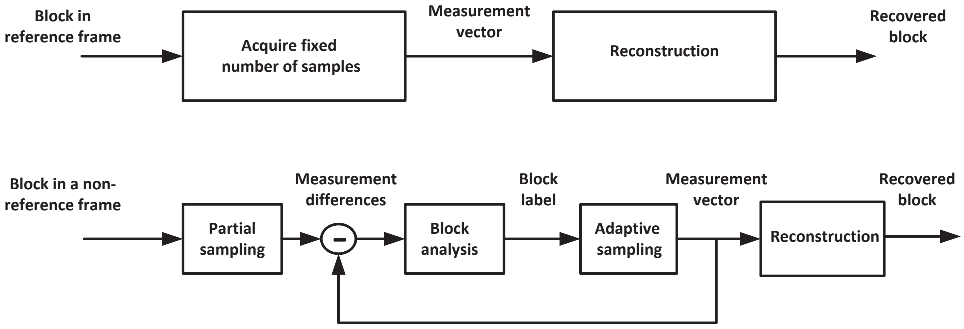


Figure 3.11: Adaptive Block based Video Codec [79].

complexity.

Another adaptive architecture is proposed in [79] and shown in Figure 3.11. Each video frame is divided into non-overlapping blocks of equal sizes. Blocks are classified into several different types and sampled and reconstructed differently. This framework can adaptively adjust the number of measurements for each image block according to its sparsity. The first frame is considered as a reference frame and the blocks in this frame are sampled and reconstructed using regular CS techniques. For a block $k$ in a non-reference frame $t$, initially $M_0$ measurements using a sensing matrix $\Phi_{M_0}$ are collected. These measurements are given by

$$y^k_{M_{0,t}} = \Phi_{M_0} x^k_t \tag{3.13}$$

Then the difference between the CS measurements of the current block and previous frame block is computed as

$$y^k_d = y^k_{M_{0,t}} - y^k_{M_{0,t-1}} \tag{3.14}$$

Then, $l_1$ norm of $y^k_d$ is computed and normalized by $M_0$. It is compared with two thresholds $T_1$ and $T_2$. This block is classified as

(i) a static block if $||y^k_d||_1/M_0 \leq T_1$. No more measurements are acquired for this block.

(ii) a small-change block if $||y_d^k||_1/M_0 \leq T_2$. An additional $M_1 > M_0$ measurements are acquired.

(iii) a high dynamics block if $||y_d^k||_1/M_0 > T_2$. An additional measurements $M_2$ are acquired.

To determine the value of $M_2$, some frames called "indicator frames" are reconstructed periodically at the encoder to estimate the sparsity of the current high dynamic block. At the decoder, static and high dynamic blocks are reconstructed using their respective CS measurements. For small change blocks, first the block difference $\hat{x}_{t,t-1}^k$ is reconstructed and then it is added to the reconstructed block $k$ at frame $t-1$, denoted as $\hat{x}_{t-1}^k$ to obtain

$$\hat{x}_t^k = \hat{x}_{t-1}^k + \hat{x}_{t,t-1}^k \tag{3.15}$$

This scheme can significantly improve CS video coding performance with the added complexity at the encoder. The disadvantage is this approach that encoder complexity is increased in terms of memory (to store previous CS measurements) and computation (to reconstruct indicator frames). Also it is not easy to determine the correct thresholds for different types of video content.

In [95], a new image/video coding framework is proposed which combines CS with traditional image/video compression approaches. It considers CS sampling/recovery as more suitable for image blocks with sparse gradients while conventional DCT based method is more suitable for complicated image blocks. Therefore, CS is integrated into JPEG and H.264/AVC coding methods as a new coding mode and rate-distortion optimization (RDO) is employed for mode selection between the new coding mode and conventional coding modes. Therefore, each 8 × 8 image block is encoded and decoded in either the conventional coding modes or the CS coding mode. The sensing operator is a partial DCT matrix consisting of the top rows of the full DCT matrix and is applied to the block residue to reduce the bit rate. At the decoder, TV-minimization is used to recover the block residue from the coded coefficients. This work is later extended to include pixel domain TV-minimization at encoder to more effectively improve coding efficiency [77].

### 3.2.6 Scalable Coding

Scalable Video Coding (SVC) is an extension of the H.264/AVC standard which provides different coding strategies in different layers for different target audiences [96]. This is achieved by splitting the video stream into different layers which can be used as temporal, spatial or quality enhancement. This is motivated by applications which involve different screen resolutions, bandwidth requirements, decoding time constraints and quality of service requirements.

Scalable CS coding is first proposed in [97]. A framework of measurement structures has been developed to improve the reconstructed CS video progressively. Let $X$ be an $n \times n$ pixel video frame and $Z$ be its $2-D$ DCT coefficients. If $z$ is an $N$-length column vector obtained by zig-zag scanning of $Z$ then $z = [z_1^T z_2^T \cdots z_L^T]^T$. Here $z_i$ is a vector of length $k_i$ denoting the $i$-th video layer with $\sum_{i=1}^{L} k_i = N$. The first layer $z_1$ is called a base layer and other layers are enhancement layers. Overlapping windows are used to obtain the CS measurements of each layer. Depending on the requirements at the decoder, a video frame can be reconstructed using the measurements of the base layer plus any number of enhancement layers. This is an adaptive approach which adaptively selects significant DCT coefficients arranged in a zig-zag order. Further each video frame is reconstructed independently without exploiting any correlation among video frames.

In [98], DVC and CS are combined to form a scalable codec. At encoder, frames are divided into Key frames and WZ frames. For a certain GOP size, first frame is considered as a key frame while remaining frames are considered as non-key frames. Key frames are intra coded using conventional H.264/AVC codec. For the WZ frames, a residue which is the difference between the current frame and the first key frame in a GOP is computed. Then only the residual is encoded using CS. The measurement rate of the WZ frames are adjusted according to the available bandwidth. At the decoder, key frames are reconstructed using conventional H.264/AVC decoder. For WZ frames, the decoder first requests a small number of measurements according to the available bandwidth. The residue is then reconstructed which is combined with the reconstructed

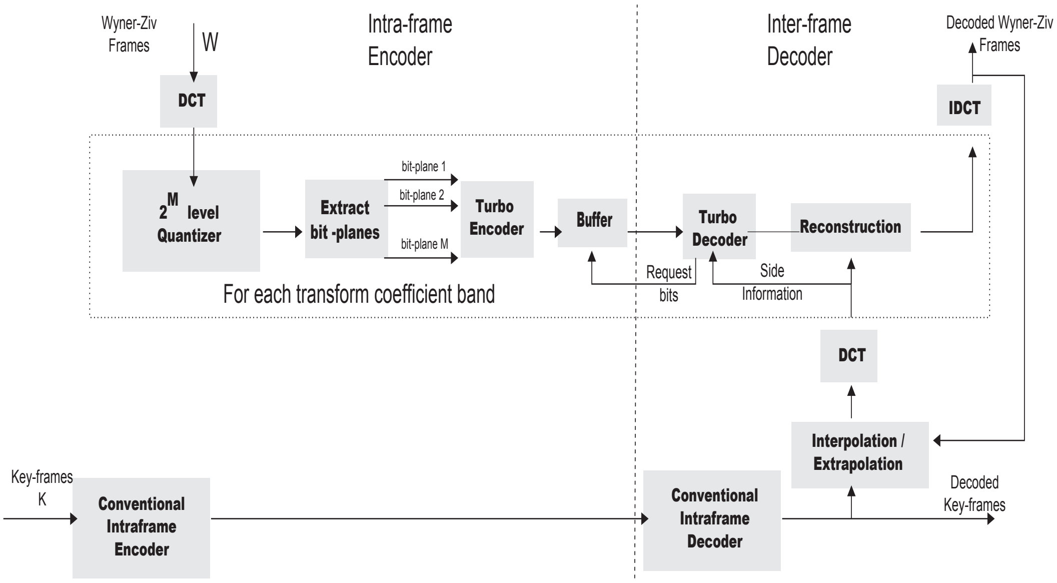


Figure 3.12: Transform Domain Wyner-Ziv video Coding architecture [99].

key frame. Additional measurement can then be subsequently requested to enhance the reconstruction quality. This approach requires a feedback channel and a conventional codec which are not applicable to low complexity encoders.

## 3.3 Conventional Distributed Video Coding

In the previous two sections, different strategies for encoding images and videos are presented. The distributed approach provides low complexity encoding. Distributed Video Coding (DVC) is an application of distributed source coding (DSC) which involves the encoding of two or more dependent random sequences. DSC is based on the Slepian-Wolf theorem [65] and Wyner-Ziv theorem [66, 67] which has been described earlier in Section 3.1.4. In this section, three conventional distributed video coding systems that have been proposed in the literature are discussed. They will provide the benchmark for performance comparisons with the systems proposed in Chapters 5 and 7.

### 3.3.1 Transform Domain Wyner-Ziv Video Coding

The first practical Wyner-Ziv video coding schemes emerged around 2002. One of them is a system developed at Stanford University, initially developed in the pixel domain [100] and subsequently reformulated in the transform domain [99]. Figure 3.12 shows a block diagram of the transform domain system structure.

The encoder divides video frames into key frames and Wyner-Ziv (WZ) frames. Key frames are encoded by traditional coding methods such as those used in MPEG/H.264. For each WZ frame, a block-wise discrete cosine transform (DCT) is computed. The transform coefficients of the whole frame are grouped into bands. These DCT bands are quantized into different levels according to the specified target quality [99]. They are then independently encoded, typically using turbo or low density parity check (LDPC) codes.

At the decoder, key frames are decoded first. Motion estimation is performed by interpolation. An estimated reference frame is generated which is used as the side information for the decoding of WZ frames. The residual statistics between SI and the WZ frame is modelled by a Laplacian distribution. The decoder decodes all the bit planes in a DCT band. After that the bit planes are reconstructed by applying inverse DCT.

### 3.3.2 The PRISM Video Codec

Another DVC system called PRISM (Power-efficient, Robust, hIgh compression Syndrome based Multimedia coding) [102] is developed at the University of California, Berkeley. The architecture of this system is shown in the Figure 3.13.

At the encoder, video frames are divided into blocks of 8 × 8 pixels and a block-wise DCT is computed. As in the Stanford system, a quantizer is used to adjust the target quality. The job of the classifier is to classify each block either as a current block or a predictor block based on their correlation. The prediction can be a co-located block or a

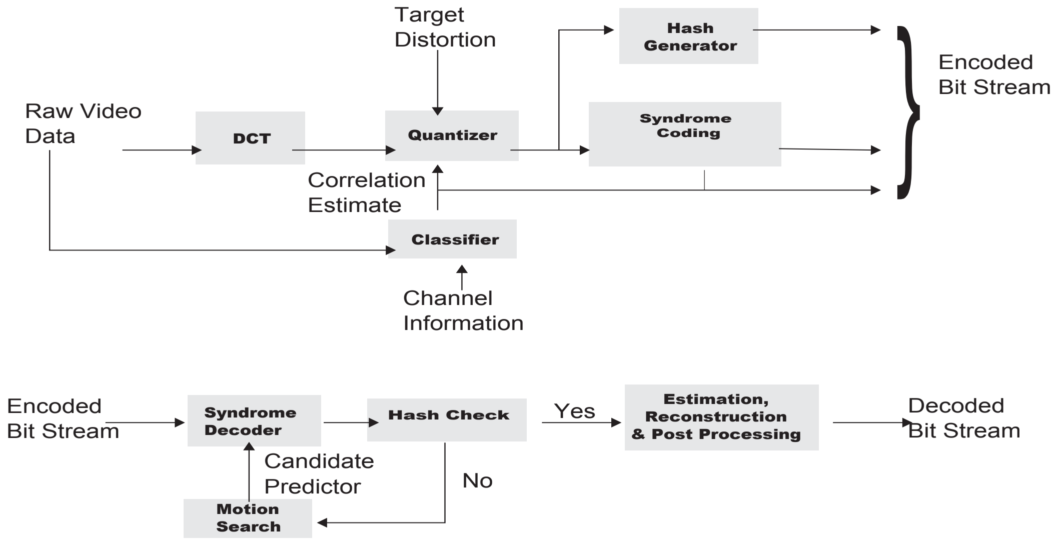


Figure 3.13: PRISM Video Coding Architecture [101]

motion compensated block. There are three types of coding classes used by PRISM and sent as the header information:

(i) SKIP coding class – a block is not coded and the predicted block is used as the current block.

(ii) Entropy coding class – same as traditional intra frame coding.

(iii) Syndrome coding class – Wyner-Ziv coding is used. Only the least significant bits of the transform coefficients are syndrome coded. It uses a 4-tuple scheme. BCH (Bose Chaudhuri Hocquenghem) codes are used for the upper part. For each block, a signature is used as a 16-bit CRC (Cyclic Redundancy Check) checksum.

At the decoder, SI blocks are created using a two step process. The first step deals with bit-planes encoded by traditional entropy coding and the second is for bit-planes that are syndrome coded. After obtaining the quantized coefficients a block is reconstructed with available side information.

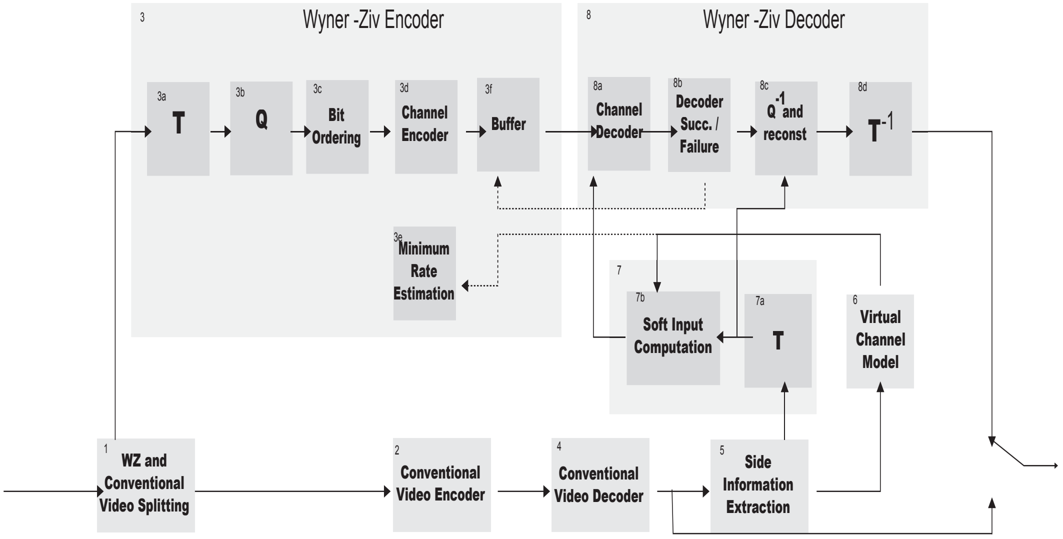


Figure 3.14: Block diagram of the DISCOVER video coding architecture [103].

### 3.3.3 The DISCOVER Video Codec

Distributed Coding for Video Services (DISCOVER) is a DVC architecture that has been proposed in [103]. Its architecture is shown in Figure 3.14.

Video frames are first divided into key frames and WZ frames. Key frames are encoded using traditional video encoding techniques while WZ frames are encoded by distributed techniques. For each WZ frame, a block-wise transform is computed. The transform coefficients are then quantized and grouped into frequency bands. They are channel coded and stored in a buffer. The number of bits needed for encoding from each bit-plane is estimated. This process is called minimum rate estimation. The job of the buffer is to provide additional bits for stable decoding using a feedback channel.

The key frames are decoded by conventional intra-frame decoding. The decoded key frames are then used as the side information for decoding the WZ frames. A motion compensated interpolation is performed for producing the SI for WZ frames. The difference between a WZ frame and its corresponding SI is modelled using a Laplacian distribution. This provides a good approximation of the residual distribution. Information bits are estimated by incorporating the noise modelled from block 6 in Figure 3.14. These information bits are fed into the WZ decoder to decode the WZ frames. A suitable criterion determines success or failure. If the decoding is not successful, the feedback channel is used to obtain more bits.

### 3.3.4 Comparison of DVC Architectures

The similarities and differences between these three DVC systems are summarized as follows.

(i) Frame Classification

In both the Stanford system and DISCOVER, the input video sequence are divided into WZ frames and key frames. In PRISM, there is no classification of frames performed. All video frames are treated similarly.

(ii) Spatial Transformation

In all three architectures, block-based DCT is used. In the Stanford system and DISCOVER, only the WZ frames are transformed. The transform coefficients of each frame are grouped according to their values into bands.

In the Stanford and DISCOVER codecs, after Turbo / LDPC decoding, inverse DCT is performed to decode the WZ frames. In the PRISM codec, a block is reconstructed from the corresponding SI and quantized bit stream.

(iii) Quantization

In the Stanford system and DISCOVER, each DCT band is uniformly quantized with a number of levels that depend on the target quality or on the DCT coefficients. For a given band, bits of the quantized symbols are grouped together, forming bitplanes, which are then independently turbo encoded or LDPC encoded. In the PRISM architecture, a scalar quantizer is used.

(iv) Block Classification

This is only done in PRISM since the other two are frame-based codecs.

(v) Turbo/LDPC Coding

Only turbo encoding is used in the Stanford system while DISCOVER makes use of both turbo and LDPC encoding for coefficient bit-planes. The Turbo/LDPC decoder receives successive chunks of parity bits from the feedback channel. To decide whether more bits are needed for the successful decoding, the decoder uses a simple request stopping criteria which checks that all Turbo/LDPC code parity check equations are satisfied for the decoded codeword. In DISCOVER, a further CRC checking is performed to obtain a good reconstruction quality.

(vi) Syndrome Coding and Hash Generation

This is performed in the PRISM codec only. For the syndrome class, only the least significant bits of the quantized DCT coefficients are syndrome encoded. In addition, for each block, the encoder sends a 16-bit cyclic redundancy check (CRC) checksum as a signature of the quantized DCT coefficients. This is needed in order to select the best candidate block (SI) at the decoder. Candidate blocks are used for syndrome decoding. A hash signature is generated for each decoded candidate block. For successful decoding, the generated hash signature is compared with the CRC hash received from the encoder.

(vii) Side Information Creation

This is an important step in DVC decoding. For both the Stanford and DISCOVER codecs, SI is created by previously decoded key frames using motion compensated frame interpolation. This is an estimate for the WZ frames. The better the estimate, the smaller the number of parity bits needed for correction. In PRISM, motion estimation is performed using a reference frame by positioning a window around the center of block to be decoded.

(viii) Correlation Noise Modelling

The correlation statistics between side information and WZ frames is modelled by the Laplacian distribution. This modelling is needed in both the Stanford system and DISCOVER. Prism does not require this step.

## 3.4 Summary

In this chapter, a review of CS based Image and Video coding is presented. Different CS image coding schemes are classified into different categories and then key points in each category are discussed. Similarly, a classification for different CS video coding schemes is discussed. The differences with the work done in this thesis and available CS image/video literature is also discussed.

# Chapter 4

# Sensing Matrix, Quantization Matrix and Reconstruction Algorithms for Image Compression

In a conventional lossy image compression system, an invertible transform is applied to the image which provides its expansion in terms of transform coefficients. Typically most of the energy of the signal is concentrated in a relatively small subset of the transform coefficients. Consequently, when quantization is then applied to the coefficients, a significant number of quantized coefficients will be zero and therefore need not be encoded. After quantization, a lossless compression process called “entropy coding” encodes the data into a bit stream for storage or transmission. Decompression is performed by inverse quantization followed by inverse transformation. This process is used in JPEG [1]. The choice of transformation and the design of the quantization matrix are important factors in the performance of the compression system.

For a system based on compressed sensing, the process is somewhat different. Instead of applying a transform to the image, a set of linear measurements is obtained through a sensing matrix. The number of measurements is typically much smaller than the original image. Figure 4.1 illustrates this process in block diagram form. Here the measurements $y$ is obtained by applying a sensing matrix $\Phi$ to an image $x$ with a total of $N$ pixels. $\Phi$ is an $m \times N$ matrix where $m$ is much smaller than $N$. Thus the dimension of the measurement

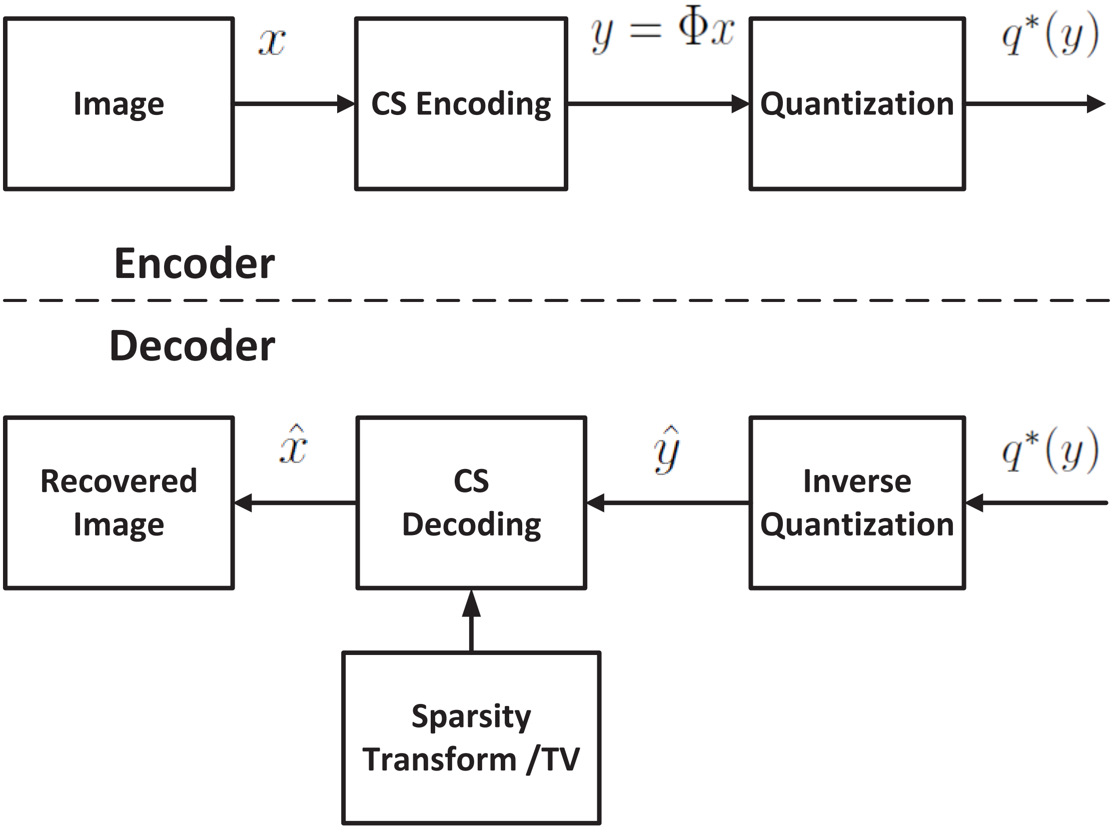


Figure 4.1: CS Image Compression

vector $y = \Phi x$ is $m$. The CS measurements are then quantized and entropy encoded. At the decoder, inverse quantization is followed by a CS recovery process to reconstruct the image. In this case, the performance of such a compression system is determined by the number of measurements, the sensing matrix, the quantization matrix, and the CS reconstruction algorithm.

In this chapter, the effects of the choice of sensing and quantization matrices, and the CS reconstruction algorithms are studied in a non-distributed image compression setting. The efficacy of several different sensing matrices are evaluated in terms of encoding complexity and ease of implementation. A quantization matrix is designed and its performance is evaluated. Finally, several different CS reconstruction algorithms are compared in terms of reconstruction time and reconstruction quality. The results obtained in this chapter is then applied to distributed image and video coding in subsequent chapters.

## 4.1 Choice of Sensing Matrices

In CS imaging applications, the design of efficient sensing (measurement) matrices $\Phi$ is a challenging problem due to data size and computational requirements. There has been many proposals in literature. Some are deterministic while others are randomly generated. They can be classified into the following three categories based on their properties.

**Randomly Generated Matrices** Randomly generated sensing matrices have entries that are identically and independently sampled from a probability distribution. A commonly used distribution is zero-mean, unit variance Gaussian $\mathcal{N}(0, 1)$ [104,105]. Another one is the Bernoulli sensing matrix with entries i.i.d. sampled from a binomial distribution. The Gaussian and Bernoulli matrices have the advantage that they are incoherent with many basis matrices [11]. Random matrices provides good recovery performance as they are incoherent with many sparsifying transforms for imaging application. The limitations of random matrices are high memory requirements and the need for the encoder to communicate the matrix to the decoder.

Random matrices can be made to have specific structures to provide better computation and storage efficiency. For example, Toeplitz matrix or diagonal-constant matrix is a matrix with constant diagonal elements. Results from [106,107] showed that a sparse signal can be recovered from reduced samples exactly with high probability when block Toeplitz random matrices are used for sensing. Another approach is to generate a block-diagonal matrix in which each diagonal matrix is a random matrix. In this way, random matrices will be less dense and require less memory compared to full random matrices.

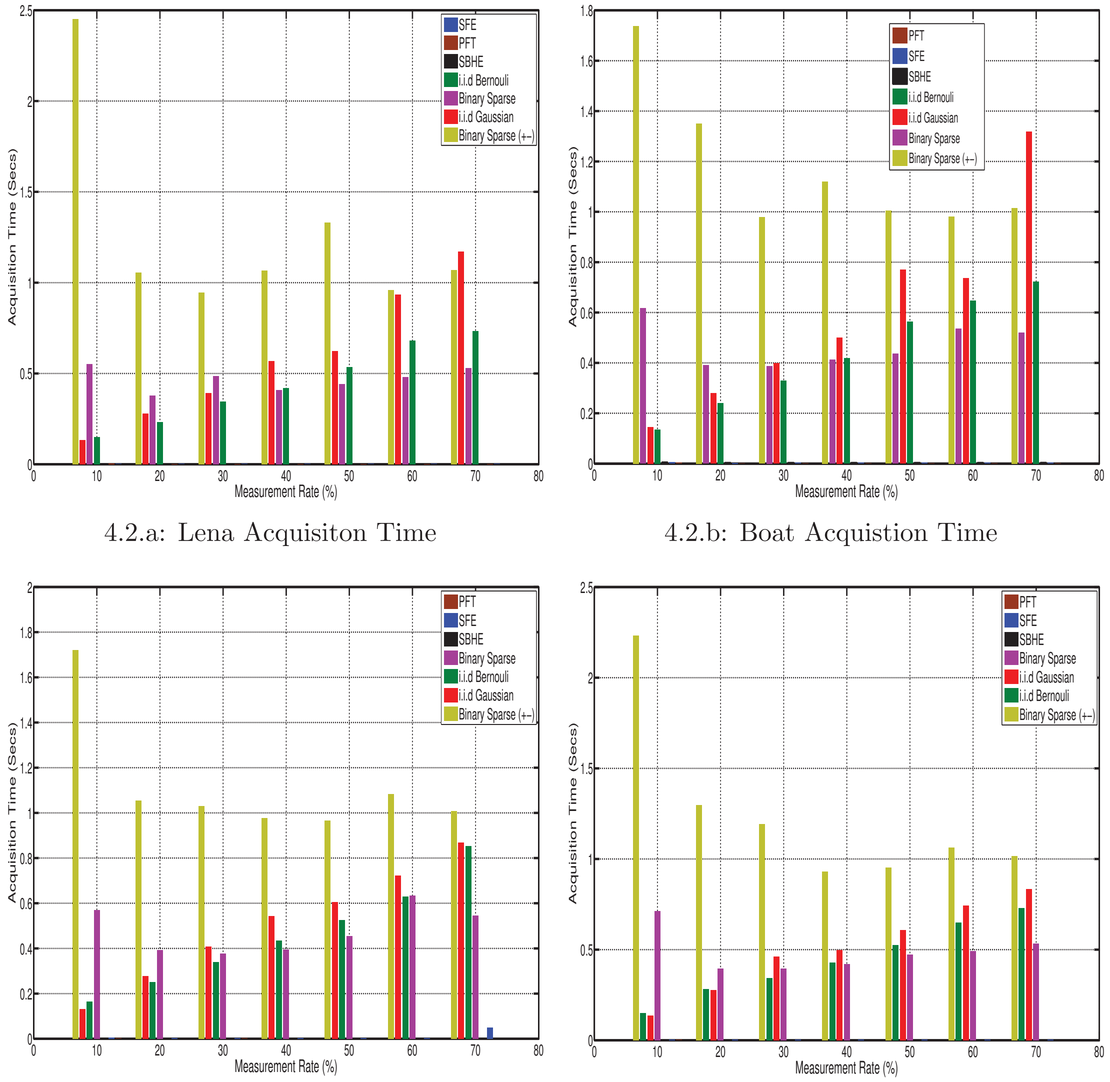
**Transform based Matrices** Transform based sensing matrices are based on linear transforms such as discrete Fourier, Hadamard and discrete cosine transforms. Partial Fourier ensemble sensing matrices are introduced in [104]. The sensing matrix $\Phi$

is obtained by sampling $M/2$ columns uniformly at random from the $N \times N$ discrete Fourier transform matrix, where $M$ is the down-sampling factor. An alternative one is the Scrambled Fourier Sensing Matrix which has better performance [108]. Here the columns of $N \times N$ discrete Fourier transform matrix are first randomly permuted and then $M/2$ columns are selected uniformly at random. In both the partial and scrambled methods, the sensing matrix is a truncated transform with only $M/2$ Fourier coefficients which are randomly chosen. Similarly, block DCT or block Hadamard transform based sensing matrices can also be used instead of Fourier transform. The benefit of Hadamard based sensing matrix is that its entries are binary ±1 which is suitable for hardware implementation. A very popular sensing matrix is the partial block Hadamard transform matrix with randomly permuted columns [53]. The benefits of transform based sensing matrices are their fast computation and that they can be generated at the encoder/decoder without need of explicit storage.

**Binary Sparse Matrices** First proposed in [109], binary sparse matrices have a fixed small number of ones in each column and the remaining entries are zeros. It is well known that a random "dense" sensing matrix works well in CS applications [104]. In [109], it was shown that, both in theory and in practice, $\Phi$ that is binary and "very sparse" is essentially as good as random Gaussian or Fourier matrices for linear programming decoding. At the same time, sparse binary matrices provide additional benefits, such as reduced encoding and decoding time. Experimental results for sparse recovery using binary sparse matrices are reported in [25,109,110].

### 4.1.1 Experimental Results

Four standard 8-bit grey-scale images – "Lena", "Boat","Cameraman" and "MRI" – are used for the evaluation of sensing matrices. These images are down-sampled to a resolution of 64 × 64 pixels from original dimension of 256 × 256 to reduce the memory requirements for random Gaussian matrices. The software provided in the $l1-magic$



4.2.a: Lena Acquisiton Time

4.2.b: Boat Acquistion Time

4.2.c: Cameraman Acquisition Time

4.2.d: MRI Acquisiton Time

Figure 4.2: Sensing Matrix Acquisition Time

software package [111] is used for reconstruction using Total Variation minimization (TV).

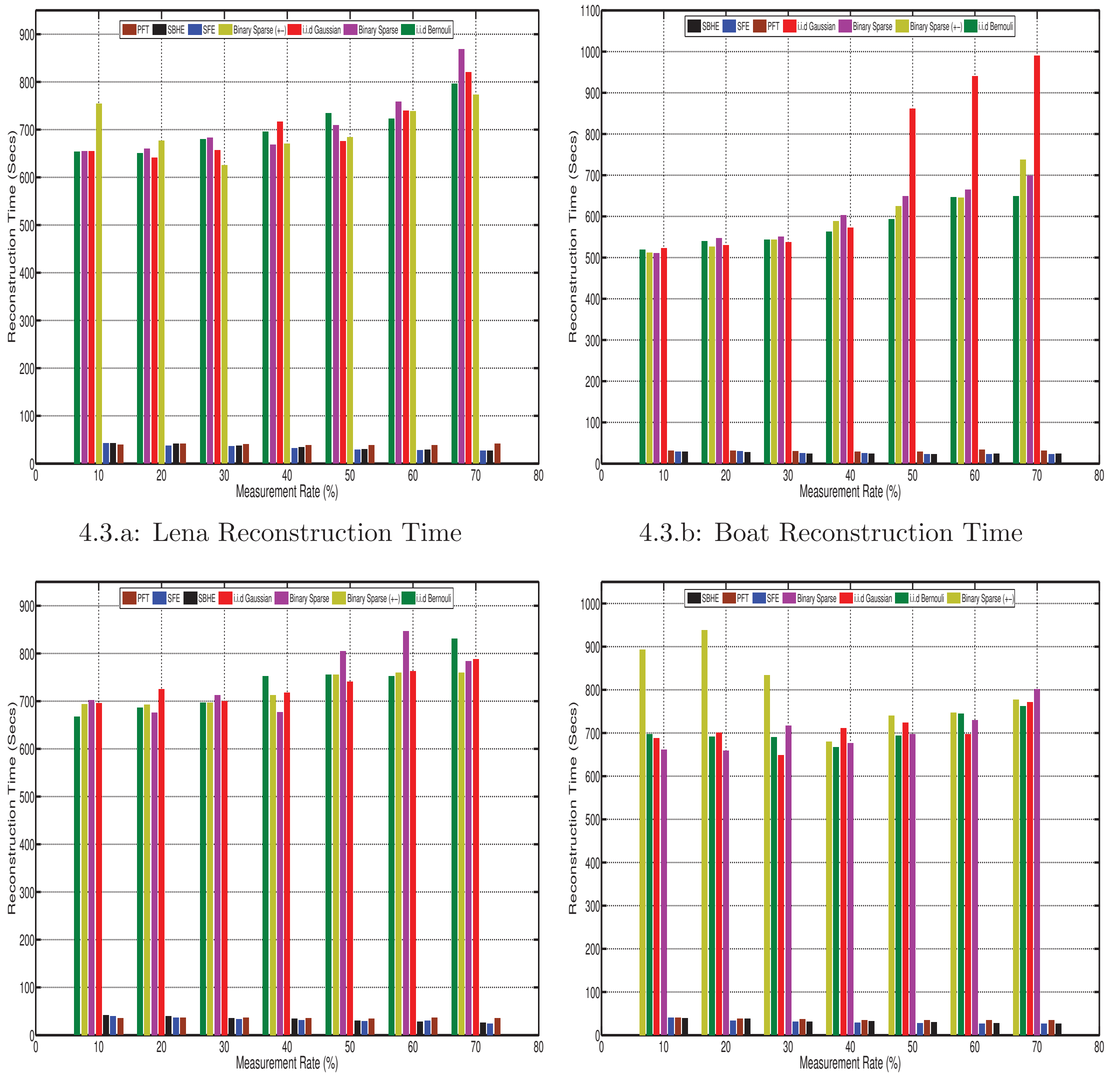
All experiments run on an Intel Pentium 4 3GHz system, running Windows 7 and MATLAB 2011. The sensing matrices that are compared include i.i.d. Gaussian, i.i.d. Bernoulli, Partial Fourier Transform (PFT), Scrambled Fourier Ensemble (SFE), Scrambled Block Hadamard Ensemble (SBHE), Binary Sparse and Binary Sparse with ±1. An image $x$ is represented as an $N \times 1$ column vector where $N$ is the total number of pixels in the image. If $\Phi$ is the sensing matrix, then CS measurements $y$ is obtained by $y = \Phi x$. The corresponding measurement rate is given by $M_r/N$. The measurement rate $M_r$ is in the range of 10% to 70%.

### Acquisition Time

In imaging applications, fast acquisition of CS measurements is essential. Figure 4.2 shows the acquisition time for different test images. The acquisition time for transform based matrices, i.e. SFE,PFT and SBHE, is less than fraction of a second for all test images. This is due to fact that fast transform methods are available for this type of matrices. For random Gaussian/Bernoulli matrices, the acquisition times at low measurement rates are similarly low as only a small number of random numbers are generated. For higher measurement rates, the acquisition time for random matrices increases due to increase in the size of the measurement matrix. Interestingly, for binary sparse matrices, the acquisition time is quite high in comparison to other sensing matrices. At low measurement rate, 10% it is quite high for binary sparse matrix having ±1. The reason for such a high acquisition time is that, after generating the binary sparse matrix, it needs to be checked for duplicate columns. This experiment suggests that instead of using a random or sparse matrix, a transform based sensing matrix provides less time in acquiring CS samples.

### Reconstruction Time and Quality

The next experiment evaluates the reconstruction performance for the sensing matrices in terms of reconstruction time and reconstruction quality. We used Total Variation (TV) minimization instead of $l_1$ minimization for reconstruction. In Section 2.1, the importance of coherence between the sensing and sparsifying matrices have been discussed. The various sensing matrices have different coherence with different sparsifying bases. For a fair comparison, the sparsifying transform is replaced by total variation (TV) minimization which uses a sparse approximation of image gradient. The benefit of using TV



4.3.a: Lena Reconstruction Time

4.3.b: Boat Reconstruction Time

4.3.c: Cameraman Reconstruction Time

4.3.d: MRI Reconstruction Time

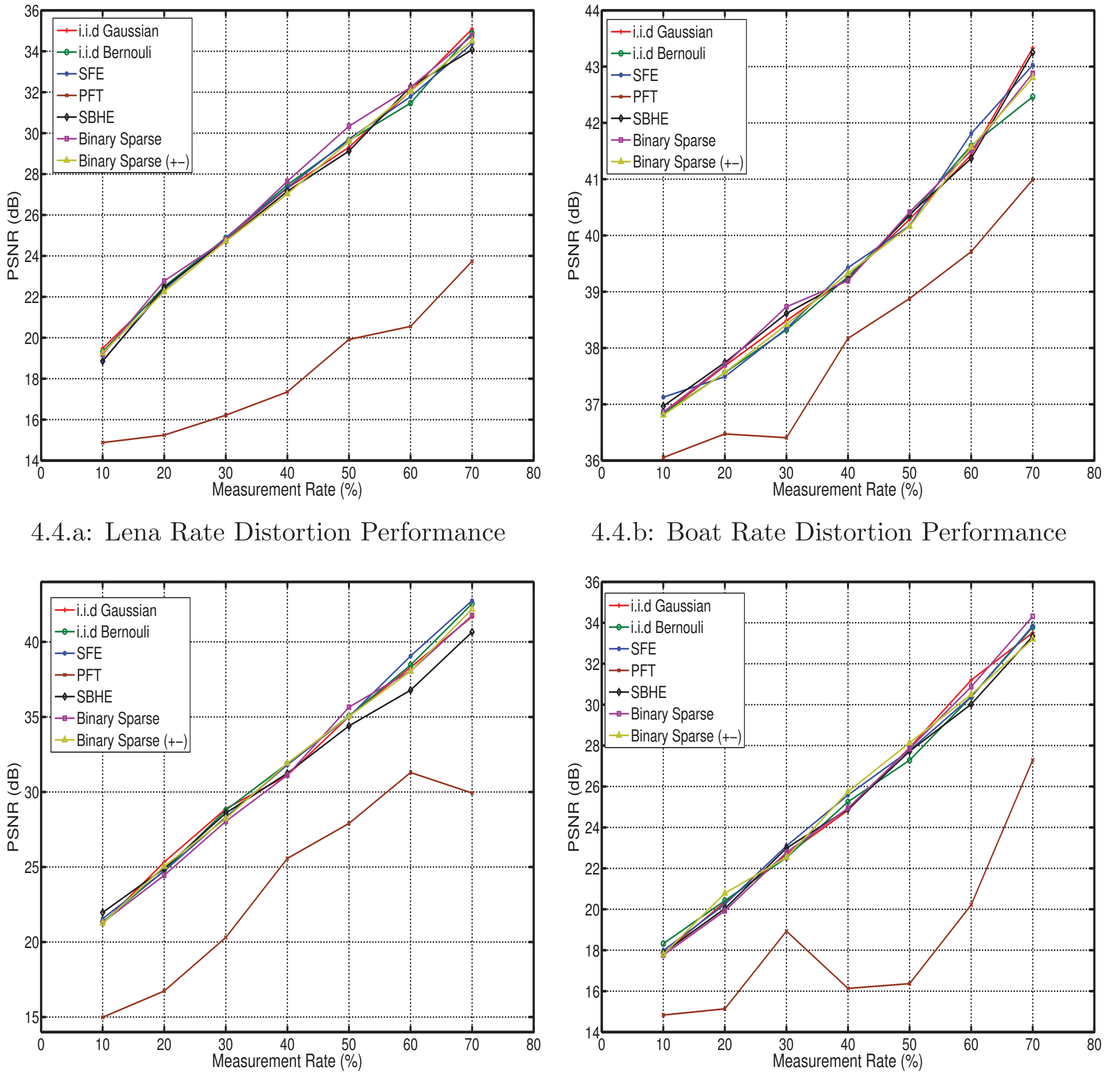
Figure 4.3: Sensing Matrix Reconstruction Time

norm for images is to avoid high frequency artefacts. The TV norm of an image is given by

$$||X||_{TV} = \sum_{i,j} \sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2} \tag{4.1}$$

where $x_{i,j}$ is the pixel value of image $X$ at row $i$ and column $j$.

Figure 4.3 shows the reconstruction times for different sensing matrices. The reconstruction times for transform based sensing matrices are less than 50 seconds for all test



4.4.a: Lena Rate Distortion Performance

4.4.b: Boat Rate Distortion Performance

4.4.c: Cameraman Rate Distortion Performance

4.4.d: MRI Rate Distortion Performance

Figure 4.4: Sensing Matrix Rate Distortion Performance

images under different measurement rates. Those for random matrices and binary sparse matrices are quite high in comparison. In particular, for the "Boat" image, at measurement rate $M_r \geq 50\%$ the random Gaussian matrix exhibits a much higher reconstruction time compared with the rest.

Figure 4.4 shows the quality of the reconstructed images in terms of PSNR at different measurement rates. The reconstruction performances are very similar for all sensing matrices tested except Partial Fourier (PFT). The performance of PFT is poor because it does not use scrambling which provides better results due to randomization process. This suggests that whether the sensing matrix is random (Gaussian/Bernouli), Sparse ($Binary/\pm 1$) or a scrambled transform based operator, the quality of the reconstructed images under TV norm is almost identical.

**Summary**

The Gaussian or Bernoulli matrices offer optimal performance and universality, but they are impractical for resource limited image encoding applications due to high memory requirements and high computational complexity. Scrambled Fourier Ensemble (SFE) and Partial Fourier transform have been used in many CS imaging literature due to their faster computation time. Partial Fourier does not perform well in terms of reconstruction performance. SFE performance is good in all performance metrics but its values are not binary. Binary Sparse matrices provides good reconstruction performance, comparable to random Gaussian and transform based operators but in our experiments we have found out that their acquisition time varies for different measurement rates. In terms of sensing matrix requirement in low complexity encoders, the implementation of random Gaussian/binary sparse matrices is challenging as they need to be stored explicitly and required at the decoder for reconstructing data. Scrambled Block Hadamard Ensemble (SBHE) not only provide similar performance as SFE but its hardware implementation is simpler due to its binary nature. It also has faster reconstruction time as transform based operators have fast matrix-vector product implementation than random Gaussian/Binary Sparse matrices.

## 4.2 Choice of Reconstruction Algorithms

In CS based compression systems, the decoder need to solve a convex optimization problem using either sparsity in a transform domain (Wavelet, DCT etc) with $l_1$ norm or image gradient with the $TV$ norm. These optimization problems are quite challenging to solve because both the $l_1$ and the $TV$ norms are non-smooth. For image processing applications the dimension of problem and its solution span are very large. The reconstruction problem has been formulated in three equivalent ways, namely, basis pursuit, basis pursuit denoising and "Lasso".

Basis Pursuit (BP) is a quadratically constrained $l_1$-minimization problem and it is given as:

$$\min_x \parallel x \parallel_{l_1} \text{ subject to } \parallel y - \theta x \parallel_{l_2} \leq \epsilon \tag{4.2}$$

where $y$ is the noisy CS measurements with noise $\epsilon$. This is a preferred CS reconstruction formulation as the estimate of noise $\epsilon$ may be known or can be computed.

A second approach in CS reconstruction is to formulate the BP problem as a second order cone program:

$$\min_x \frac{1}{2}||y - \theta x||_{l_2}^2 + \lambda||x||_{l_1} \tag{4.3}$$

This formulation is also known as Basis Pursuit Denoising (BPDN) [28]. It is tractable due to its bounded convex optimization nature. The term $\lambda||x||_{l_1}$, which is also known as regularization, can be interpreted as a maximum *a posteriori* estimate in a Bayesian setting. This makes BPDN very popular for signal and image processing applications.

The third formulation is known as the Least Absolute Shrinkage and Selection Operator (Lasso) [29]. It is the minimization of an $l_2$ norm subject to $l_1$ norm constraints:

$$\min_x ||y - \theta x||_{l_2} \text{ subject to } ||x||_{l_1} \leq \tau \tag{4.4}$$

There are a number of different algorithms developed to solve these three CS reconstruction problems. Linear programming (LP) techniques have been shown to be effective in solving such problems with high accuracy [10,11]. However, due to large solution space in image processing applications, more efficient methods are preferred. Some algorithms requires fewer iterations or less computation time per iteration and some work only on sparsity in the $l_1$ norm or the $TV$ norm. They can be broadly classified as greedy algorithms, gradient-based algorithms, and iterative shrinkage algorithms. We will evaluate their performance in Section 4.2.4 using the following criteria:

- Reconstruction Time: Due to large dimension of image data, an efficient algorithm which can provide solution in minimal running time is highly desirable.

- Number of Iterations: An algorithm which can found a solution in a minimum number of iterations will be very efficient.

- Sparsity Flexibility: CS reconstruction can be formulated differently as shown in Equations 4.2, 4.3 and 4.4. An algorithm which can provide solution to all three CS formulations is highly desirable. For example an algorithm which can support both $l_1$ and $TV$-norm implementation will be very useful for image and video applications.

### 4.2.1 Greedy Algorithms

Greedy algorithms are based on Matching Pursuit (MP) [30] and its variants such as Orthogonal Matching Pursuit (OMP) [31], Stagewise Orthogonal Matching Pursuit (StOMP) [32], and Compressive Sampling Matching Pursuit (CoSaMP) [33]. They are iterative algorithms that decomposes a signal into a linear expansion of functions that form a dictionary $\theta$. Greedy algorithms starts with an initial solution $\hat{x} = 0$ and then greedily chooses elements of $\hat{x}$ to assume non-zero magnitudes by iteratively processing residual errors between measurement vector $y$ and $\theta\hat{x}$. At each iteration, algorithm attempts to choose the best approximation for the current residual from the dictionary. We will briefly discuss two of these greedy algorithms.

#### Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) is the orthogonalized version of a Matching Pursuit (MP) algorithm. The idea is that at every iteration an element is picked from the dictionary that best approximates the residual. With OMP, instead of simply taking the scalar product of the residual and the new dictionary element, the original function is fitted to all the selected dictionary elements via a least squares or a projection of the function orthogonally onto all the selected dictionary atoms.

The algorithm starts with residual $r_0 = y$, i.e. the solution $\hat{x} = 0$. At every iteration $i > 0$, the column of $\theta$ that is most correlated with current residual $r_{i-1}$ is selected. It then solves the least square problem for the new signal estimate using only the dictionary atoms that have been previously selected. A new residual is computed using the most recent approximation. It continues until a predetermined stopping criteria or a fixed number of iterations is reached.

**Stagewise Orthogonal Matching Pursuit**

Stagewise Orthogonal Matching Pursuit (StOMP) is introduced in [32]. It is an improvement on the OMP algorithm. In contrast to OMP, which allows only one dictionary atom to be added in an iteration, StOMP allows multiple ones to be added in a single iteration. It is significantly faster than OMP. A sequence of approximations $\{x_0, x_1, ...\}$ are obtained by removing the detected structure from residual vectors $r_1, r_2, ....$

StOMP starts with initial solution at zero and initial residual equal to observation as for OMP. It applies matched filtering to the current residual with $\theta$ to obtain a vector of residual correlations. All the vectors above a threshold are selected using hard thresholding. Then least squares method is applied to find an approximation. The algorithm then updates the residual and check for a stopping condition. An advantage of using a thresholding approach is that it can produce good approximation of the original signal with only a small number of iterations.

### 4.2.2 Gradient based Algorithms

The gradient projection methods proposed in [34,35] recast Equation 4.3 as a constrained optimization problem by introducing additional variables. Gradient descent directions, which are generally easy to compute, are used at each iteration, and are then projected onto the constraint set so that each step is feasible. The projection involves only simple thresholding and can be done very quickly, which leads to fast computation at each iteration. We will briefly discuss two of the popular gradient based algorithm.

### Gradient Projection for Sparse Reconstruction (GPSR)

In [34], a GPSR algorithm is introduced to solve the standard $l_1$ minimization problem in the unconstrained quadratic form as in Equation 4.3. GPSR uses a penalty term involving the scaled $l_1$ norm of the signal which is added to a least-squares term, resulting in a problem that can be reformulated as a convex quadratic program with bounded constraints. It is a gradient projection solver which uses a special line search and termination technique to give faster solutions.

GPSR divides original problem into positive and negative parts by splitting $x$ as $x = u - v, u \geq 0, v \geq 0$. The problem is then converted to the following bounded constraint quadratic program (BCQP):

$$\begin{aligned} \min_{u,v} \frac{1}{2}||y - \theta(u-v)||_{l_2}^2 + \lambda 1_n^T u + \lambda 1_n^T v \\ \text{subject to } u \geq 0, v \geq 0 \end{aligned} \tag{4.5}$$

It can be written as a more standard BCQP:

$$\begin{aligned} \min_z c^T z + \frac{1}{2} z^T B z = F(z), \\ \text{subject to } z \geq 0 \end{aligned} \tag{4.6}$$

where

$$z = \begin{bmatrix} u \\ v \end{bmatrix}, b = \theta^T y, c = \lambda 1_{2n} + \begin{bmatrix} -b \\ b \end{bmatrix}$$

and

$$B = \begin{bmatrix} \theta^T A & -\theta^T A \\ -\theta^T A & \theta^T A \end{bmatrix}$$

Here $A$ is the measurement matrix and $F(z)$ is the change of variable. It is important to use a suitable step size so that an approximation can be reached with a minimum number of iterations. GPSR uses backtracking and Barzilai-Borwein methods [112] for determining the step size. The authors also provided a debiasing approach in which the computed solution is checked against the least-square objective. The algorithm also supports warm start in which it will be solved for a range of different values of $\lambda$. The MATLAB code for the algorithm is available online at [113]. One disadvantage of GPSR is that it does not work for $TV$ minimization.

### Nesterov's Algorithm (NESTA)

In [114], a specialized algorithm for solving CS reconstruction problem called NESTA is introduced. This algorithm is based on Nesterov's work on minimizing non-smooth functions [115]. It makes use of two ideas due to Nesterov. The first one is an accelerated convergence scheme for first-order methods, giving optimal convergence rates. The second one is a smoothing technique that replaces the non-smooth $l_1$ norm with a smooth version. NESTA uses a subtle averaging of sequences of iterates, which improves the convergence properties of standard gradient-descent algorithms. It can be used to solve both $l_1$ and $TV$ minimization.

The algorithm minimizes a function $f$ by iteratively estimating three sequences $x_k, y_k$ and $z_k$. Step 1 is to compute the gradient $\nabla f_\mu(x)$ using Nesterov's smooth approximation to the $l_1$ norm. At iteration $i$, this is given by

$$\nabla f_\mu(x)[i] = \begin{cases} \mu^{-1}, & \text{if } \|x[i]\| < \mu \\ sgn(x[i]), & \text{otherwise} \end{cases} \tag{4.7}$$

Step 2 computes the term $y_k$. $y_k$ evolve the iteration in the opposite direction of the gradient. If the sensing matrix is a fast transform and orthogonal, $y_k$ can be computed easily as it only depends on $x_k$ and its gradient $\nabla f_\mu(x_k)$. Step 3 computes the third sequence $z_k$ which keeps track of the previous gradient directions. It has been proved in [115] that this additional term helps to improve the convergence properties of the algorithm.

The major benefit of NESTA is that it can solve all three formulations of CS. It supports both $l_1$ and $TV$ norms. The MATLAB code of the algorithm is available from [116].

### 4.2.3 Iterative Shrinkage Thresholding Algorithms

If a signal is represented by transformation to a suitable basis, coefficients below some threshold value can usually be set to zero without affecting the quality of the signal significantly. This produces a sparser signal and thus is very much applicable to image and video compression. Iterative shrinkage thresholding algorithms transform the original optimization problem into a sequence of simpler problems which can be solved efficiently by shrinking or thresholding small values in the current estimate of the signal $x$ [36, 37].

**Two-Step Iterative Shrinkage Thresholding (TwIST)**

TwIST is first proposed in [37]. It is called "two-step" because each iteration of TwIST depends on two previous iterations. At each iteration $k+1$, the solution is updated to $x_{k+1}$ by

$$x_{k+1} = (1-\alpha)x_{k-1} + (\alpha-\beta)x_k + \beta\Psi_\lambda(x_k + \theta^T(y - \theta x_k)) \tag{4.8}$$

where $\Psi_\lambda$ is a denoising operation. The term $\theta^T(y - \theta x_k)$ is the gradient of the term $\frac{1}{2}||y-\theta x_k||_2^2$. $\alpha, \beta$ are the parameters of the TwIST algorithm and different criteria for selecting them is defined in [37].

**Sparse Reconstruction by Seperable Approximation (SpaRSA)**

SpaRSA is proposed in [36]. It minimizes the function $g(x) = f(x) + \lambda c(x)$ which is composed of a smooth term $f$ and a separable non-smooth term $c$. At every step, a sub-problem in the form of Equation 4.3 is solved. The $(k+1)^{th}$ iterate is obtained by solving:

$$x_{k+1} \in \arg\min_z \frac{1}{2}||z - u_k||_2^2 + \frac{\lambda}{\alpha_k}c(z) \tag{4.9}$$

where

$$u_k = x_k - \frac{1}{\alpha_k}\nabla f(x_k) \tag{4.10}$$

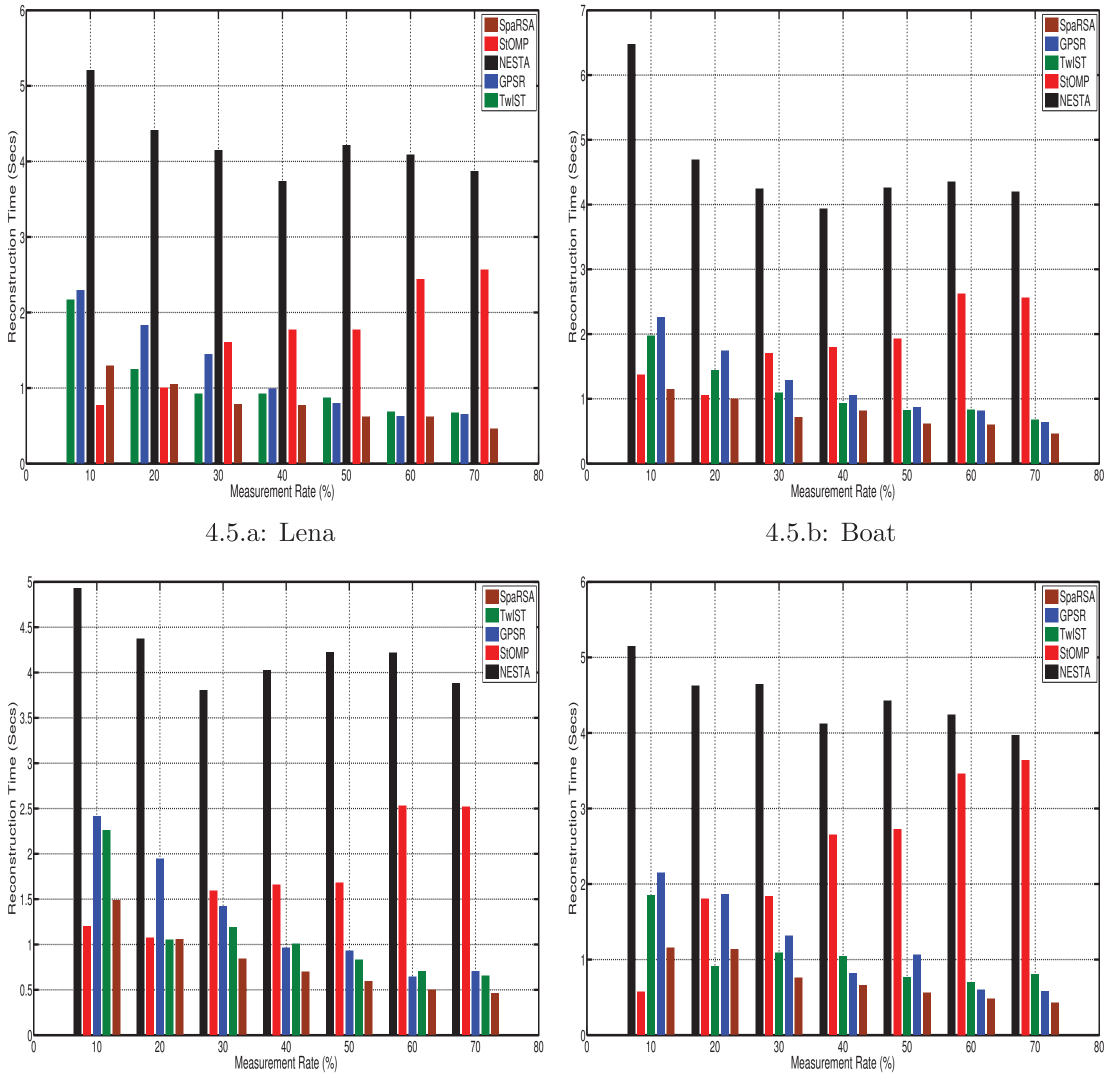
The authors of [36] uses a separable form for $c$. Individual components are separated into a sum of functions of its argument as $c(x) = \sum_{i=1}^{n} c_i(x_i)$, which is an $l_1$ regularizeer form. The algorithm starts with the initial solution $x_0 = 0$. At each subsequent iteration, it chooses the value of $\alpha_k$ from the interval $[\alpha_{min}; \alpha_{max}]$. Then problem 4.9 is solved to obtain the solution $x_{k+1}$ of the sub-problem. It then updates $\alpha_k$ and the process is repeated until an acceptable solution is obtained. After finding an acceptable solution to the sub-problem, the algorithm updates the iteration and check for the stopping condition. If the stopping condition is not satisfied, it will proceed with the next iteration. SpaRSA supports warm starting and debiasing.

### 4.2.4 Experimental Results

The same images as in Section 4.1.1 – “Lena”, “Boat”,”Cameraman” and “MRI” are used in the evaluation of the reconstruction algorithms here. The dimension of each image is $256 \times 256$ . All experiments run on Intel Pentium 4 3GHz system, running Windows 7 and MATLAB 2011. $l_1$ norm is used in our analysis as not all algorithms support the use of the $TV$ norm. Block Hadamard Transform is used as the sensing matrix with a block size of $32 \times 32$ and Wavelet transform is used as sparsifying transform. The measurement rate $M_r$ is in the range of 10% to 70%. The algorithms – StOMP, GPSR, NESTA, TwIST and SpaRSA are compared. The regularization parameter $\lambda$ used for all the algorithms is fixed and it is given in [34] as

$$\lambda = \frac{F(y)}{16/sqrtM_r} \tag{4.11}$$

where $F(y)$ is the Frobenius norm of measurement vector $y$. The stopping criterion uses a threshold to stop the algorithm when the change in the objective function falls below a threshold. It is also fixed same for all testing algorithms as $10^{-3}$.



4.5.a: Lena

4.5.b: Boat

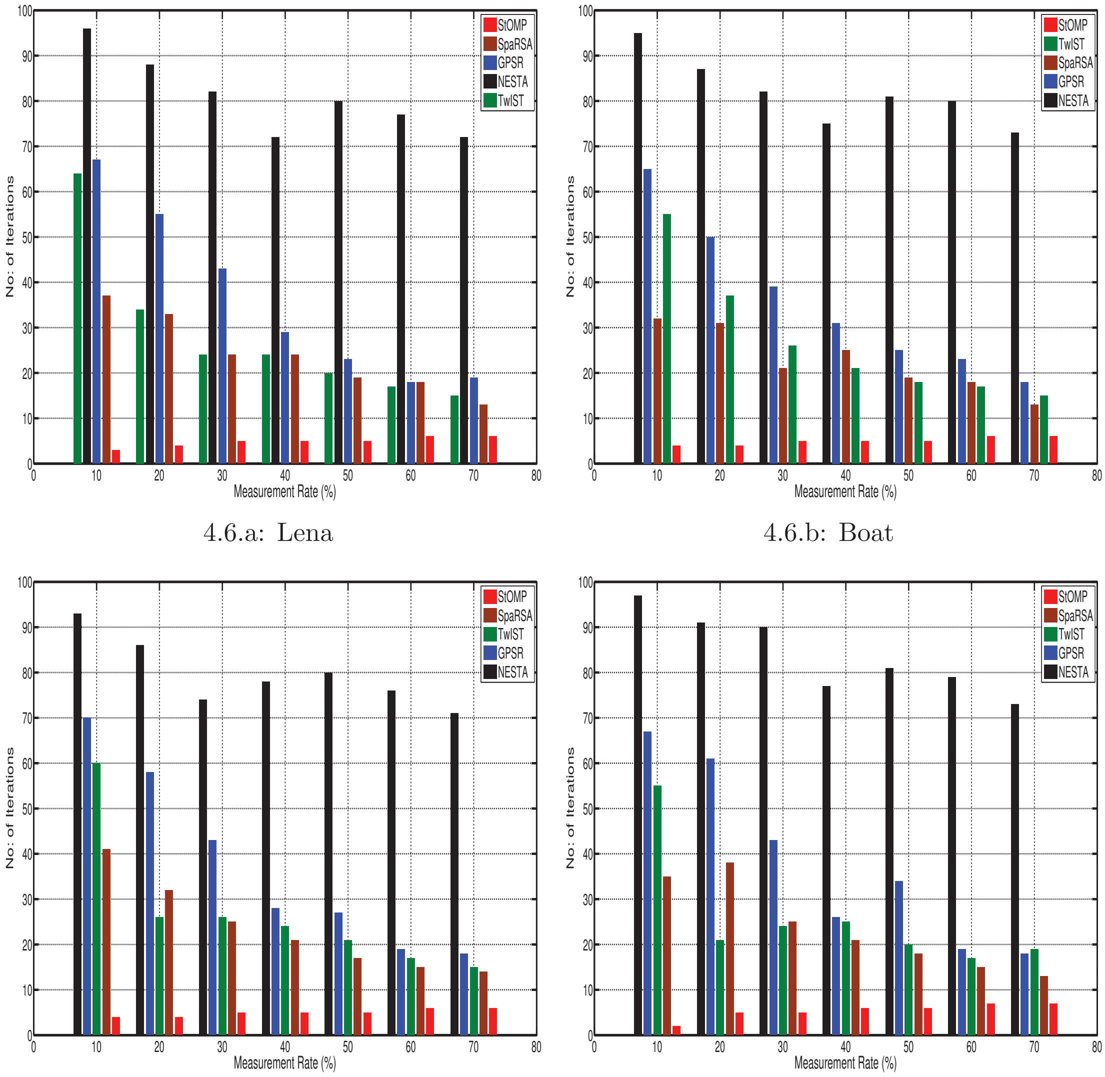
4.5.c: Cameraman

4.5.d: MRI

Figure 4.5: Reconstruction Times for CS Reconstruction Algorithms

### Reconstruction Time

Fast reconstruction time is a desirable feature of any CS reconstruction algorithm for image and video applications due to large amount of data involved. Figure 4.5 shows a comparison of the reconstruction times for four test images at various measurement rates. NESTA is the slowest among the algorithms compared at all measurement rates even though previous literature suggests that it is very efficient [114]. The performance of the other gradient-based algorithm, GPSR, is much better. GPSR is able to obtain



4.6.a: Lena

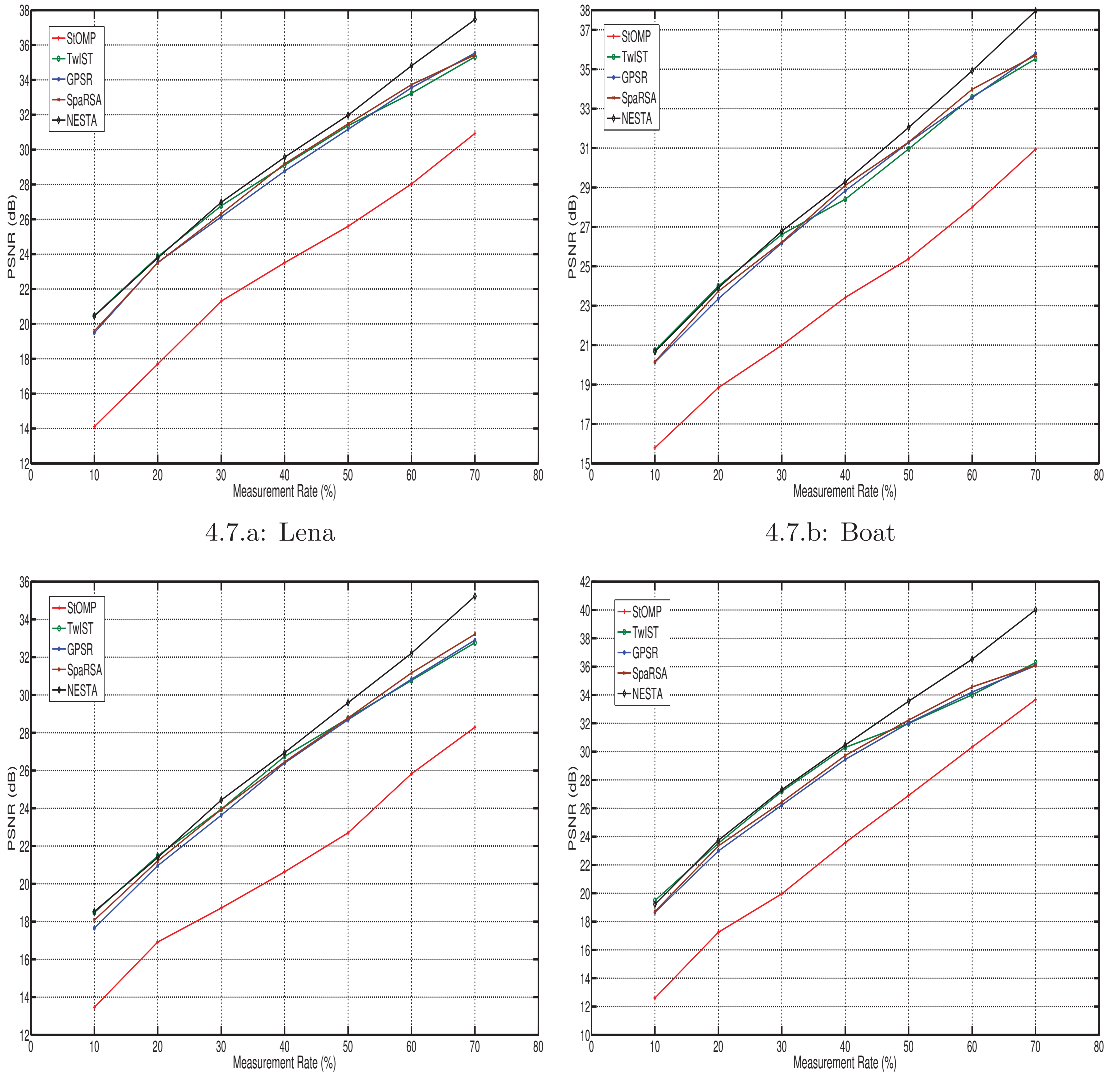
4.6.b: Boat

4.6.c: Cameraman

4.6.d: MRI

Figure 4.6: Number of Iterations Required for CS Reconstruction

the solution faster when more measurements are available. On the other hand, the reconstruction time for greedy algorithm StOMP increases with measurement rate. This is understandable as it depends on the inner product of the residual with the dictionary. As the dictionary size increases, more time is needed to compute this product. The performances of Iterative algorithms TwIST and SpaRSA are very good at all measurement rates. At higher measurement rates, they are comparable with or better than GPSR. They generally require more time at low measurement rates but overall SpaRSA consistently performs better than TwIST. Next, the number of iterations each algorithm took



4.7.a: Lena

4.7.b: Boat

4.7.c: Cameraman

4.7.d: MRI

Figure 4.7: Rate-Distortion Performance of CS Reconstruction Algorithms

to find the solution is compared in Figure 4.6. Among the algorithms considered, greedy algorithm StOMP requires the minimum number of iterations to solve the optimization problem. SpaRSA perform well at lower measurement rates and its performance is comparable with TwIST at higher measurement rates. At higher measurement rates, GPSR is comparable to TwIST and SpaRSA. Again, NESTA is the worst performer. The much higher number of iterations is reflected in the longer reconstruction time shown earlier.

### Reconstruction Quality

The quality of the reconstruction images in terms of PSNR is compared in Figure 4.7. Obviously, the quality improves as the measurement rate increases. All algorithms perform similarly except StOMP. The performance of StOMP is much poorer than other algorithms. NESTA performs marginally better than TwIST,SpaRSA and GPSR but the difference is not substantial. These results indicate that gradient projection algorithms and iterative algorithms perform similarly in imaging applications.

### Summary

Overall, considering reconstruction quality, number of iterations and reconstruction time, SpaRSA performs really well. SpaRSA would be the first choice as CS reconstruction algorithm for image coding. In terms of reconstruction quality, NESTA is excellent at all measurement rates but its high reconstruction time and number of iterations makes it unsuitable for imaging applications. TwIST is comparable to SpaRSA algorithm in all aspects. Another important factor in imaging application is the ability to use of $TV$ norm and algorithms capable of supporting this norm are highly desirable. The performance of GPSR is comparable to iterative algorithms. However, since it cannot be used for $TV$ norm minimization, it is not as useful.

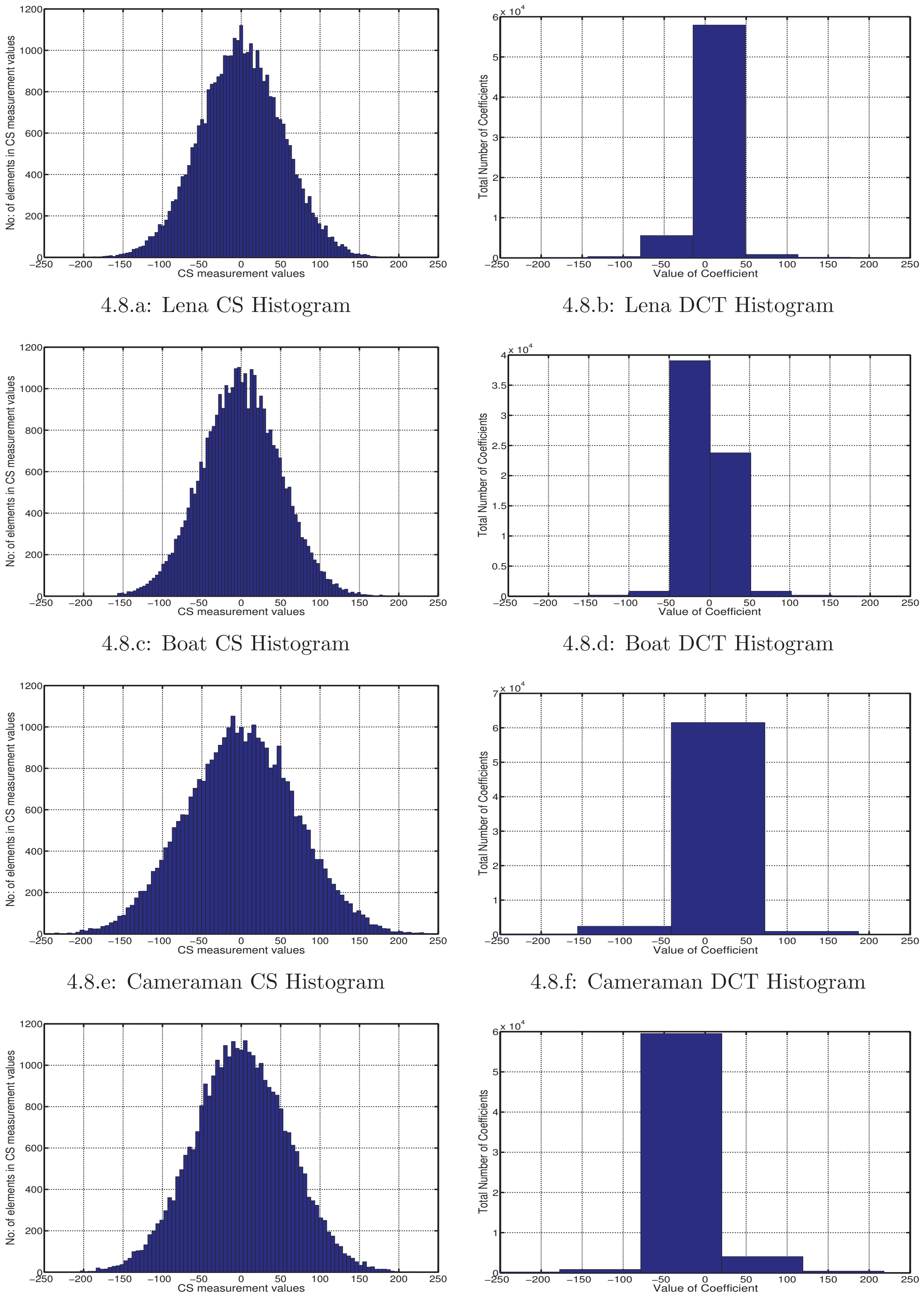
## 4.3 Design of Quantizer

The quantizer is very important part of the lossy encoding process. An optimal quantizer should be tailored to the signal concerned and minimize the amount of distortion in the reconstructed signal [117]. However, for practical reasons, fixed quantizers that are sub-optimal are always used. In image compression standards, pre-defined quantization matrices are used. For example, with the JPEG standard [1], the DC and the lower frequency Discrete Cosine Transform coefficients are finely quantized while the higher frequency coefficients are coarsely quantized [2]. This is based on the fact that the human visual system is less sensitive to errors in the higher frequencies compared to the lower frequencies. Also, the values of the DCT coefficients tend to be larger at the lower end of the spectrum. Uniform quantization schemes are also used [3].

While these quantizers are designed based on the knowledge of the human visual system as well as the distribution of linear transform coefficients for a variety of images, the CS measurement process is very different from the traditional approaches. The distribution of CS coefficients is directly related to the measurement matrix used. Due to the need for satisfying the restricted isometry property as described in Chapter 2, the i.i.d. Gaussian matrix is often used. In section 4.1, we have discussed different sensing matrices and evaluated their performance. Our analysis suggests that using a Scrambled Block Hadamard Ensemble (SBHE) is not only hardware friendly (due to its binary values) but also provides faster reconstruction. Hence we shall design the quantizer based on the SBHE as sensing matrix.

### 4.3.1 Distribution of CS Measurements

The same four images used in Sections 4.1.1 and 4.2.4 are again used here to analyze the distribution of the values of CS measurements compared with DCT. Figure 4.8 shows the histograms of the DCT and the CS coefficients of these images. $x-axis$ defines the coefficient values in DCT and CS measurements while $y-axis$ defines total number of coefficient having the same values. It can be observed that the CS measurement process spreads the energy of the coefficients. In contrast to DCT which concentrates the energy to a small number of coefficients. The DCT coefficient values are concentrated at the lower end of the frequency spectrum. The majority of the DCT coefficients are zero or close to zero. On the other hand, the CS measurement values follow a more or less normal (Gaussian) distribution. This indicates that the quantizer for CS should be different from those used in the current image coding standards.



4.8.a: Lena CS Histogram

4.8.b: Lena DCT Histogram

4.8.c: Boat CS Histogram

4.8.d: Boat DCT Histogram

4.8.e: Cameraman CS Histogram

4.8.f: Cameraman DCT Histogram

4.8.g: MRI CS Histogram

4.8.h: MRI DCT Histogram

Figure 4.8: CS Measurements and DCT Coefficients Histogram for Test Images

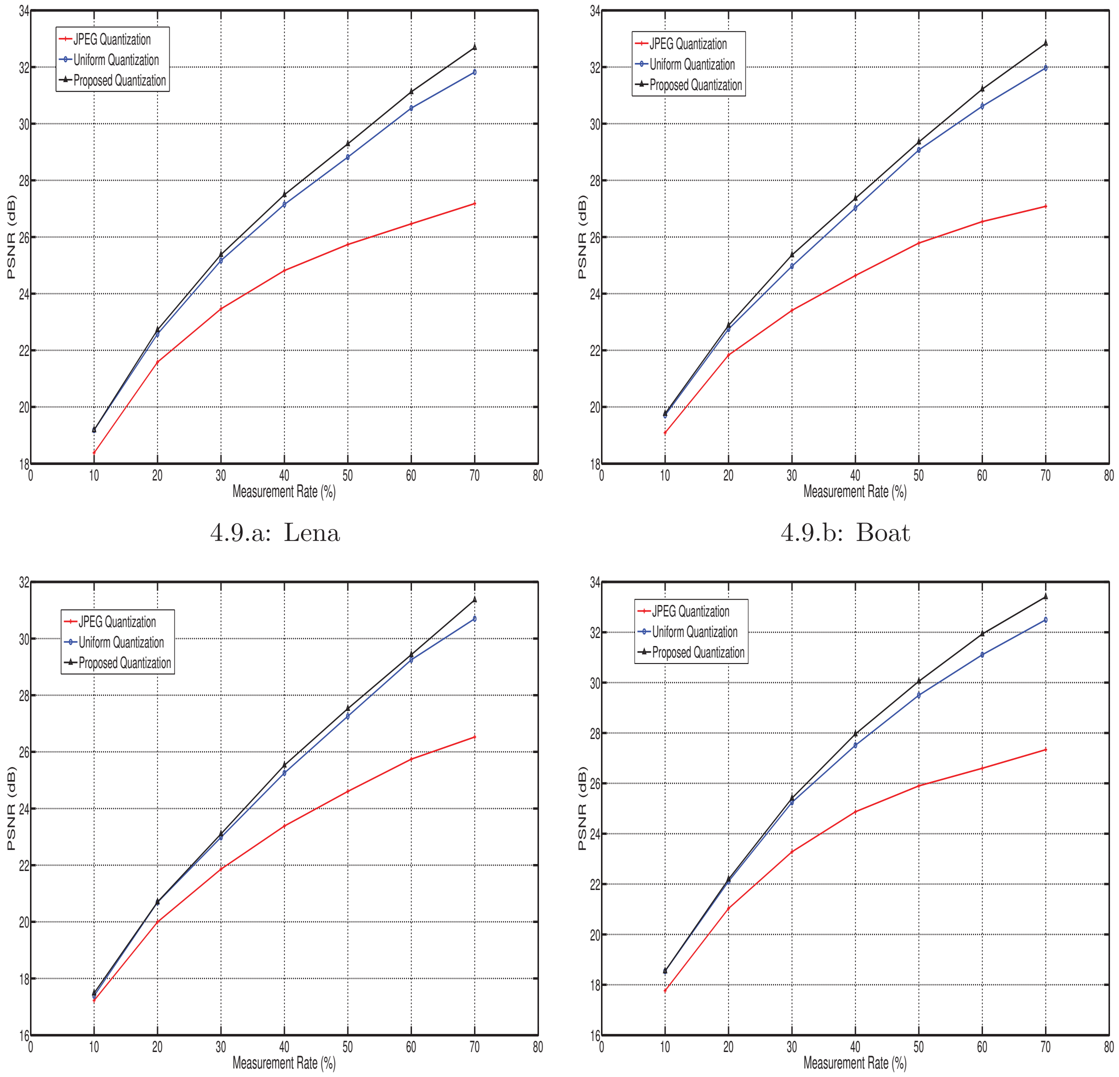
### 4.3.2 Proposed Quantization Scheme

Since the CS measurements are normally distributed, it is reasonable to quantize them using a quantizer that is also normally distributed. Thus the basis of the quantization matrix is a random matrix $R$ with elements that are Gaussian distributed with zero mean and unit variance. The average value of the mean and standard deviation of a number of similar type of images can be used to generate the quantization matrix. $R$ is then modified by this mean $\mu$ and standard deviation $\sigma$, giving us the quantization matrix $Q$ given by

$$Q = R * \sigma + \mu \tag{4.12}$$

Equation 4.12 is then used to generate the quantization matrix to quantize CS measurements. The $Q$ could be generated for any dimension as per image dimension requirements. As $Q$ is generated with Gaussian distribution, it is possible that there will be few matrix values close to zero or too high. To obtain appropriate weights, values that are close to zero are changed to a fixed non-zero constant such as 16 (which is used in the experiments described below) as done in uniform quantization [3]. Values that are much too high (e.g ¿ 100) are replaced by their square root.

The performance of this quantization scheme is evaluated as part of the model as shown in Figure 4.1. CS measurements are obtained by a SBHE sensing matrix known as the structurally random matrix [118]. The sparsifying transform is the Daubechies 9/7 wavelet transform. The images have dimension of 256 × 256 pixels. GPSR [34] is used for reconstruction. The proposed Gaussian quantization scheme is compared with the JPEG quantization matrix as defined in the JPEG standard [1], and a uniform quantization matrix [3]. In our experiments, only the mean and standard deviation of the "Lena" image is used to obtain the Gaussian quantizer. Figure 4.9 shows the PSNR performance at different measurements rates for four test images. It shows that the proposed Gaussian quantization method produces better results than both the other two quantization schemes for all measurement rates. It should be pointed out that even though



4.9.a: Lena

4.9.b: Boat

4.9.c: Cameraman

4.9.d: MRI

Figure 4.9: Quantization Matrix Rate-Distortion Performance

the Gaussian quantization matrix is generated based on the statistics of a one particular image, it works well for other images with different statistics. It seems to suggest that the quantizer is quite robust and therefore a fixed quantization matrix can be used. This has significant practical implications. In order to confirm this, the same quantization matrix is applied to a number of other images obtained from USC-SIPI Image Database [119]. The results are summarized in Table 4.1 for a measurement rate of 50%. They show that a fixed Gaussian quantization matrix works well in all cases. Figure 4.10 shows the visual reconstruction quality of “Lena” image at 50% measurement rate. It shows that

Table 4.1: Performance Comparison of Quantization Schemes, in PSNR(dB)

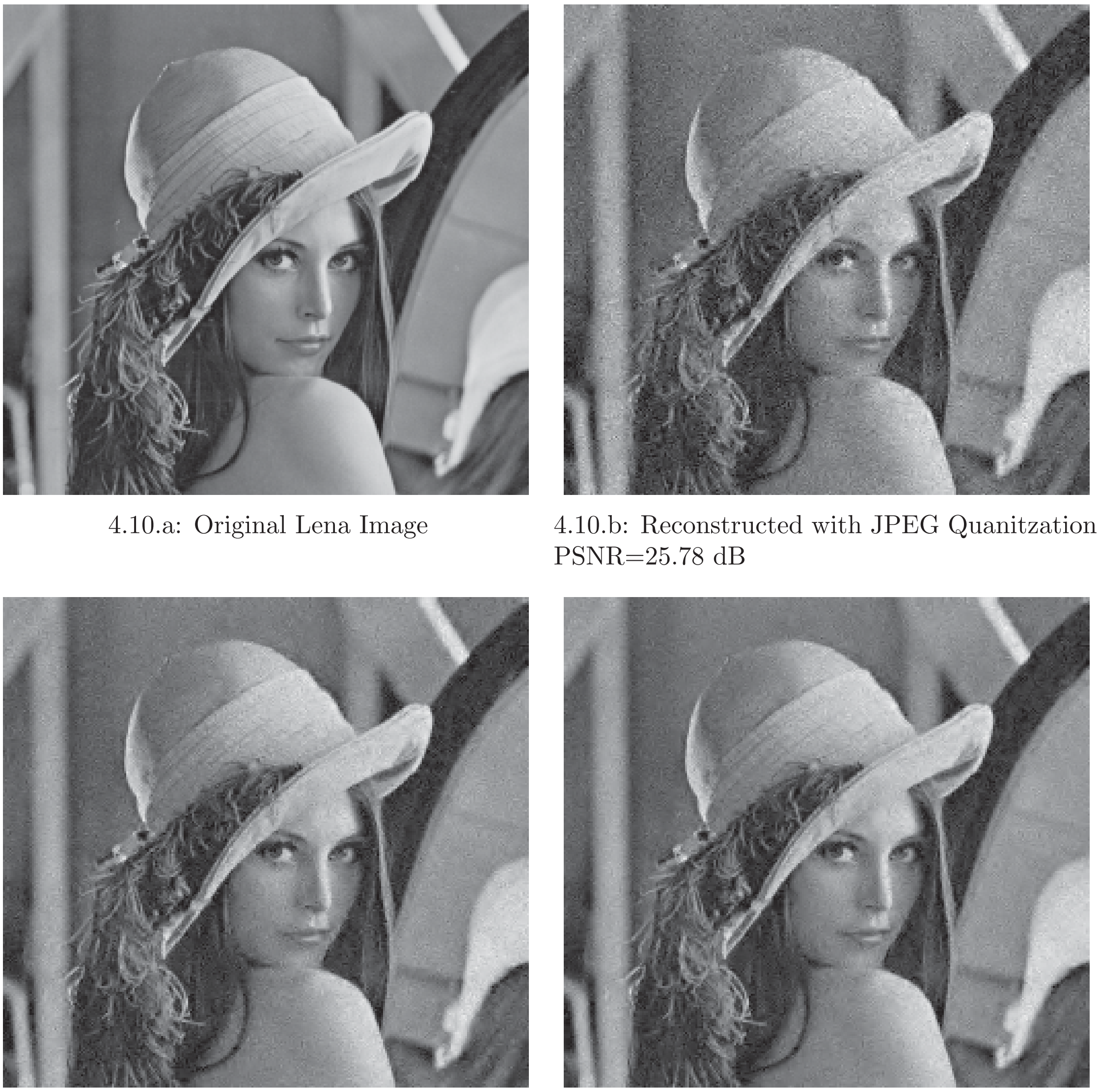
| Image | JPEG | Uniform | Gaussian |
|---|---|---|---|
| Lena | 25.77 | 28.81 | 29.57 |
| Boat | 25.85 | 28.83 | 29.51 |
| Cameraman | 24.76 | 27.15 | 27.72 |
| MRI | 25.72 | 29.39 | 30.38 |
| Barbara | 25.29 | 27.83 | 28.54 |
| Peppers | 25.66 | 29.02 | 29.55 |
| Goldhill | 25.77 | 28.64 | 29.22 |
| Mandrill | 23.50 | 24.69 | 24.82 |

our proposed quantization scheme produces better results visually than using uniform and JPEG quantization. In particular, the visual quality is particularly poor for JPEG quantization.

These results showed that a quantization matrix for CS coefficients can be designed using random Gaussian distribution. They open up a way to design practical CS codecs with quantization incorporated.

## 4.4 Summary

In this chapter, we studied the effects of the sensing matrix, the reconstruction algorithm, and the quantization matrix for image compression that is based *only* on CS in the way that is shown in Figure 4.1. Scrambled Block Hadamard Ensemble matrices have been found to be the best choice as sensing matrices in terms of structural properties, reconstruction time and reconstruction performance under different measurement rates. As for reconstruction algorithms, empirical results shows that for most of the them yields almost similar reconstruction quality. They only differ in reconstruction time and the number of iterations. GPSR is a good choice when $l_1$ norm is used.



4.10.a: Original Lena Image

4.10.b: Reconstructed with JPEG Quanitzation, PSNR=25.78 dB

4.10.c: Reconstructed with Unifrom Quanitzation, PSNR=28.82 dB

4.10.d: Reconstructed with Proposed Quanitzation, PSNR=29.57 dB

Figure 4.10: Reconstruction visual quality for Lena

A new quantization scheme has been proposed. The quantization matrix is a scaled random Gaussian matrix. It has been shown to perform better than JPEG and uniform quantization. Furthermore, it has been shown that such a quantizer is robust with respect to the statistics of individual images. Thus a fixed quantizer can be applied to different images. This is important for practical image codecs.

# Chapter 5

# Distributed Inter-frame Video Compressed Sensing

The encoding of video data in conventional video compression standards is a computationally demanding process mainly because it involves motion estimation to attain higher compression rates [3]. Decoding these compressed videos, on the other hand, is much simpler. For modern applications where video acquisition is performed by resource limited devices such as wireless sensors and decoding is performed by relatively resource rich computers, a new approach to video encoding and decoding is needed. It basically requires a low-power, low-complexity encoder while the computational burden is shifted from the encoder to the decoder.

Research in this direction has been developed along the lines of Distributed Video Coding (DVC) [9]. DVC is an application of distributed source coding, pioneered by Slepian and Wolf [65] for lossless coding and also Wyner and Ziv [66] for lossy coding. Two or more correlated data sources can be encoded independently and yet achieve the same compression rate as a single optimal encoder with all correlated data as input. This is achieved by exploiting the correlation between these data sources when they are jointly decoded. When applied to video coding, consecutive video frames could be treated as these correlated data sources and be encoded independently. This implies that no motion estimation need to be performed at the encoder, significantly simplifying the encoding process.

A number of DVC schemes that makes use of CS has recently been proposed [13, 14, 17, 72]. However, they either require a conventional video codec or a feedback channel from the decoder for effective operation, thus increasing the complexity of the codecs as discussed in Chapter 3.

In this chapter, a distributed Compressed Video Sensing (DCVS) codec is proposed. It is a DVC that only makes use of CS at the encoder. It also does not require a feedback channel from the decoder. A simple and effective side information generation scheme is incorporated in the decoder which exploits the correlation between CS measurements of nearby frames. This technique is based on the fact that CS measurements between video frames are highly correlated. It is much simpler than other schemes found in the literature and yet effective without putting extra complexity on the decoder. Side information is generated directly from CS measurements of the key frames. Thus it does not depend on the decoded key frames, unlike other DVC techniques. The performance of this DCVS codec is evaluated with several different types of video sequences. It is compared with that of other DCVS codecs, distributed video codecs and traditional video codecs.

## 5.1 Proposed DCVS Codec

Figure 5.1 shows a block diagram of the proposed distributed compressed video sensing (DCVS) codec. At the encoder, video frames are grouped into group of pictures (GOP) consisting of a key frame and a number of non-key frames, which are also called Wyner-Ziv (WZ) frames. Both the key frames and WZ frames are encoded as CS measurements using an appropriate sensing matrix. No traditional video encoding is involved. Typically, the measurement rate of non-key frames are much lower than that for key frames. The CS measurements are then quantized for transmission or storage. At the decoder, the key frames are reconstructed using its own CS measurements. WZ frames are reconstructed with the help of side information since their encoding rates are much lower. Side information is generated through a process involving a dictionary and the CS measurements of the current WZ frame. This side information is used to improve the reconstruction

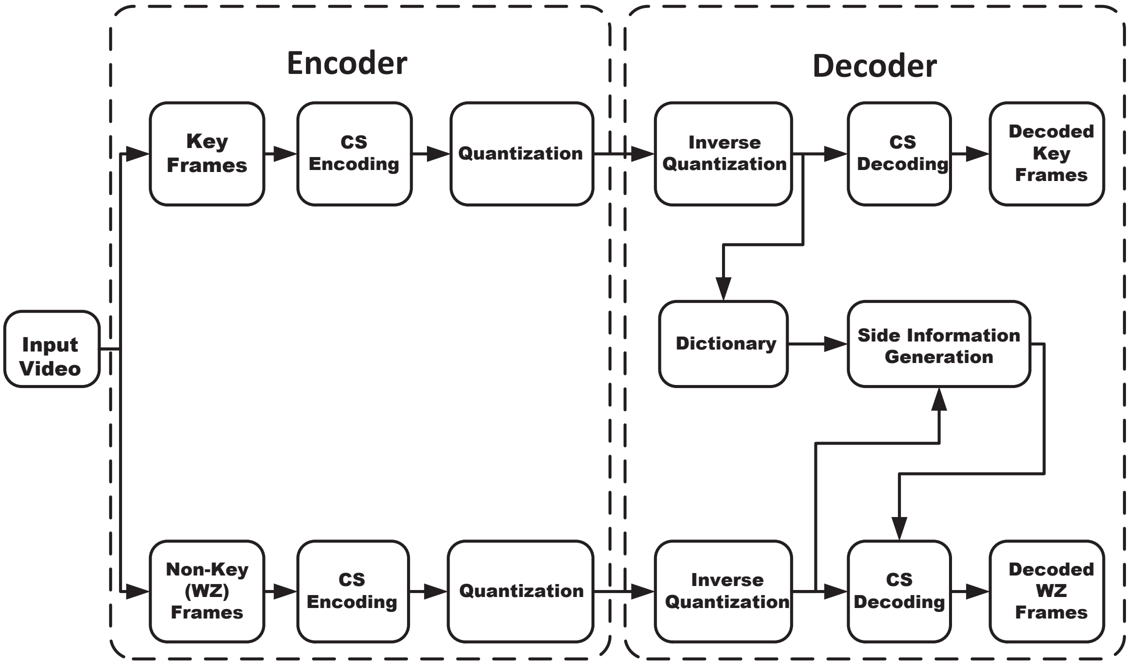


Figure 5.1: Proposed Video Codec

performance of WZ frames. Details of the encoding and decoding processes are described below.

### 5.1.1 Encoder

The encoding process is very simple. It only involves CS encoding and quantization. A video sequence is first broken up into a sequence of group of pictures. Each GOP consists of a key frame followed by some non-key WZ frames. Both key and non-key frames are encoded in a similar way using CS. Given the frame size of each frame as $w \times h$, then data $x$ of each frame represented as an $N \times 1$ column vector where $N$ is the total number of pixels in the frame ($N = w \times h$) and a $M_r \times N$ sensing matrix $\Phi$, the CS measurements $y$ is obtained by $y = \Phi x$. Let the number of measurements for key and non-key frames be $M_k$ and $M_w$ respectively. Then the corresponding measurement rates are given by $M_k/N$ and $M_w/N$. Key frames are encoded with a higher measurement rate than WZ frames such that $M_w < M_k < N$.

The measurements $y$ are then quantized by a Gaussian quantization scheme. This choice is based on the results presented in Section 4.3.2. Conventionally, different quantization matrices are used for intra-frame and inter-frame coding. For MPEG, the DC and the lower frequency Discrete Cosine Transform (DCT) coefficients are finely quantized while the higher frequency coefficients are coarsely quantized [2]. This design is based on the fact that the human visual system is less sensitive to errors in higher frequencies than it is for lower frequencies. Also, the values of the DCT coefficients tend to be larger at the lower end of the spectrum. For the H.264 baseline, main and extended profiles, the quantization matrix gives equal weight to all coefficients and uses a uniform quantization scheme [3]. The CS measurement process is very different from orthogonal transforms such as the DCT. The distribution of CS coefficients is directly related to the sensing matrix used. Consequently, both uniform quantization and the standard quantization matrices in MPEG do not perform well for compress-sensed images and videos. As shown in Section 4.3.2, Gaussian quantization performs better than both uniform and JPEG quantization for CS measurements.

### 5.1.2 Decoder

At the decoder, key frames are reconstructed from their own CS measurements. CS reconstruction can be formulated as an $l_1$-minimization problem given by

$$\min_{\alpha_k} \frac{1}{2}||y_k - \theta\alpha_k||_2^2 + \lambda||\alpha_k||_1 \tag{5.1}$$

where $y_k$ is an $M_k \times 1$ CS measurements of the key frame received at the decoder, $\theta = \Phi\Psi$ is the measurement matrix with $\Phi$ and $\Psi$ being the sensing and sparsifying matrices as described in Section 4.1 respectively. $\alpha_k \in R^N \times 1$ is the sparse coefficient vector which is solved by reconstruction algorithm. The key frame $\hat{x}_k$ is obtained by $\hat{x_k} = \Psi\hat{\alpha}_k$ where $\hat{\alpha}_k$ is the optimal solution for $\alpha_k$ in (5.1). The performance of a number of different CS reconstruction algorithms have been examined in Section 4.2.

Since WZ frames are encoded at a lower rate, the quality of the reconstructed frames can only be maintained with the aid of side information which is generated through a dictionary. The dictionary consists of inverse quantized CS measurements of the key frame. Side information is only useful if the CS measurements of the key and WZ frames exhibit sufficient correlation. Before discussing dictionary and side information generation, a correlation analysis of the CS measurements between video frames will first be presented.

### 5.1.3 Correlation Analysis of CS Measurements

In a video sequence, adjacent frames in same scene are highly correlated with each other. Therefore we postulate that the CS measurements of such adjacent frames are also highly correlated even though the CS measurement process is very different from linear transforms such as the DCT. DCT coefficients follows the Laplacian distribution [120]. On the other hand, the CS measurements follows a more or less normal (Gaussian) distribution. So the CS measurements can be modelled as random Gaussian sources. The dependence between two random quantities can be measured by Pearson's correlation coefficient [121]. It can be obtained by dividing the covariance of two variables by the product of their standard deviations. For two video frames with CS measurements $y_t$ and $y_{t+1}$, their correlation coefficient is given by

$$\rho = \frac{\sum\limits_{i=0}^{M-1} \left(y_{t,i} - \mu_{y_t}\right)\left(y_{t+1,i} - \mu_{y_{t+1}}\right)}{\sqrt{\sum\limits_{i=0}^{M-1} \left(y_{t,i} - \mu_{y_t}\right)^2 \left(y_{t+1,i} - \mu_{y_{t+1}}\right)^2}} \tag{5.2}$$

Here, $M$ is the number of measurements, $y_{t,i}$ and $y_{t+1,i}$ are the $i^{th}$ measurements, and $\mu_{yt}$ and $\mu_{y_{t+1}}$ are the mean values of $y_t$ and $y_{t+1}$ respectively.

Six standard QCIF video sequences, namely, "Foreman","News","Coastguard", "Students", "Container" and "Hall Monitor" available from [122] have been chosen to evaluate correlation between adjacent frames. CS measurements of the luminance data are obtained for each frame with a measurement rate of 50%. Each video sequence consists of 300 frames. They are divided into GOPs with a size of three. The first frame in a

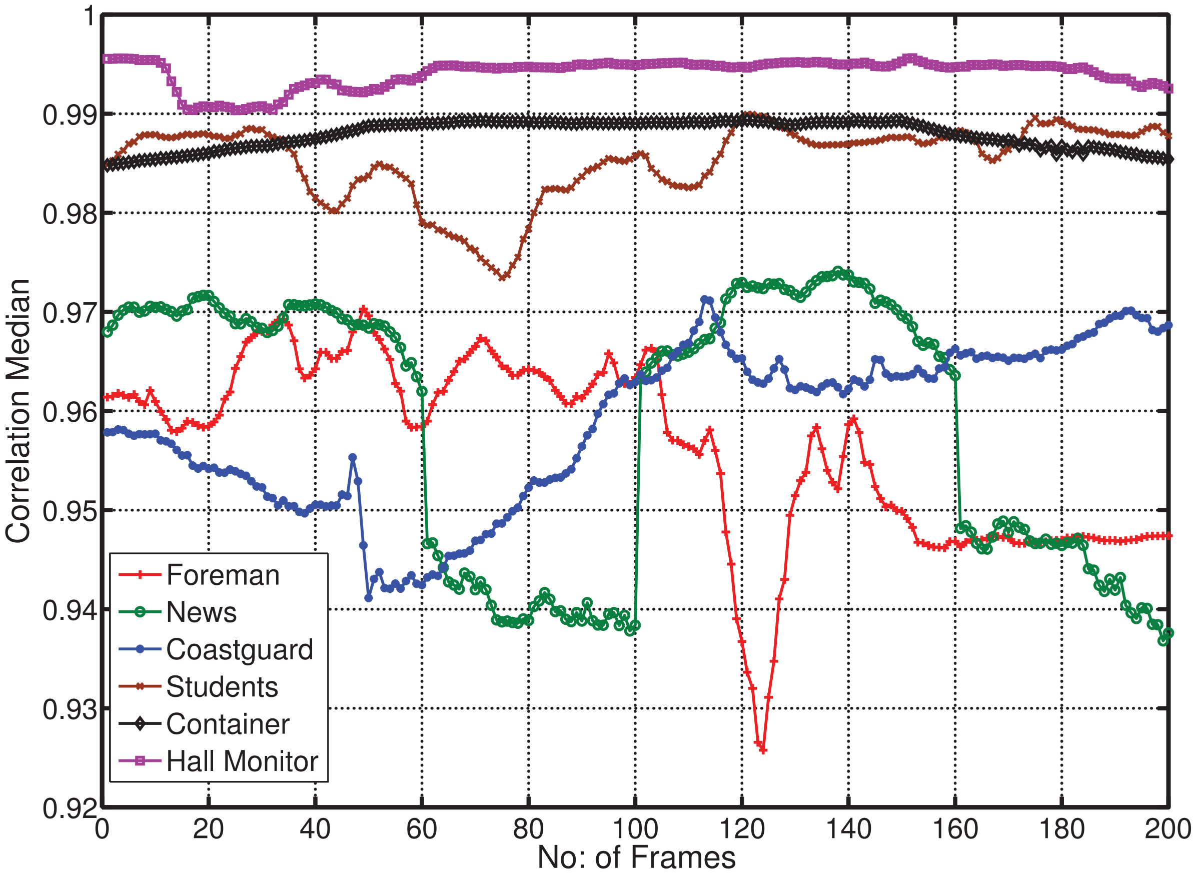
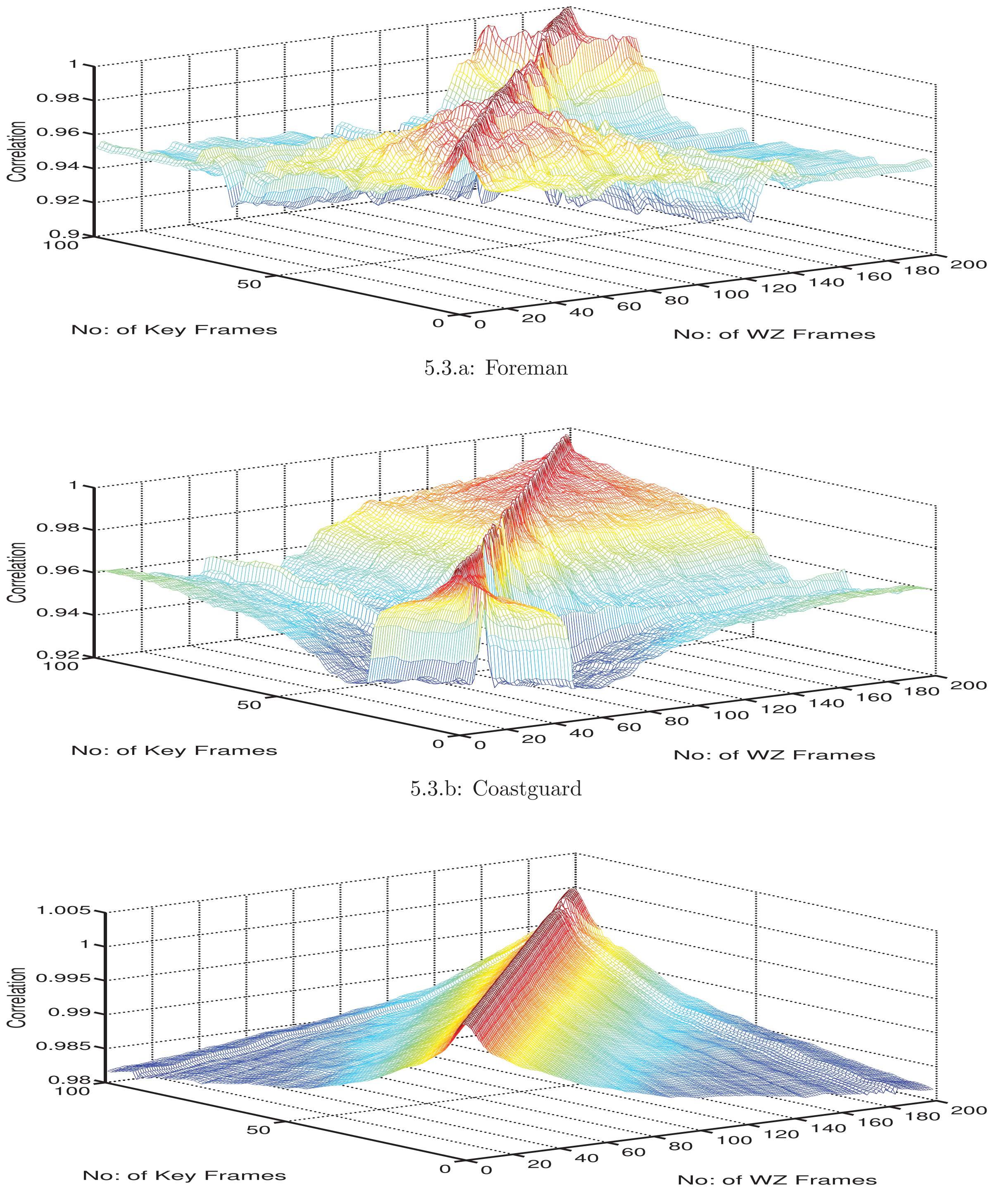


Figure 5.2: Correlation Analysis for CS Measurements

GOP is considered a key frame, followed by two non-key (WZ) frames. The correlation coefficient for each WZ frame with each key frame is computed and shown in Figure 5.2.

The correlation median is the median value of all the correlations for a WZ frame with each key frame. All the video frames show high correlation with a median correlation coefficient above 0.9. Even for those videos having moderate motion of objects and camera movements the correlation is very high.

Figure 5.3 shows the correlation between each WZ frame with all other key frames for the video sequences "Foreman", "Coastguard" and "Container". While the correlation between WZ frames with their corresponding key frames is high, the correlation with other key frames remain above 0.95. This is particularly so for low motion videos such as "Container". This high correlation will be exploited to generate side information to



5.3.a: Foreman

5.3.b: Coastguard

5.3.c: Container

Figure 5.3: Correlation of CS Measurements of WZ frames with Key frames

improve coding performance of WZ frames.

### 5.1.4 Correlation and Mean Square Error

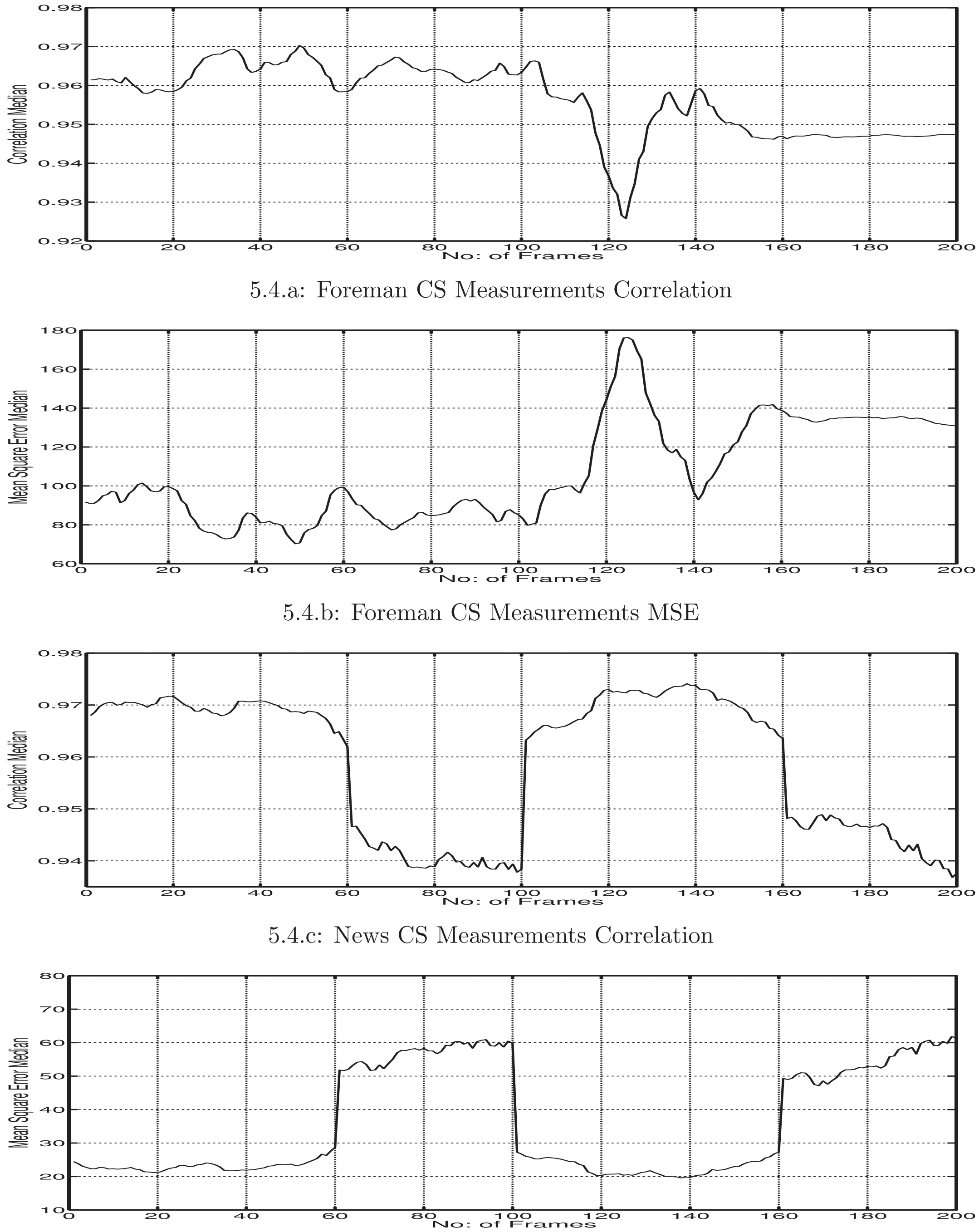
The Pearson's correlation coefficient is a good measure of linear relationships. Another important measure of the relationship between quantities is the Mean Square Error (MSE). In this case it is the cumulative squared error between CS measurements of adjacent frames. For two video frames with CS measurements $y_t$ and $y_{t+1}$, their MSE value is given by

$$MSE = \frac{1}{M} \sum_{i=0}^{M-1} (y_{t,i} - y_{t+1,i})^2 \tag{5.3}$$

Here, $M$ is the number of measurements, $y_{t,i}$ and $y_{t+1,i}$ are the $i^{th}$ measurements of $y_t$ and $y_{t+1}$ respectively. MSE between CS measurements of WZ frames and their key frames of "Foreman" and "News" are shown in Figure 5.4 together with the corresponding correlation coefficients. It can be observed that, for both videos, whenever the correlation coefficient is high, MSE is low and vice versa. This relationship suggests that for the proposed DCVS codec, MSE can be used as a measure of correlation for the purpose of generating side information for the decoding of WZ frames.

## 5.2 Side Information Generation

In DVC, WZ frames are encoded at much lower rates than key frames. To compensate for this, the reconstruction of WZ frames is aided by side information (SI) that is generated using the key frames at the decoder. SI plays an important role in distributed video coding. If SI is not accurate, then the rate-distortion (RD) performance will suffer. Unfortunately, the performance of DVC is not as good as traditional video coding because SI is generally not as accurate as the reference frame from motion compensated prediction. In the DVC approach, the correlation between frames is performed at the decoder side. The decoder has to estimate the motion between a current frame and its reference frames



5.4.a: Foreman CS Measurements Correlation

5.4.b: Foreman CS Measurements MSE

5.4.c: News CS Measurements Correlation

5.4.d: News CS Measurements MSE

Figure 5.4: Correlation and MSE Comparison of CS Measurements of WZ frames with Key frames

without having access to the current frame. This becomes a problem especially in high motion regions.

The Laplacian distribution is commonly used to model the correlation noise [9, 100, 103]. It provides a good trade-off between model accuracy and complexity [123]. The distributed compressed video sensing scheme proposed in [14] used the Laplacian distribution to model the correlation between WZ frame and its side information frame. The statistical dependency between a WZ frame $W$ and its side information $SI$ is modelled as a virtual correlation channel, where $SI$ is viewed as a noisy version of $W$. The probability density function between $W$ and $SI$ is then modelled using a Laplacian distribution as:

$$p(W(x,y) - SI(x,y)) = \frac{\alpha}{2} e^{-\alpha |W(x,y) - SI(x,y)|} \tag{5.4}$$

where $W(x,y)$ and $SI(x,y)$ are the $(x,y)$-th pixel in $W$ and $SI$ respectively. $\alpha$ is the Laplacian distribution model parameter given by

$$\alpha = \sqrt{\frac{2}{\sigma^2}} \tag{5.5}$$

where $\sigma^2$ is the variance of the residue between the $W$ and $SI$. Therefore, the more similar $W$ and $SI$ are, the higher the value of $\alpha$.

### 5.2.1 Motion Compensated Interpolation

Motion-compensated interpolation (MCI) is the side information generation method used in [14, 72]. Figure 5.5 shows a typical encoder/decoder system with interpolation based side information. Each key frame is encoded by traditional intra frame codecs such as those for MPEG/H.264. Different approaches to encoding the WZ frames in DVC has already been discussed in Chapter 3. The side information for a WZ frame is generated by interpolating the reconstructed key frames. The generated side information is then used to improve the reconstruction quality of WZ frame. There are three common approaches used in generating motion compensated interpolation. The simplest approach is to interpolate neighbouring key frames to estimate the motion, but this generally does

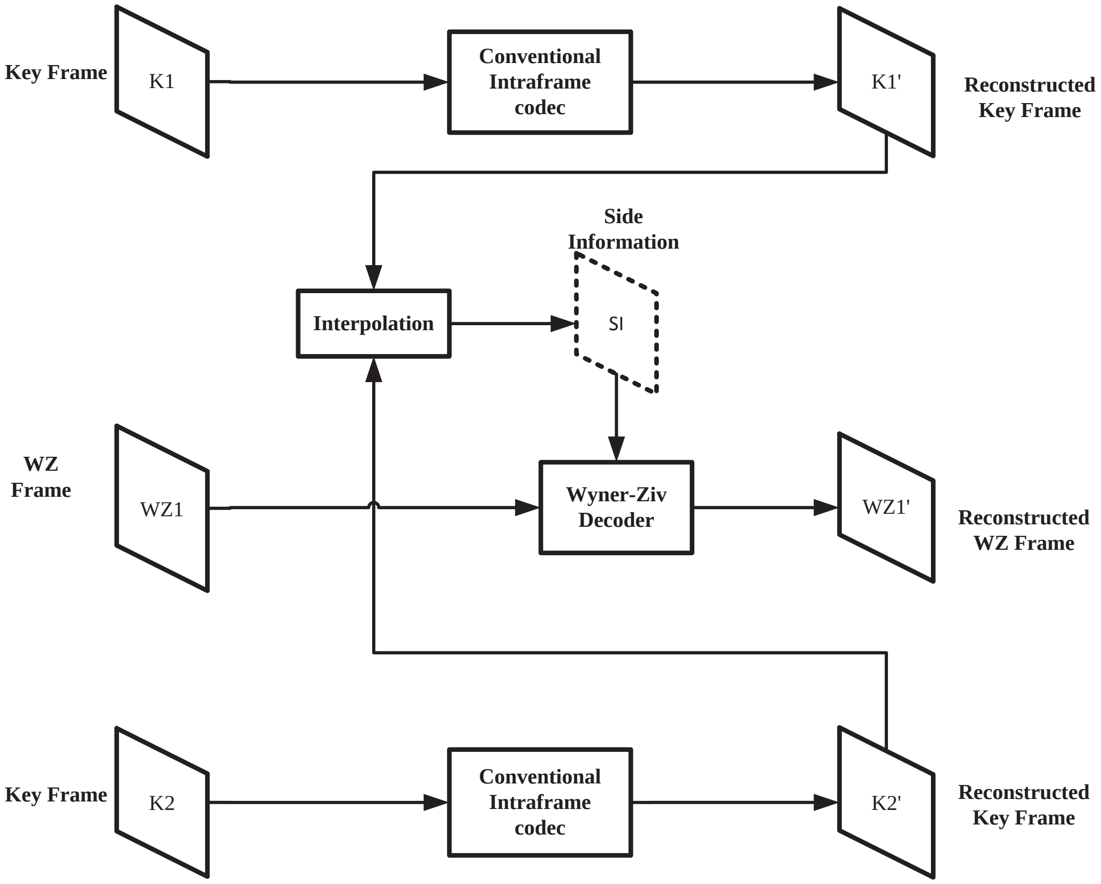


Figure 5.5: DVC Side Information Generation

not produce good results. The second approach makes use of forward motion estimation to predict the motion using the previous frame similar to coding of P frames. The best approach is to use forward and backward motion estimation from previous and future frames. This is similar to motion estimation used for B-frames in MPEG, H.264/AVC coding. This process is shown in Figure 5.6.

Let $W_n$ denote a WZ frame at time $n$, and let $K_{n-1}$ and $K_{n+1}$ be the key frames adjacent to $W_n$. If the motion contained in three successive frames can be assumed to be linear, then the motion vectors for $W_n$ can be derived from the motion vectors from the adjacent two key frames. For forward prediction, if the motion vector of a block $b_i$ in $W_n$ is $MV_f$, then $MV_f$ can be derived from the motion vector of the co-located block

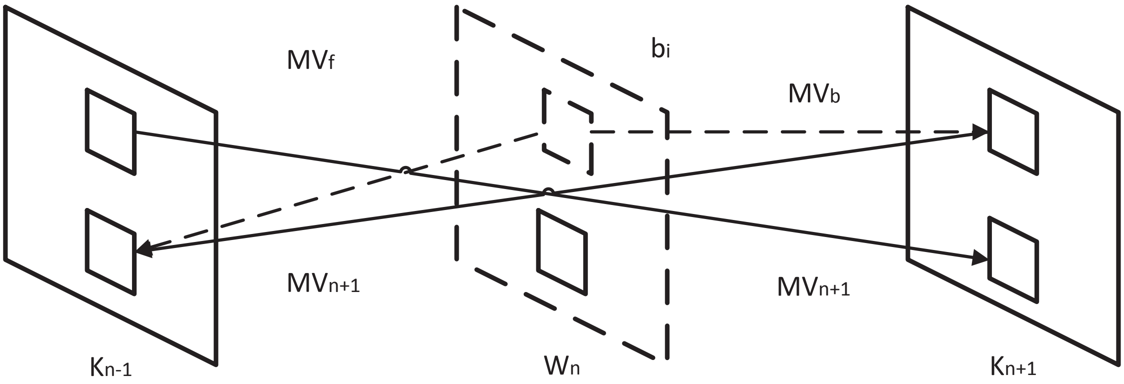


Figure 5.6: Motion Compensated Interpolation

in $K_{n+1}$ by $MV_f = MV_{n+1}/2$. Using the same method, the backward prediction motion vector is given by $MV_b = MV_{n-1}/2$. The two motion predicted blocks of $b_i$ can then be computed from $K_{n-1}$ and $K_{n+1}$. Let $P_b$ represent the prediction value of $b_i$, then $P_b = (P(MV_f) + P(MV_b))/2$ where $P(MV_f)$ and $P(MV_b)$ are the predicted values based on the forward and backward motion vectors respectively. In this way, most blocks of $W_n$ can be predicted. Different block sizes and search ranges can be used for MCI. In traditional video standards, motion estimation is performed at the encoder and only the motion vector is encoded. In DVC, this process is performed at the decoder and is used to provide side information for decoding WZ frames.

### 5.2.2 Proposed SI Generation Method

A new SI generation method is used in proposed CS based video codec shown in Figure 5.1. It has been shown in Section 5.1.3 that CS measurements of adjacent video frames are highly correlated. Therefore we can directly make use of the CS measurements of key frames as side information. A dictionary $D$ that consists of the CS measurements of the key-frames available at the decoder is generated. Starting with an empty dictionary, the first column of $D$, denoted $D_1$, is the CS measurements of the first key frame received. Subsequent columns of $D$ are populated with the CS measurements of the subsequent key frames as they are received. Assume that the GOP size is 3, i.e. each key frame is followed by two WZ frames. When the first two WZ frames are decoded, $D$ has only one column $D_1$. So $D_1$ is used as the side information to reconstruct these two WZ frames. When it comes to reconstructing the third WZ frame $W_3$, the dictionary will have two columns from the two key frames received so far. One of them will be used as side information for $W_3$. The column in the dictionary that is chosen is the one that has the highest correlation with $W_3$. Thus we need to compute the Pearson correlation coefficients

$$r(i) = corr\left(W_3, D_i\right), \qquad i = 1, 2 \tag{5.6}$$

and choose the column $D_1$ if $r(1) > r(2)$ or $D_2$ if $r(2) > r(1)$. This process continues until all the WZ frames are reconstructed. In order to limit the size of the dictionary, only the measurements of the most recent key frames need to be stored as the most recent WZ frame will most likely be more correlated with the most recent key frames.

In Section 5.1.4, it has been shown that CS measurement have a low MSE-high correlation and high MSE-low correlation relationship. Therefore, instead of computing the correlation coefficient, MSE can be computed instead. In this case, SI will be the column in $D$ having the minimum MSE.

There are several advantages with this SI generation method. Firstly, it does not require the key frames to be decoded. In traditional methods such as those using MCI, it is necessary to reconstruct the key frames first and then interpolated to generate the SI for decoding WZ frames. In [13] both block based and frame based CS measurements are combined with decoded key frames to generate the SI. Thus the computational complexity of the proposed method is significantly lower. Secondly, the dictionary does not need to be learnt, again reducing computational complexity. This is in comparison with, e.g. [72], where a dictionary is learned from the neighbouring frames of a video frame. Thirdly, a feedback channel is not required. Some SI generation techniques require a feedback channel from the decoder to the encoder [17, 73]. Fourthly, no motion estimation is required. Finally, the SI obtained in this way can be directly used by the CS reconstruction algorithm without any further processing.

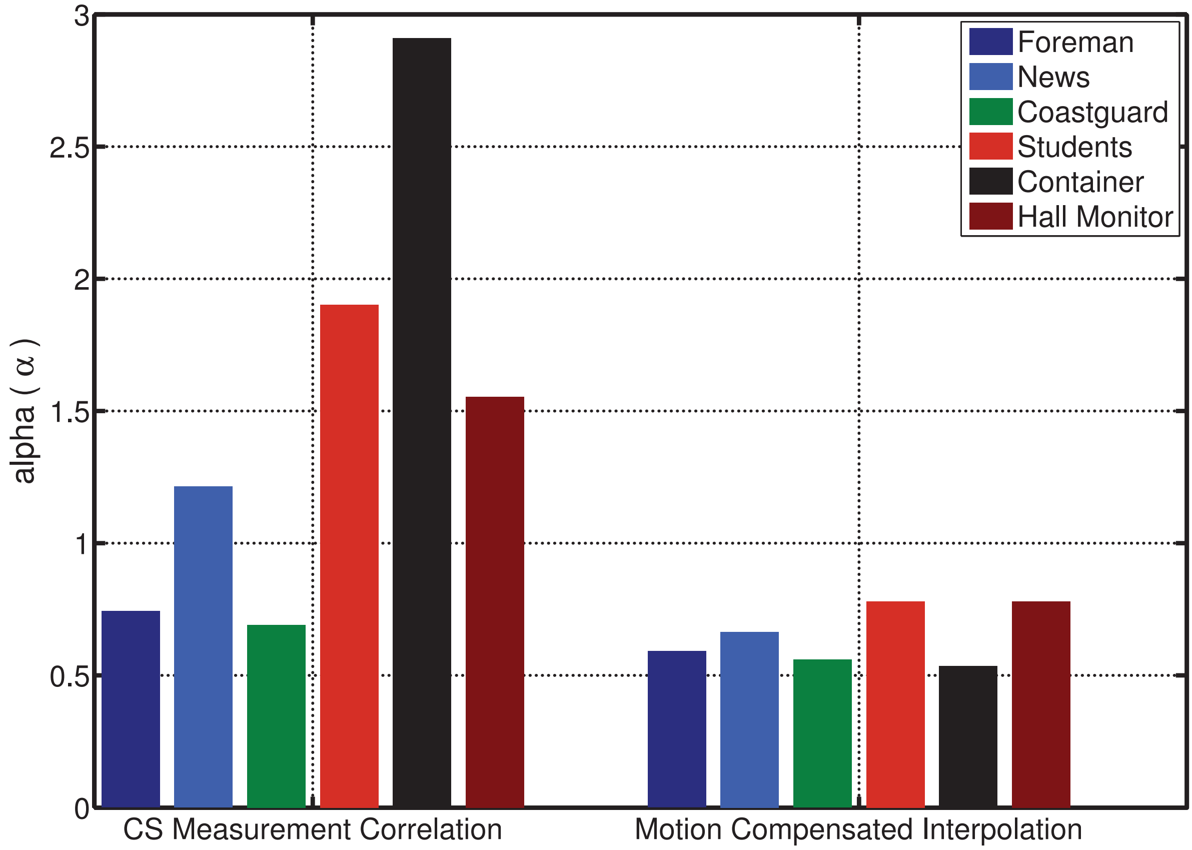


Figure 5.7: Median of Laplacian Distribution Parameter for Two Types of SI

The Laplacian distribution parameter $\alpha$ in Equation 5.5 is computed for the proposed correlation based SI and motion compensated SI. The same video sequences as in Section 5.1.3 are used. Figure 5.7 shows the median values of $\alpha$ for these videos. It can be observed that $\alpha$ is substantially larger for side information generated by the correlation-based method compared with MCI. These results suggest that the proposed correlation-based side information should perform better than MCI based ones.

## 5.3 CS Reconstruction with Side Information

The reconstruction of a WZ frame involves the CS measurements $y_w$ for this frame and a column of the dictionary that has the highest correlation with these measurements, denoted as $SI$. These two sets of data are combined into a single set of measurements

$\beta_w$. If the WZ frame has a CS measurement rate of $M_w$ and the key frames have a measurement rate of $M_k$. Then $\beta_w$ represents the updated measurement rate $M_{w+z}$ for current WZ frame (where $z = k - w$) equal to measurement rate of $M_k$ of key frame. The WZ frame is reconstructed using $\beta_w$. This process is shown as Algorithm 1.

**Algorithm 1** Reconstruction with Side Information

```
Input:$y_w, D$
Output:Reconstructed WZ Frame, $\hat{x}_w$
for each column $i\ inD$ do
    Calculate $r(i) = Correlation(y_w, D(i))$
end for
Calculate $[value, index] = max(r)$
$SI = D(index)$
$\beta_w = [y_w, SI]$
$\hat{x}_w = Reconstruction(\beta_w)$
```

## 5.4 Experimental Results

To evaluate the effectiveness of the proposed distributed compressed sensing video codec, several QCIF (frame size: 172 × 144) video sequences are used [122]. These videos sequences include slow to fast motion videos. Table 5.1 lists the video sequences used in the experiments, specifying their length in terms of the number of frames and the type of content. Those having "low" content type have low spatial details and relatively small amount of movement. For example, "Akiyo" video has a static background and only the newscaster's facial movements. "Medium" content type videos have medium amount of spatial details and motion. For instance, the "foreman" video has disordered motion while "harbour", "coastguard" have slow forward motion. "High" content type videos have higher spatial details and fast camera and object movements. An example is the "Soccer" sequence with rapid motion involving soccer players moving around the pitch.

Only the luminance (Y) component is used in the experiments. CS measurements are obtained using Structurally Random Matrices [52] with Hadamard matrix as the sensing matrix. Daubechies 9/7 wavelets are used as the sparsifying matrix. The GPSR

Table 5.1: Video Test Sequences

| Video Sequence | No: of Frames | Content Type |
|---|---|---|
| Akiyo | 300 | Low |
| Bowing | 300 | Medium |
| Bus | 150 | High |
| Coastguard | 300 | Medium |
| Container | 300 | Low |
| Football | 260 | High |
| Foreman | 300 | Medium |
| Hall Monitor | 300 | Medium |
| Harbour | 150 | Medium |
| Mobile Calendar | 300 | High |
| Mother Daughter | 300 | Low |
| News | 300 | Medium |
| Silent | 300 | Medium |
| Students | 300 | Medium |
| Soccer | 300 | High |

algorithm is used for reconstruction so that the results can be fairly compared with the system proposed in [14] which uses a modified GPSR algorithm with stopping criteria based on side information. However, any algorithm discussed in Section 4.2 could have been used for reconstruction instead. Three different GOP sizes – 3, 5 and 8 are used. The first frame in each GOP is considered a key frame, followed by the respective number of WZ frames. Different measurement rates (MR) have used to evaluate the proposed DCVS method. For example, for a GOP size of 3, an average MR of 37% means that the MRs for the key and non-key (WZ) frames are 50% and 30%, respectively.

The performance of the proposed DCVS codec is compared with three other techniques. The first one is Frame DWT which is a basic CS video codec that does not exploit any side information in recovering WZ frames. In Frame DWT, all WZ frames are reconstructed as per their corresponding MR at the decoder. The second one is found in [14] which uses a relative stopping criteria based on the side information generated through an efficient frame rate up-conversion tool. The side information is generated using MCI from previous reconstructed key frames. This technique is labelled as "DCVS" in subsequent results. The third technique is motion compensated interpolation without any relative stopping criteria. This technique is labelled as "MCI" in subsequent results.

All codecs are coded in MATLAB and simulations run on an Intel i5 3.6GHz, Windows 7 Enterprise Edition, 64-bit Operating System in MATLAB R2012b 64-bit with 4GB RAM. For fair analysis, no other programs were running on the simulation computer except the operating system and MATLAB.

### 5.4.1 Reconstruction Complexity Evaluation

The reconstruction complexity for different schemes are evaluated by calculating the average reconstruction time (in seconds) for key frames and WZ frames. The results for the six video sequences with three different GOP sizes are shown graphically in Figure 5.8 and tabulated numerically in Table 5.2. It can be observed that using SI generally improves the reconstruction time regardless of the type of SI used. The proposed SI scheme performs better than MCI based side information because it does not require motion estimation. However, its performance is worse than DCVS because DCVS uses warm start in the GPSR algorithm which reduces the reconstruction time. It also used a relative stopping criteria based on fixed thresholds to reduce the number of iterations during reconstruction.

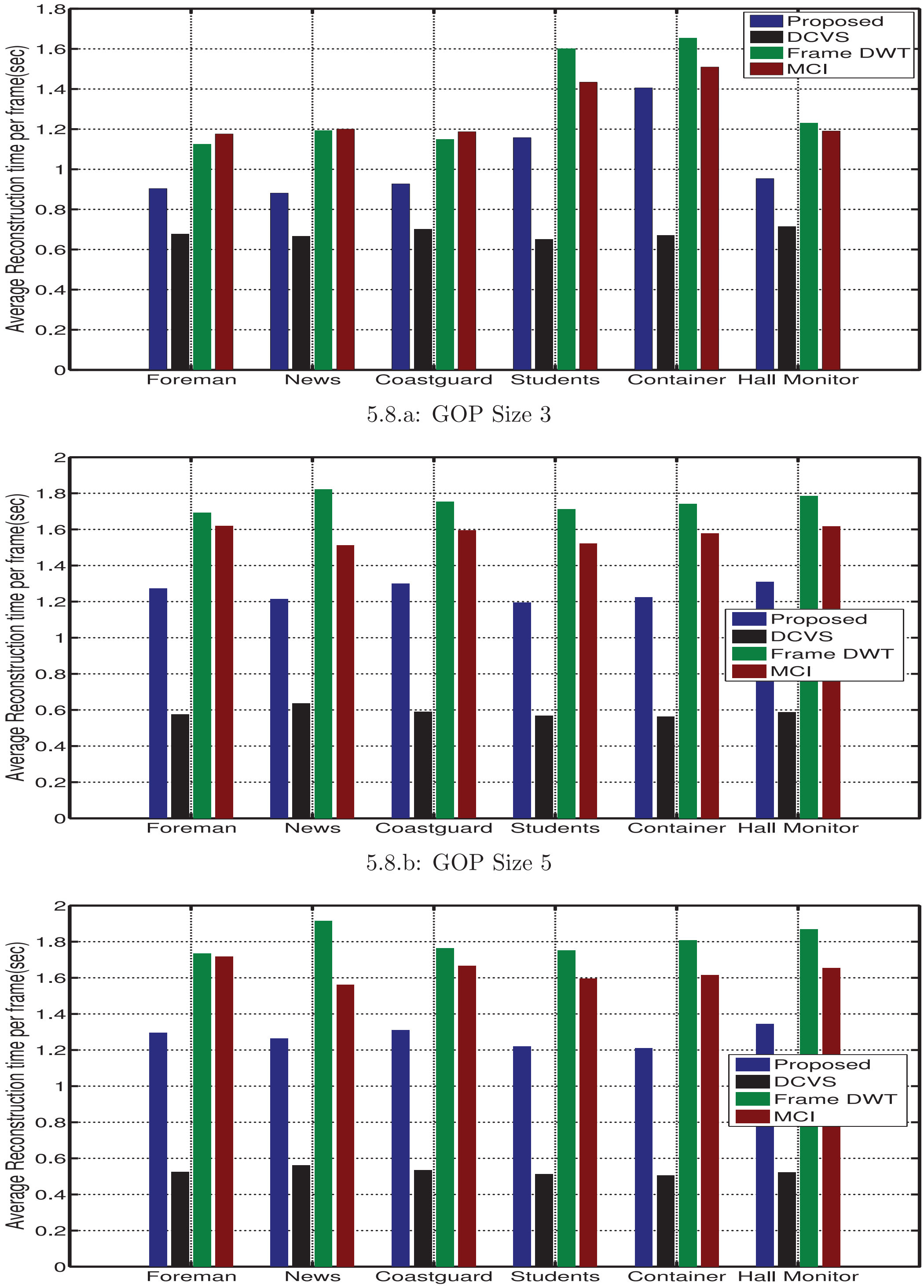
### 5.4.2 Rate Distortion Evaluation

Rate Distortion (R-D) analysis for CS based codecs are quite different from traditional coding. R-D curves in the CS video coding literature typically plots Measurement Rate (MR) against achieved reconstruction quality in terms of PSNR instead of bit rates.

Figure 5.9 shows the R-D curves for four test video sequences ("News", "Container", "Students" and "Soccer") using a GOP size of 3. With these four videos, all three content types are represented. The proposed codec outperforms all the other three codecs for the slow motion video "Container". Its performance is comparable to both MCI and DCVS

Table 5.2: Average Reconstruction Time (in Seconds) of Video Sequences

| **Video** | **GOP Size 3, Ave MR=27%** | | | | **GOP Size 5, Ave MR=24%** | | | | **GOP Size 8, Ave MR=23%** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Frame DWT | DCVS | MCI | Proposed | Frame DWT | DCVS | MCI | Proposed | Frame DWT | DCVS | MCI | Proposed |
| **Akiyo** | 0.98 | 0.58 | 0.98 | 0.72 | 1.56 | 0.52 | 1.25 | 0.96 | 1.62 | 0.47 | 1.32 | 0.96 |
| **Bowing** | 0.99 | 0.63 | 1.08 | 0.84 | 1.52 | 0.55 | 1.43 | 1.16 | 1.52 | 0.51 | 1.5 | 1.19 |
| **Bus** | 1.7 | 0.75 | 1.66 | 1.34 | 1.71 | 0.61 | 1.71 | 1.34 | 1.79 | 0.55 | 1.7 | 1.31 |
| **Coast guard** | 1.15 | 0.7 | 1.19 | 0.93 | 1.75 | 0.59 | 1.59 | 1.3 | 1.76 | 0.53 | 1.66 | 1.3 |
| **Container** | 1.65 | 0.67 | 1.51 | 1.4 | 1.74 | 0.56 | 1.58 | 1.22 | 1.81 | 0.51 | 1.61 | 1.21 |
| **Football** | 1.67 | 0.72 | 1.59 | 1.28 | 1.77 | 0.59 | 1.6 | 1.25 | 1.81 | 0.53 | 1.67 | 1.24 |
| **Foreman** | 1.12 | 0.68 | 1.18 | 0.9 | 1.69 | 0.58 | 1.62 | 1.27 | 1.74 | 0.52 | 1.71 | 1.29 |
| **Hall Monitor** | 1.23 | 0.72 | 1.19 | 0.95 | 1.79 | 0.59 | 1.61 | 1.31 | 1.86 | 0.52 | 1.65 | 1.34 |
| **Harbour** | 1.54 | 0.73 | 1.65 | 1.37 | 1.56 | 0.61 | 1.7 | 1.33 | 1.55 | 0.54 | 1.72 | 1.35 |
| **Mobile Calendar** | 1.6 | 0.71 | 1.62 | 1.34 | 1.67 | 0.59 | 1.68 | 1.32 | 1.72 | 0.52 | 1.72 | 1.33 |
| **Mother Daughter** | 1.08 | 0.68 | 1.16 | 0.88 | 1.52 | 0.57 | 1.5 | 1.24 | 1.55 | 0.52 | 1.57 | 1.23 |
| **News** | 1.19 | 0.67 | 0.88 | 1.2 | 1.82 | 0.64 | 1.51 | 1.21 | 1.91 | 0.56 | 1.56 | 1.26 |
| **Silent** | 0.99 | 0.64 | 1.1 | 0.8 | 1.51 | 0.54 | 1.44 | 1.01 | 1.55 | 0.5 | 1.5 | 1.11 |
| **Students** | 1.65 | 0.68 | 1.51 | 1.24 | 1.7 | 0.57 | 1.59 | 1.23 | 1.78 | 0.51 | 1.6 | 1.26 |
| **Soccer** | 1.6 | 0.65 | 1.43 | 1.16 | 1.71 | 0.57 | 1.52 | 1.2 | 1.75 | 0.51 | 1.6 | 1.22 |



5.8.a: GOP Size 3

5.8.b: GOP Size 5

5.8.c: GOP Size 8

Figure 5.8: Reconstruction complexity comparison of Video Sequences

for medium motion videos “News” and ”Students”. The same is true for the high motion video “Soccer”. In fact, in this case MCI could not compensate for the motion present in the video and its performance is even worse than Frame DWT with no SI for average measurement rate of above 28%.

Table 5.3 shows R-D performance of all test video sequences at three different measurement rates of 17%, 27%, and 37%. The performance of the proposed codec is better than MCI and DCVS for all slow motion videos (“Akiyo”, “Bowing”,“Container” and “Mother Daughter”). For medium motion videos, its performance is better for “News”, “Silent” and “Students” and comparable to DCVS for “Foreman”, “Coastguard” and “Harbour”. For high motion videos, the proposed codec performs better at the higher measurement rate (37%) while comparable at the two lower MRs. These results show that even though correlation based side information is much simpler to generate, it provides a reconstruction quality that is similar to or better than MCI.

Similar results are obtained for larger GOP sizes. R-D curves for the four test videos as in Figure 5.9 are shown with GOP sizes of 5 and 8 in Figures 5.10 and 5.11 respectively. In general, with increased GOP size, the performance gap between the proposed codec performance and DCVS and MCI also increases. Tables 5.4 and 5.5 show the PSNR performance of all video sequences at different MRs for GOP sizes of 5 and 8, respectively. In terms of Structurally Similarity Index (SSIM) [124], Table 5.6 shows average SSIM index for GOP size 3, 5 and 8. For slow motion videos SSIM index of propsoed codec performance is similar or better than other codecs. The proposed codec provides better results with high motion vidoes “Football” and “Soccer”.

Figure 5.12 shows the visual reconstruction quality for the $89^{th}$ frame (a WZ frame) of the “News” video with a WZ measurement rate of 30%. It can be observed that the proposed scheme produces better visual quality of this reconstructed frame than DCVS and MCI. Similar observation can be made from Figure 5.13 for the $56^{th}$ frame (a WZ frame) of the “Container” video .

Table 5.3: Rate Distortion Performance (in dB) of Video Sequences for GOP Size 3

| **Video** | **Ave MR=17%** | | | | **Ave MR=27%** | | | | **Ave MR=37%** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Frame DWT | DCVS | MCI | Prop osed | Frame DWT | DCVS | MCI | Prop osed | Frame DWT | DCVS | MCI | Prop osed |
| **Akiyo** | 24.9 | 28.51 | 28.44 | **29.03** | 28.15 | 30.82 | 30.51 | **31.27** | 30.54 | 32.85 | 32.38 | **33.27** |
| **Bowing** | 24.49 | **27.62** | 27.24 | 27.61 | 27.48 | 29.5 | 29.06 | **29.63** | 29.78 | 31.14 | 30.69 | **31.47** |
| **Bus** | 14.48 | **15.51** | 15.41 | 15.38 | 15.43 | **16.58** | 16.44 | 16.5 | 16.57 | 17.22 | 17.57 | **17.74** |
| **Coast guard** | 20.43 | **23.52** | 22.87 | 23.33 | 22.99 | **25.34** | 24.61 | 25.1 | 25.12 | **27.02** | 26.36 | 26.81 |
| **Container** | 19.99 | 23.21 | 22.87 | **23.94** | 22.99 | 25.46 | 25.2 | **26.73** | 25.75 | 27.37 | 27.31 | **29.25** |
| **Football** | 17.77 | 18.42 | 18.34 | **18.52** | 18.91 | 19.51 | 19.34 | **19.61** | 20.21 | 20.53 | 20.45 | **20.83** |
| **Foreman** | 21.44 | **24.77** | 24.13 | 24.33 | 24.47 | **26.62** | 25.99 | 26.19 | 26.91 | **28.24** | 27.71 | 27.9 |
| **Hall Monitor** | 20.78 | **24.4** | 24.07 | 24.25 | 23.54 | **26.8** | 26.41 | 26.68 | 26.09 | **29.25** | 28.74 | 29.1 |
| **Harbour** | 18.73 | **21.04** | 20.67 | 20.78 | 20.46 | **22.65** | 22.22 | 22.38 | 22.22 | **24.25** | 23.8 | 24.01 |
| **Mobile Calendar** | 16.69 | **18.98** | 18.58 | 18.69 | 18.36 | **20.32** | 19.8 | 20.03 | 19.82 | **21.65** | 21.08 | 21.45 |
| **Mother Daughter** | 25.05 | **28.78** | 28.37 | 28.77 | 27.98 | 30.76 | 30.31 | **30.82** | 30.3 | 32.72 | 32.14 | **32.72** |
| **News** | 19.52 | 23.29 | 22.88 | **23.32** | 22.59 | 25.65 | 25.18 | **25.84** | 25.26 | 27.66 | 27.24 | **28.1** |
| **Silent** | 21.55 | 24.38 | 24.03 | **24.7** | 24.21 | 26.05 | 25.82 | **26.7** | 26.52 | 27.59 | 27.51 | **28.57** |
| **Students** | 21.2 | **24.3** | 23.99 | 24.26 | 23.56 | 26.12 | **25.77** | 26.14 | 25.61 | 27.89 | 27.51 | **28.05** |
| **Soccer** | 16.63 | 17.35 | 17.35 | **17.59** | 17.86 | 18.5 | 18.42 | **18.85** | 19.32 | 19.67 | 19.63 | **20.24** |

Table 5.4: Rate Distortion Performance (in dB) of Video Sequences for GOP Size 5

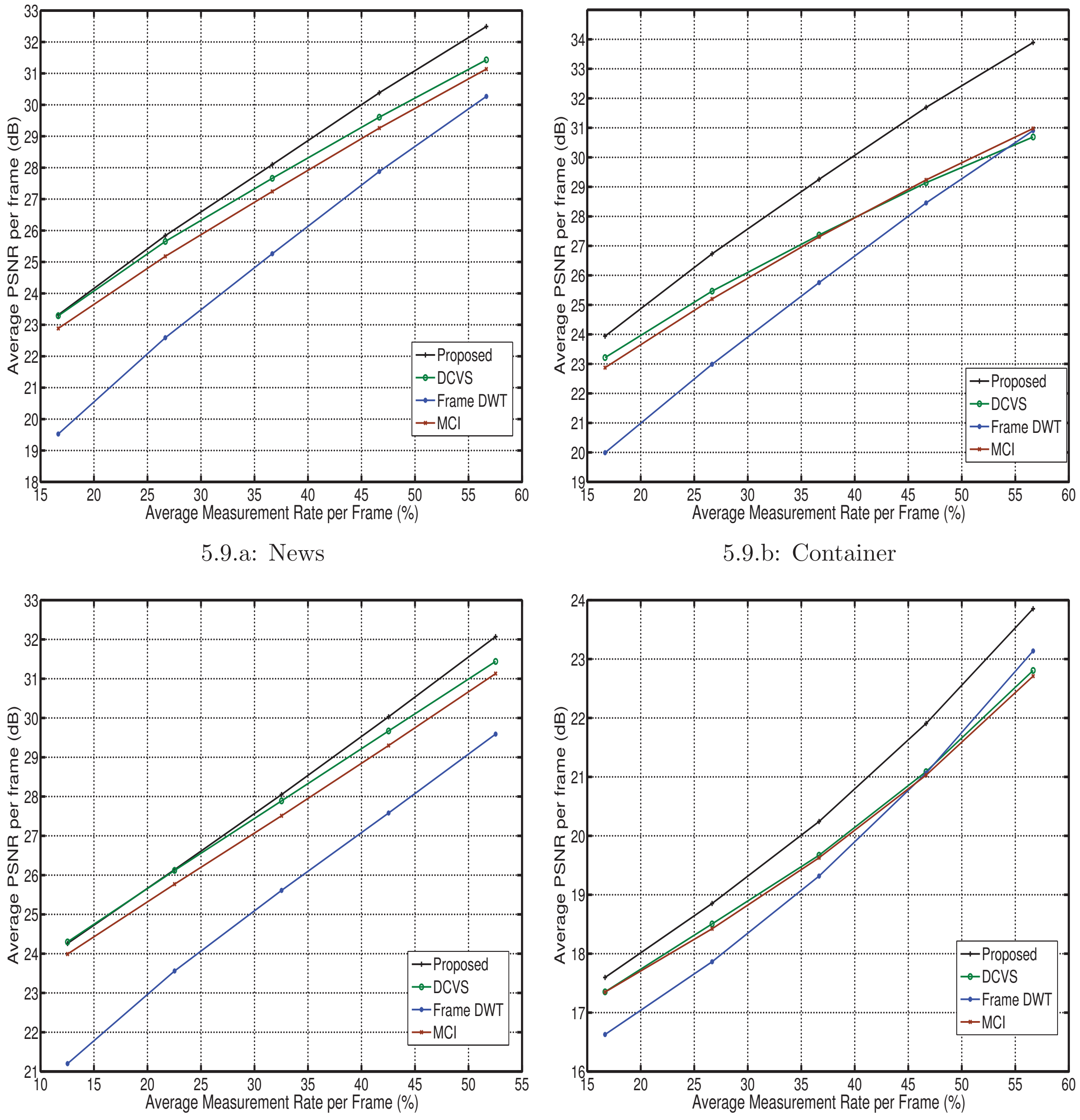
| **Video** | **Ave MR=17%** | | | | **Ave MR=27%** | | | | **Ave MR=37%** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Frame DWT | DCVS | MCI | Prop osed | Frame DWT | DCVS | MCI | Prop osed | Frame DWT | DCVS | MCI | Prop osed |
| **Akiyo** | 24.06 | 28.44 | 28.22 | **28.94** | 27.5 | 30.72 | 30.22 | **31.16** | 29.96 | 32.64 | 32.03 | **33.13** |
| **Bowing** | 23.72 | 26.93 | 26.57 | **27.3** | 26.87 | 28.7 | 28.25 | **29.26** | 29.21 | 30.19 | 29.76 | **31.04** |
| **Bus** | 14.24 | **15.36** | 15.24 | 15.23 | 15.13 | **16.37** | 16.18 | 16.29 | 16.22 | 17.44 | 17.24 | **17.47** |
| **Coast guard** | 19.76 | **23.17** | 22.35 | 22.94 | 22.43 | **24.8** | 23.96 | 24.54 | 24.58 | **26.35** | 25.57 | 26.07 |
| **Container** | 19.19 | 23.08 | 22.66 | **23.91** | 22.23 | 25.2 | 24.88 | **26.69** | 25.04 | 26.94 | 26.85 | **29.18** |
| **Football** | 17.47 | 18.05 | 17.99 | **18.27** | 18.57 | 19.03 | 18.84 | **19.25** | 19.82 | 19.82 | 19.8 | **20.36** |
| **Foreman** | 20.63 | **23.97** | 23.3 | 23.69 | 23.82 | **25.65** | 25.02 | 25.41 | 26.34 | **27.09** | 26.58 | 26.98 |
| **Hall Monitor** | 20.06 | **24.35** | 23.91 | 24.14 | 22.87 | **26.69** | 26.15 | 26.49 | 25.4 | **28.97** | 28.38 | 28.83 |
| **Harbour** | 18.28 | **20.9** | 20.42 | 20.59 | 20 | **22.46** | 21.89 | 22.13 | 21.75 | **23.98** | 23.39 | 23.68 |
| **Mobile Calendar** | 16.26 | **18.88** | 18.14 | 18.47 | 17.98 | **20.23** | 19.33 | 19.72 | 19.43 | **21.49** | 20.59 | 21.04 |
| **Mother Daughter** | 24.27 | **28.69** | 28.09 | 28.61 | 27.36 | 30.57 | 29.95 | **30.59** | 29.75 | 32.39 | 31.7 | **32.42** |
| **News** | 18.72 | **23.11** | 22.56 | 23.04 | 21.86 | 25.31 | 24.73 | **25.41** | 24.58 | 27.16 | 26.66 | **27.48** |
| **Silent** | 20.84 | 23.97 | 23.57 | **24.41** | 23.59 | 25.51 | 25.24 | **26.32** | 25.96 | 26.96 | 26.82 | **28.09** |
| **Students** | 20.58 | **24.27** | 23.84 | 24.18 | 23.02 | **26.04** | 25.56 | 26.03 | 25.09 | 27.7 | 27.23 | **27.89** |
| **Soccer** | 16.31 | 16.94 | 16.96 | **17.29** | 17.48 | 17.97 | 17.9 | **18.44** | 18.89 | 18.95 | 18.94 | **19.7** |

Table 5.5: Rate Distortion Performance (in dB) of Video Sequences for GOP Size 8

| **Video** | **Ave MR=17%** | | | | **Ave MR=27%** | | | | **Ave MR=37%** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Frame DWT | DCVS | MCI | Prop osed | Frame DWT | DCVS | MCI | Prop osed | Frame DWT | DCVS | MCI | Prop osed |
| **Akiyo** | 23.6 | 28.4 | 28.01 | **28.86** | 27.15 | 30.59 | 29.94 | **31.03** | 29.64 | 32.35 | 31.68 | **32.96** |
| **Bowing** | 23.28 | 26.16 | 25.88 | **26.88** | 26.52 | 27.83 | 27.47 | **28.79** | 28.9 | 29.25 | 28.9 | **30.51** |
| **Bus** | 14.1 | **15.1** | 14.98 | 15.07 | 14.96 | **16.13** | 15.89 | 16.1 | 16.03 | 17.15 | 16.89 | **17.2** |
| **Coast guard** | 19.39 | **22.58** | 21.82 | 22.51 | 22.12 | **24.05** | 23.31 | 23.97 | 24.29 | **25.52** | 24.79 | 25.36 |
| **Container** | 18.76 | 23.02 | 22.52 | **23.88** | 21.81 | 25.01 | 24.59 | **26.61** | 24.64 | 26.63 | 26.49 | **29.06** |
| **Football** | 17.32 | 17.8 | 17.78 | **18.06** | 18.38 | 18.72 | 18.55 | **18.97** | 19.6 | 19.4 | 19.43 | **20** |
| **Foreman** | 20.18 | 23.01 | 22.46 | **23.01** | 23.46 | 24.51 | 24.03 | **24.57** | 26.03 | 25.81 | 25.46 | **26** |
| **Hall Monitor** | 19.66 | **24.2** | 23.7 | 23.99 | 22.49 | **26.45** | 25.85 | 26.25 | 25.04 | **28.61** | 27.98 | 28.51 |
| **Harbour** | 18.02 | **20.63** | 20.11 | 20.35 | 19.75 | **22.1** | 21.48 | 21.8 | 21.49 | **23.49** | 22.87 | 23.25 |
| **Mobile Calendar** | 16.02 | **18.67** | 17.8 | 18.14 | 17.77 | **19.95** | 18.97 | 19.28 | 19.22 | **21.22** | 20.17 | 20.47 |
| **Mother Daughter** | 23.84 | **28.49** | 27.8 | 28.41 | 27.02 | 30.29 | 29.57 | **30.33** | 29.45 | 31.99 | 31.26 | **32.1** |
| **News** | 18.27 | 22.75 | 22.18 | **22.87** | 21.46 | 24.77 | 24.18 | **25.15** | 24.2 | 26.49 | 26 | **27.15** |
| **Silent** | 20.44 | 23.58 | 23.17 | **24.14** | 23.25 | 25.03 | 24.76 | **25.96** | 25.65 | 26.49 | 26.26 | **27.66** |
| **Students** | 20.24 | **24.19** | 23.67 | 24.07 | 22.73 | **25.89** | 25.32 | 25.88 | 24.81 | 27.47 | 26.92 | **27.67** |
| **Soccer** | 16.13 | 16.58 | 16.66 | **17** | 17.28 | 17.55 | 17.5 | **18.08** | 18.64 | 18.37 | 18.44 | **19.24** |

Table 5.6: Average SSIM Index Performance of Video Sequences for GOP Size 3, 5 and 8

| **Video** | **GOP Size = 3** | | | | **GOP Size = 5** | | | | **GOP Size = 8** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Frame DWT | DCVS | MCI | Proposed | Frame DWT | DCVS | MCI | Proposed | Frame DWT | DCVS | MCI | Proposed |
| **Akiyo** | 0.72 | **0.85** | 0.83 | 0.84 | 0.70 | **0.85** | 0.82 | 0.84 | 0.68 | 0.84 | 0.82 | **0.84** |
| **Bowing** | 0.70 | **0.81** | 0.78 | 0.79 | 0.67 | **0.79** | 0.75 | 0.78 | 0.66 | 0.76 | 0.71 | **0.76** |
| **Bus** | 0.40 | 0.53 | 0.52 | **0.54** | 0.36 | 0.50 | 0.49 | **0.52** | 0.34 | 0.46 | 0.45 | **0.50** |
| **Coast guard** | 0.52 | **0.67** | 0.62 | 0.63 | 0.49 | **0.65** | 0.58 | 0.60 | 0.47 | **0.61** | 0.54 | 0.57 |
| **Container** | 0.54 | 0.69 | 0.65 | **0.71** | 0.51 | 0.69 | 0.64 | **0.71** | 0.49 | 0.69 | 0.63 | **0.70** |
| **Football** | 0.42 | 0.46 | 0.45 | **0.50** | 0.38 | 0.40 | 0.40 | **0.46** | 0.37 | 0.36 | 0.37 | **0.44** |
| **Foreman** | 0.60 | 0.79 | 0.68 | **0.83** | 0.57 | **0.69** | 0.63 | 0.63 | 0.55 | **0.64** | 0.57 | 0.59 |
| **Hall Monitor** | 0.59 | **0.77** | 0.74 | 0.74 | 0.56 | **0.77** | 0.73 | 0.73 | 0.55 | **0.76** | 0.72 | 0.72 |
| **Harbour** | 0.42 | 0.59 | 0.57 | **0.63** | 0.39 | **0.63** | 0.59 | 0.61 | 0.43 | **0.60** | 0.56 | 0.59 |
| **Mobile Calendar** | 0.41 | **0.56** | 0.52 | 0.55 | 0.38 | **0.54** | 0.47 | 0.52 | 0.36 | **0.52** | 0.47 | 0.48 |
| **Mother Daughter** | 0.70 | **0.83** | 0.80 | 0.81 | 0.67 | **0.83** | 0.79 | 0.80 | 0.66 | **0.82** | 0.78 | 0.79 |
| **News** | 0.55 | **0.72** | 0.69 | 0.71 | 0.51 | **0.72** | 0.67 | 0.69 | 0.50 | **0.70** | 0.64 | 0.68 |
| **Silent** | 0.58 | **0.73** | 0.69 | 0.71 | 0.55 | **0.71** | 0.67 | 0.70 | 0.54 | **0.70** | 0.64 | 0.68 |
| **Students** | 0.76 | **0.84** | 0.82 | 0.82 | 0.75 | **0.82** | 0.78 | 0.79 | 0.74 | **0.78** | 0.74 | 0.75 |
| **Soccer** | 0.45 | 0.52 | 0.51 | **0.57** | 0.41 | 0.46 | 0.46 | **0.54** | 0.39 | 0.42 | 0.42 | **0.52** |



5.9.a: News

5.9.b: Container

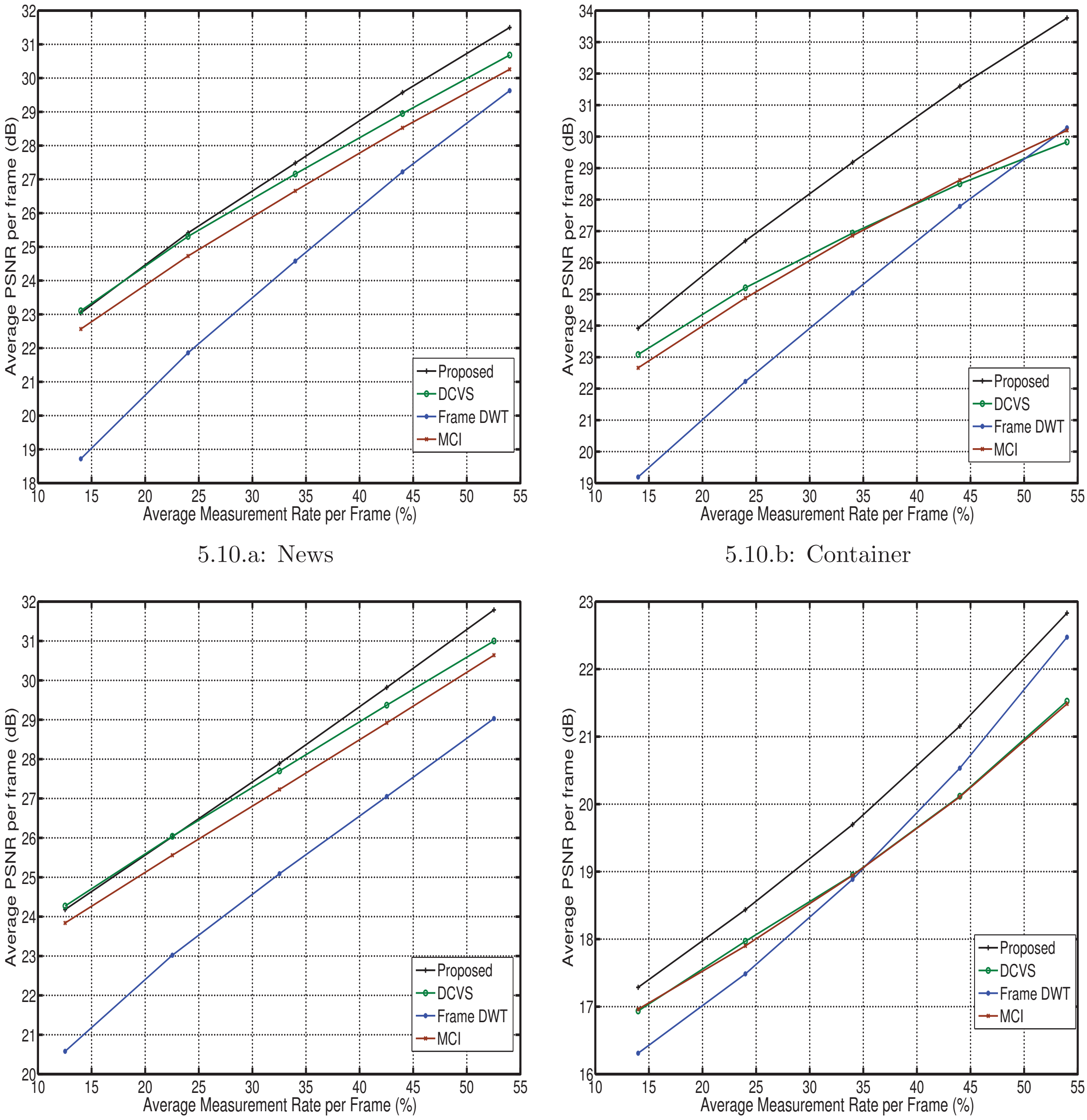
5.9.c: Students

5.9.d: Soccer

Figure 5.9: Rate Distortion Curve for GOP Size 3

### 5.4.3 Performance Comparison with Distributed and Conventional Codecs

In the previous section, the proposed codec is compared with other distributed codecs that makes use of CS. In this section, it is compared against an efficient distributed video



5.10.a: News

5.10.b: Container

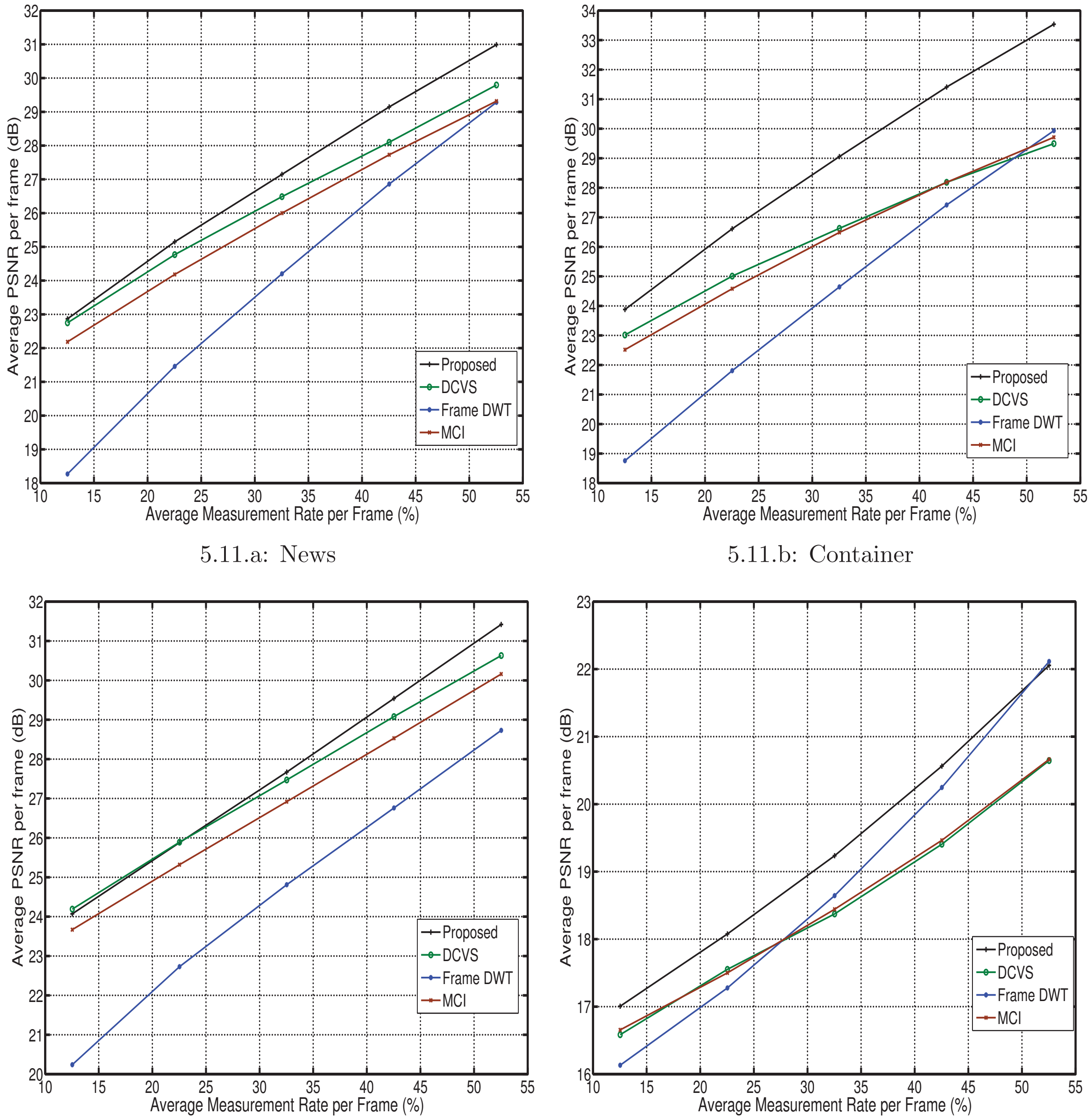
5.10.c: Students

5.10.d: Soccer

Figure 5.10: Rate Distortion Curve for GOP Size 5

codec called "DISCOVER" [103] as discussed in Section 3.3.3 and existing conventional video codecs H.264 and H.263+ in terms of compression ratio and bit rates.

The proposed codec that is used for comparison is the same as that used in Section 5.4 except for the choice of CS reconstruction algorithm. Previously, GPSR is used



5.11.a: News

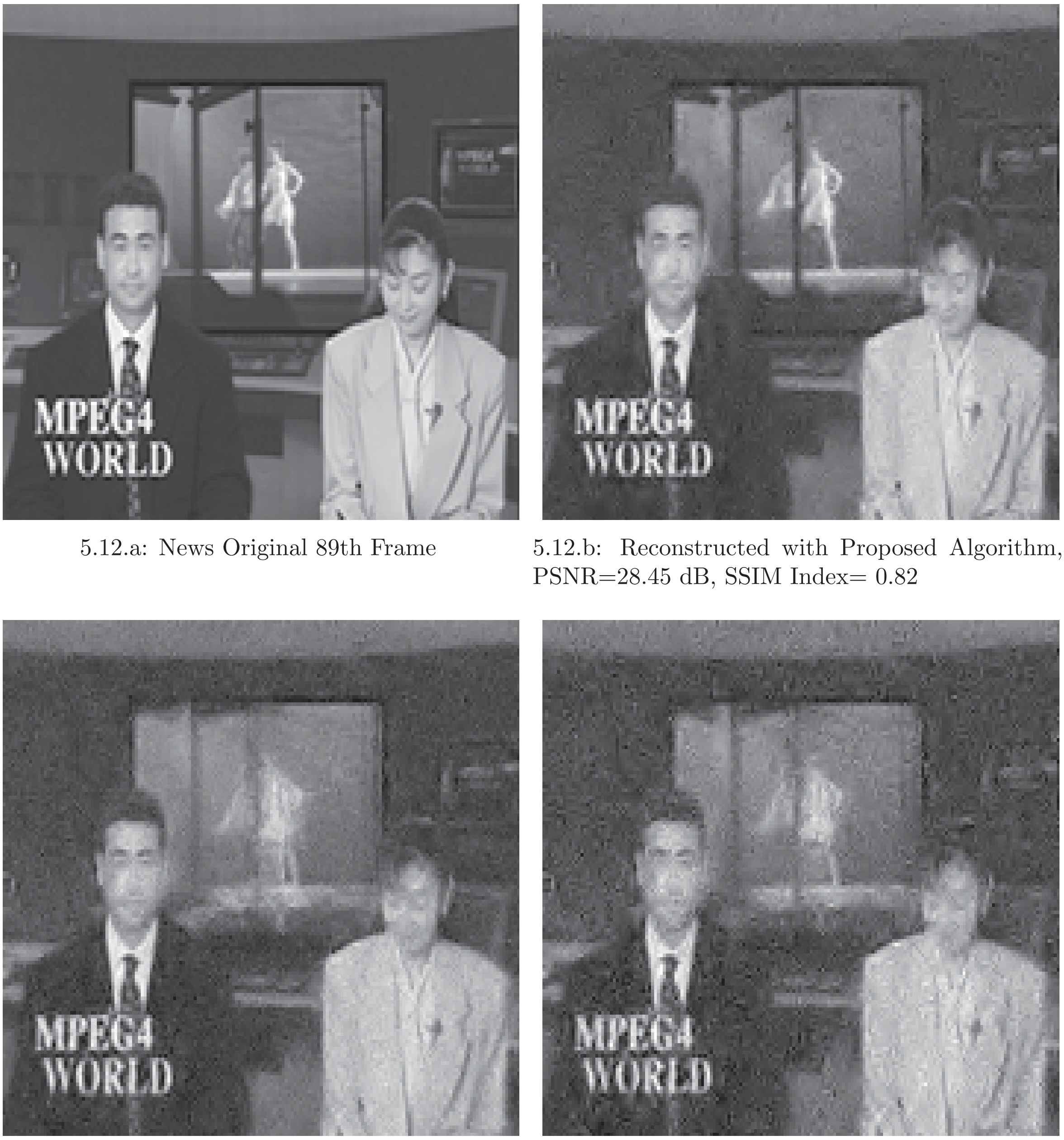
5.11.b: Container

5.11.c: Students

5.11.d: Soccer

Figure 5.11: Rate Distortion Curve for GOP Size 8

so that a fair comparison can be made with DCVS. Here, the more efficient SpaRSA algorithm since it has found to be the best among different reconstruction algorithms in Section 4.2. The CS measurements are quantized using the quantization scheme developed in Section 4.3.2. Each CS measurement is allotted 8 bits for quantization. The



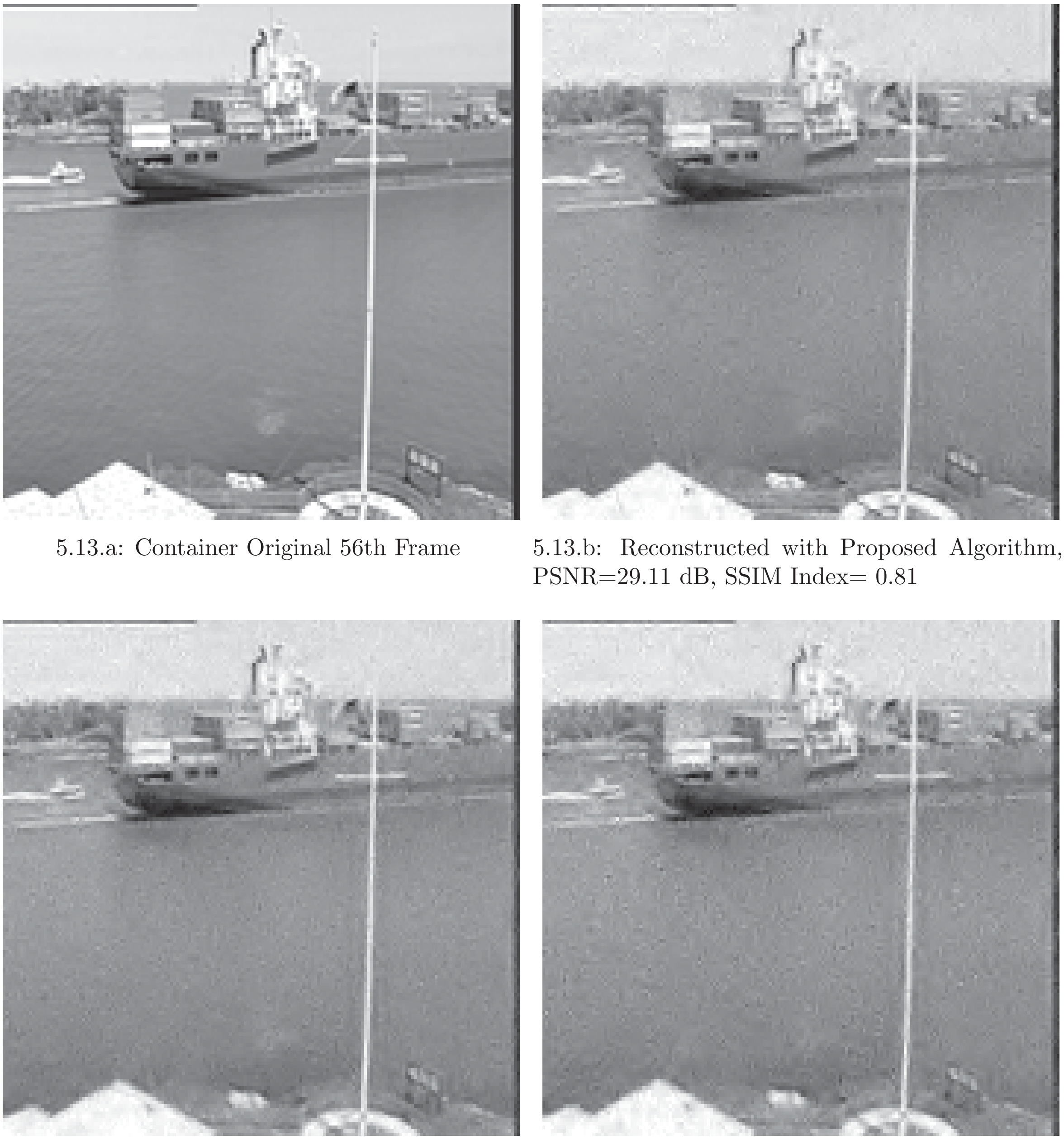
5.12.a: News Original 89th Frame

5.12.b: Reconstructed with Proposed Algorithm, PSNR=28.45 dB, SSIM Index= 0.82

5.12.c: Reconstructed with DCVS, PSNR=23.66 dB, SSIM Index= 0.67

5.12.d: Reconstructed with MCI, PSNR=23.34 dB, SSIM Index= 0.59

Figure 5.12: Visual Reconstruction Quality of News 89th Frame for GOP Size 3



5.13.a: Container Original 56th Frame

5.13.b: Reconstructed with Proposed Algorithm, PSNR=29.11 dB, SSIM Index= 0.81

5.13.c: Reconstructed with DCVS, PSNR=26.28 dB, SSIM Index= 0.76

5.13.d: Reconstructed with MCI, PSNR=26.23 dB, SSIM Index= 0.71

Figure 5.13: Visual Reconstruction Quality of Container 56th Frame for GOP Size 3

quantized CS measurements are then entropy coded using Huffman coding. The compression ratio is then calculated as

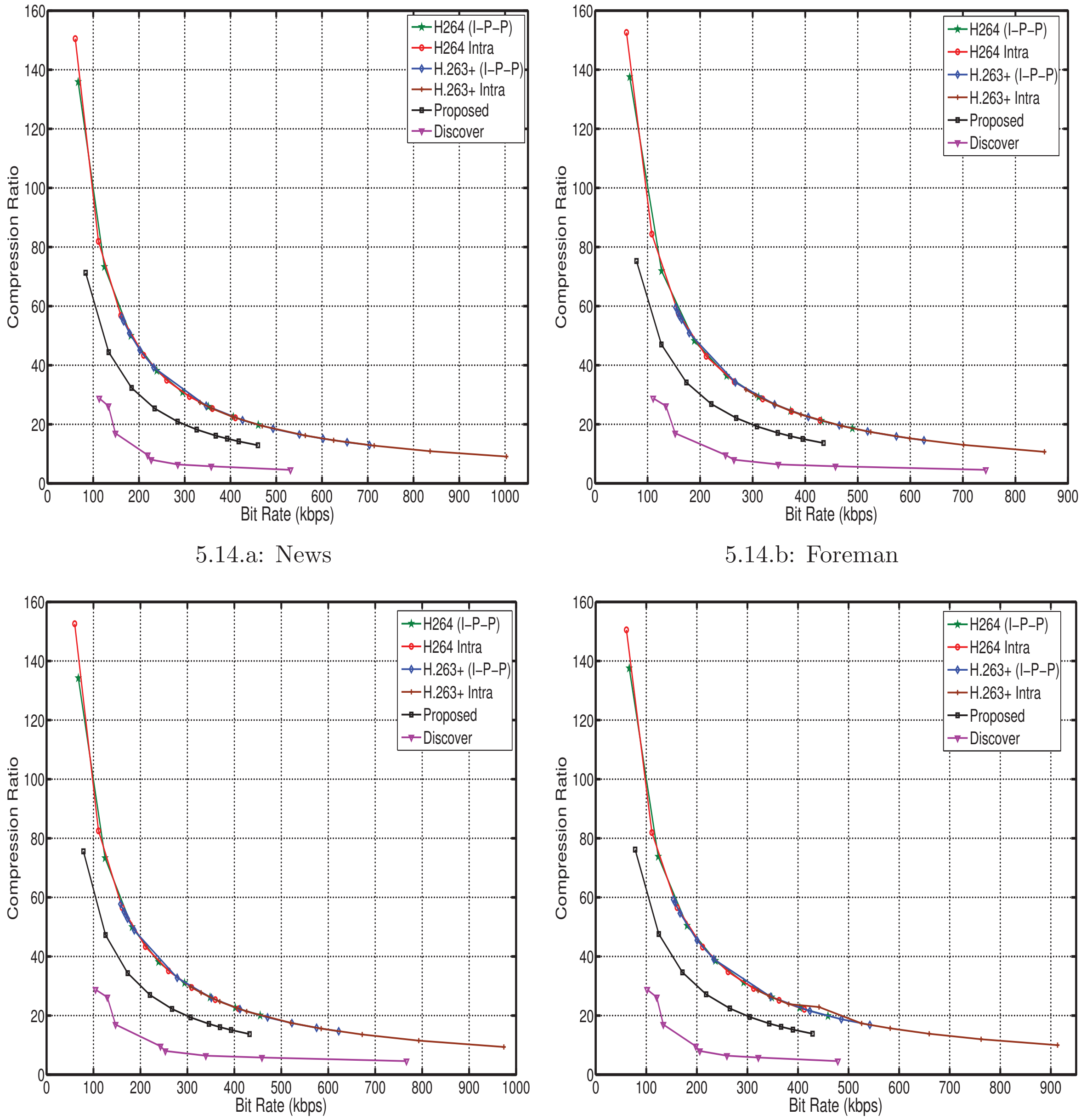
$$\text{Compression Ratio} = \frac{\text{Uncompressed Size}}{\text{Compressed Size}} \tag{5.7}$$

Each raw (uncompressed) pixel requires 8 bits since only the luminance component is used. Four QCIF videos – "Foreman", "News", "Coastguard" and "Hall Monitor" are used in the experiments. These videos have a frame rate of 30 frames per second. A GOP size of 3 is used.

### Codec Compared

The three codecs chosen to be compared with the proposed codec are DISCOVER, H.264, and H.263+. DISCOVER has been discussed in detail in Section 3.3.3. The DISCOVER codec software available from the DISCOVER website [125] is used in our experiments. H.264 is a video codec standard developed by ITU-T [3]. This coding standard is also known as H.264/AVC (Advanced Video Coding). It is one of the best known video codecs available and it supports good video quality under low bit rates. H.264/AVC supports many profiles. In these experiments, the Main profile which is suitable for a standard definition TV broadcast is used. Two different coding options are used – H.264 (I-P-P) in which GOP size 3 is used and H.264 Intra which uses only Intra coding of frames.

H.263+ is a video codec standard developed by ITU-T [126]. This coding standard is an enhanced version of H.263. Two different coding options for H.263+ are chosen in these experiments. H.263+ (I-P-P), in which GOP size 3 is used and H.263+ Intra which uses only Intra coding of frames under different quantization parameters. The quantization for Intra mode chooses is 8,10,12,14,16,20,24,28 and 32 to obtain different bit rates and quality. The H.264 and H.263+ codecs used are those provided by the free software FFmpeg (Fast Forward MPEG) [127]. FFmpeg is a free software which supports encoding/decoding of several video codecs. It is a command line software which allows many parameters like bit rate, GOP size, quantization parameters to be set. FFmpeg encodes the video and return PSNR(Y, U and V separately), bit rate and encoding time for YUV raw videos.



5.14.a: News

5.14.b: Foreman

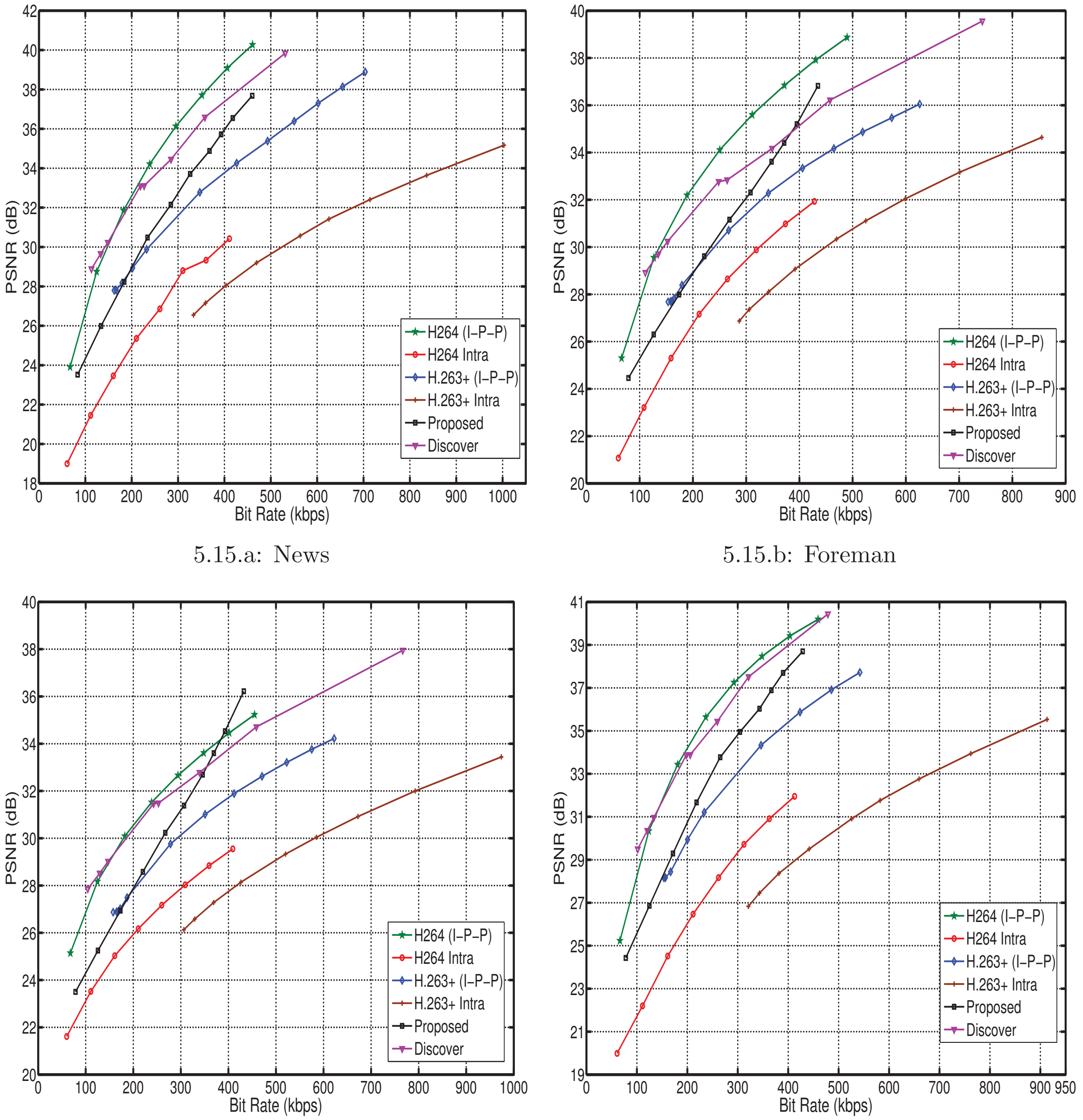
5.14.c: Coastguard

5.14.d: Hall Monitor

Figure 5.14: Bit Rate vs Compression Ratio for Video Sequences - GOP Size 3

### Results

First, the compression ratio that are achieved by the codecs are compared. Figure 5.14 shows the compression ratio under different bit rates for the four test video sequences. The proposed codec outperforms DISCOVER at all bit rates. However its performance



5.15.a: News

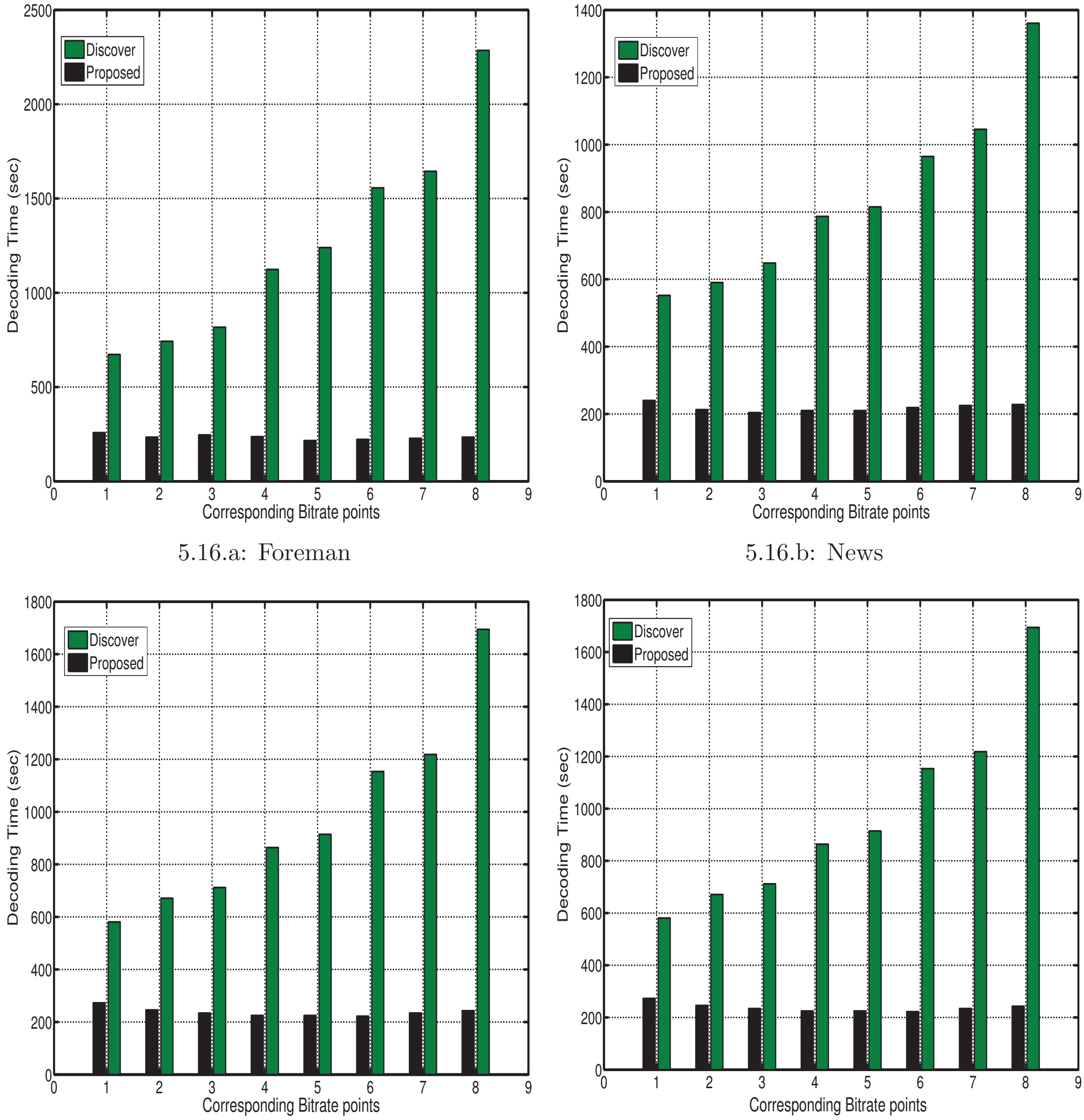
5.15.b: Foreman

5.15.c: Coastguard

5.15.d: Hall Monitor

Figure 5.15: Bit Rate vs PSNR for Video Sequences - GOP Size 3

against conventional video codecs is not as good due to the efficient intra coding used by H.264 and H.263+ which produces higher compression. The reconstruction quality of different codecs at different bit rates are shown in Figure 5.15. The proposed codec performs better than H.264 Intra, H.263+ Intra and H.263+ (I-P-P) for all four test videos at all bit rates. For "Foreman" and "Coastguard", the proposed codec is better



5.16.a: Foreman

5.16.b: News

5.16.c: Coastguard

5.16.d: Hall Monitor

Figure 5.16: Reconstruction Time Complexity Comparison with DISCOVER

than DISCOVER [103] at higher bit rates. It is important to note that DISCOVER uses a feedback channel to improve the quality of WZ frames and also uses H.264 Intra coding for key frames.

The decoding times required by the proposed codec and by DISCOVER are shown in Figure 5.16. Different quantization parameters were used in DISCOVER to achieve different bit rates and quality [128], corresponding Bitrate points on $x - axis$ shows corresponding bitrates in proposed codec and DISCOVER. Note that conventional codec are designed in such a way that the encoding process is complex and decoding process is simple whereas the proposed codec is designed with a different philosophy. Hence their decoding complexity are not compared.

The decoding time required by the proposed distributed compressed sensing video codec stays roughly the same at all bit rates. However, that required for DISCOVER increases as bit rate increases. Furthermore, the decoding time for the proposed codec is much shorter than DISCOVER at all bit rates. The results are similar as reported on DISCOVER performance evaluation website [128].

## 5.5 Summary

In this chapter, a simple but effective distributed video compressed sensing codec is proposed. The encoding is entirely performed using compressed sensing which can be implemented with much reduced hardware complexity compared with conventional video coders. At the decoder, a simple side information generation technique is proposed that is based on the correlation analysis of CS measurements between video frames. This technique does not require a feedback channel nor motion estimation that are usually required by other distributed codecs. It does not even need the key frames to be decoded before side information can be generated. The proposed technique directly uses CS measurements of the key frames. The proposed SI technique can easily be integrated with any CS reconstruction algorithm without need of modification such as those required by DCVS. Detailed experimental results have been performed using videos with slow, medium and fast motions. The proposed codec performs comparably with DCVS with much simpler decoding. It is also better than H.264 Intra, H.263 Intra and H.263 (I-P-P) in terms of reconstructed video quality. Furthermore, it outperforms DISCOVER in terms of simplicity, compression ratio and reconstruction complexity.

# Chapter 6

# Distributed CS Image Compression

In the previous two chapters, CS measurements are acquired for the whole image or the whole frame of a video. With the desire to reduce the complexity of both the encoder and decoder, a distributed approach to CS based image compression is considered in this Chapter. The full image is divided into relatively small blocks and CS measurements for each block are then acquired. This is a well established idea in traditional image coding. The advantage is that the size of the measurement matrix for block-based CS acquisition is much smaller than that for the full image. Also, decoding delay is reduced because the decoder can start decoding each block as they become available. Furthermore, computational complexity for reconstruction is also reduced due to the smaller block sizes. Block based CS encoding is first proposed in [54].

Structurally random matrices (SRM) that have been evaluated in Chapter 4.1 can be effectively used for block-based CS acquisition. The reconstruction strategy together with the choice of block size are discussed in Section 6.1. Based on that, a distributed intra image coding scheme is proposed in Section 6.3. Some blocks are designated as key blocks and others as non-key blocks. The non-key blocks are encoded in reduced rates. At the decoder, the reconstruction performance of the non-key blocks is improved by using side information generated from the key blocks. Experimental results are presented in Section 6.4 to show the effectiveness of proposed distributed image codec.

## 6.1 Block Based Encoding and Decoding

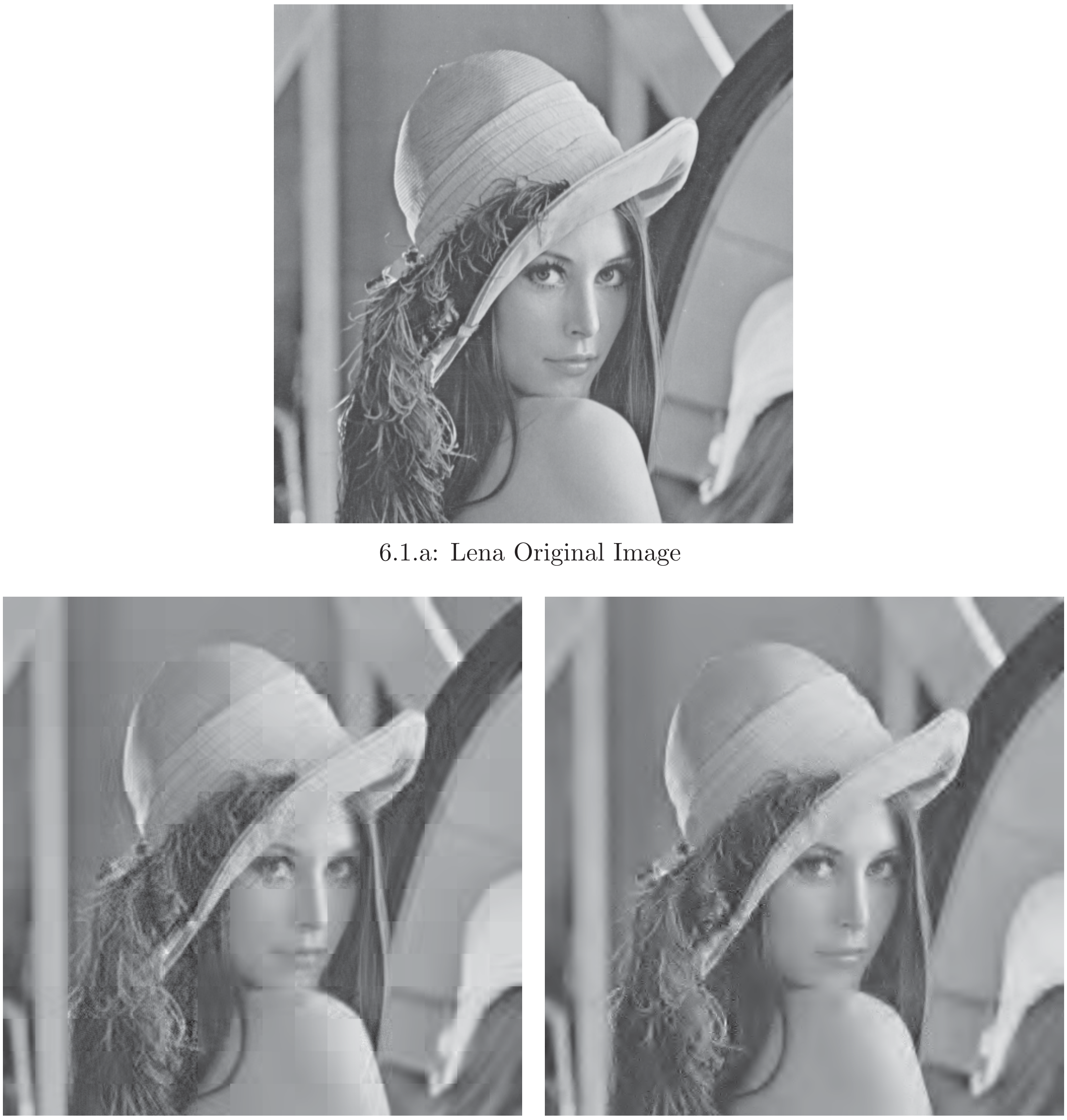
In Block based CS, an image $X$ with dimension $N \times N$ pixels is divided into non-overlapping blocks of $B \times B$ pixels. Let $x_i$ represent the $i$-th block, then its $M_r$ CS measurements are given by

$$y_i = \Phi_B x_i \tag{6.1}$$

where $\Phi_B$ is an $M_r \times B^2$ measurement matrix, $y_i$ is the $M_b \times 1$ CS measurement vector. The measurement rate per block is given by $M_b = M_r/B^2$. Any sensing matrix discussed in Section 4.1 can be used. However, structurally random matrices as proposed in [52] has some advantages. Firstly, SRM does not need to be stored explicitly for CS acquisition or reconstruction which helps to reduce the complexity of the encoders. Secondly, acquisition using SRM is originally designed to be block based. Hence, SRM will be used exclusively.

### 6.1.1 Recovery Methods

There are three approaches by which the full image can be recovered from the block-based CS measurements. The first one is to reconstruct each block from its CS measurements independently of the other blocks, using the corresponding sensing and block sparsifying matrices. This will be referred to as the independent block recovery approach. Another approach is to place block sparsifying transform in to a block diagonal sparsifying matrix and then use the block diagonal sparsifying transform to reconstruct the full image [74]. In this approach, instead of reconstructing independent blocks, CS measurements of all blocks are used to reconstruct full image. The reconstruction performance in block diagonal and independent block recovery are almost similar [74]. Alternatively, all the image blocks are reconstructed jointly using a full sparsifying matrix [74]. This approach will be called joint recovery. In joint recovery, instead of using block sparsifying matrix or block diagonal sparsifying matrix, a full sparsifying transform is used to reconstruct full image. CS measurements of each sampled blocks are combined together at the decoder



6.1.a: Lena Original Image

6.1.b: Lena Independent Block Recovery, PSNR=28.04dB

6.1.c: Lena Joint Recovery, PSNR=28.23

Figure 6.1: Block Based CS Reconstruction Comparison

and then a full image transform is applied to reconstruct full image. This is more useful than independent and diagonal reconstruction as full transform provides more sparser representation and avoided blocking artefacts as well.

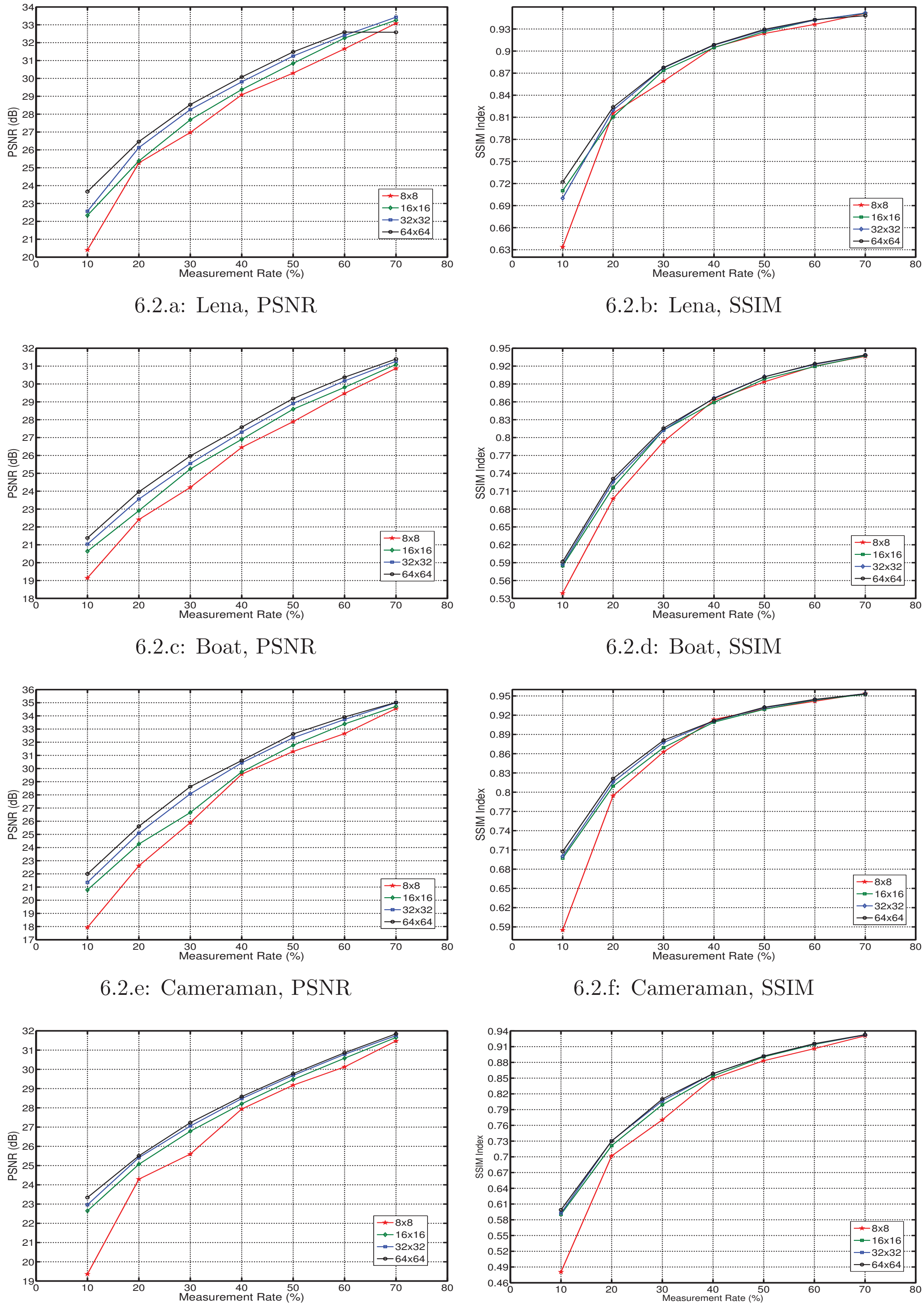
Generally speaking, the difference in PSNR between independent and joint recovery is not significant. An example of the visual qualities for the two recovery approaches is shown in Figure 6.1. Blocking artefacts are clearly visible in the independently recovered image and the jointly recovered one does not have such artefacts. At the decoder, the image blocks can be recovered as they become available at the decoder and develops an initial solution. Once all the blocks are available at the decoder, a joint reconstruction is performed for a visually better results.

### 6.1.2 Impact of Block Size

A block size of $32 \times 32$ or $16 \times 16$ is typically employed in previous studies of block based CS for images [54, 55, 60]. In [54], the authors conducted an empirical study and suggest using $32 \times 32$. However, no details of this study were given. In this section, an empirical study of the impact of block size on the quality of the reconstructed image is performed.

For this study, the publicly available test images “Lena”, “Boat”, “Cameraman” and “Goldhill” are used. Each of these images has a dimension of $512 \times 512$ pixels. Four different block sizes – $64 \times 64$, $32 \times 32$, $16 \times 16$, and $8 \times 8$ are tested. Reconstruction performance is evaluated in terms of PSNR and the structural similarity index (SSIM) [124] which measures the structural quality. The performance is obtained for measurement rates $M_r$ ranging from 0.1 to 0.7. The experiment is run five times for each measurement rate and then the average is computed.

Figure 6.2 shows the PSNR and SSIM for the four test images. For all test images, a larger block size provides better reconstruction quality in terms of PSNR. At very low measurement rates, the performance of $8 \times 8$ block size is the worst. However, as the MR increases the differences in both PSNR and SSIM are no longer significant for all block sizes. This results suggests that for a lower MR, a larger block size provides better results. For moderate MR, i.e. between 0.2 and 0.4, the difference in performance is not significant and larger block sizes perform slightly better than smaller block ones. Thus the choice of block size depends very much on the resources available for encoding and constraints on the processing and storage of CS measurements.



6.2.a: Lena, PSNR

6.2.b: Lena, SSIM

6.2.c: Boat, PSNR

6.2.d: Boat, SSIM

6.2.e: Cameraman, PSNR

6.2.f: Cameraman, SSIM

6.2.g: Goldhill, PSNR

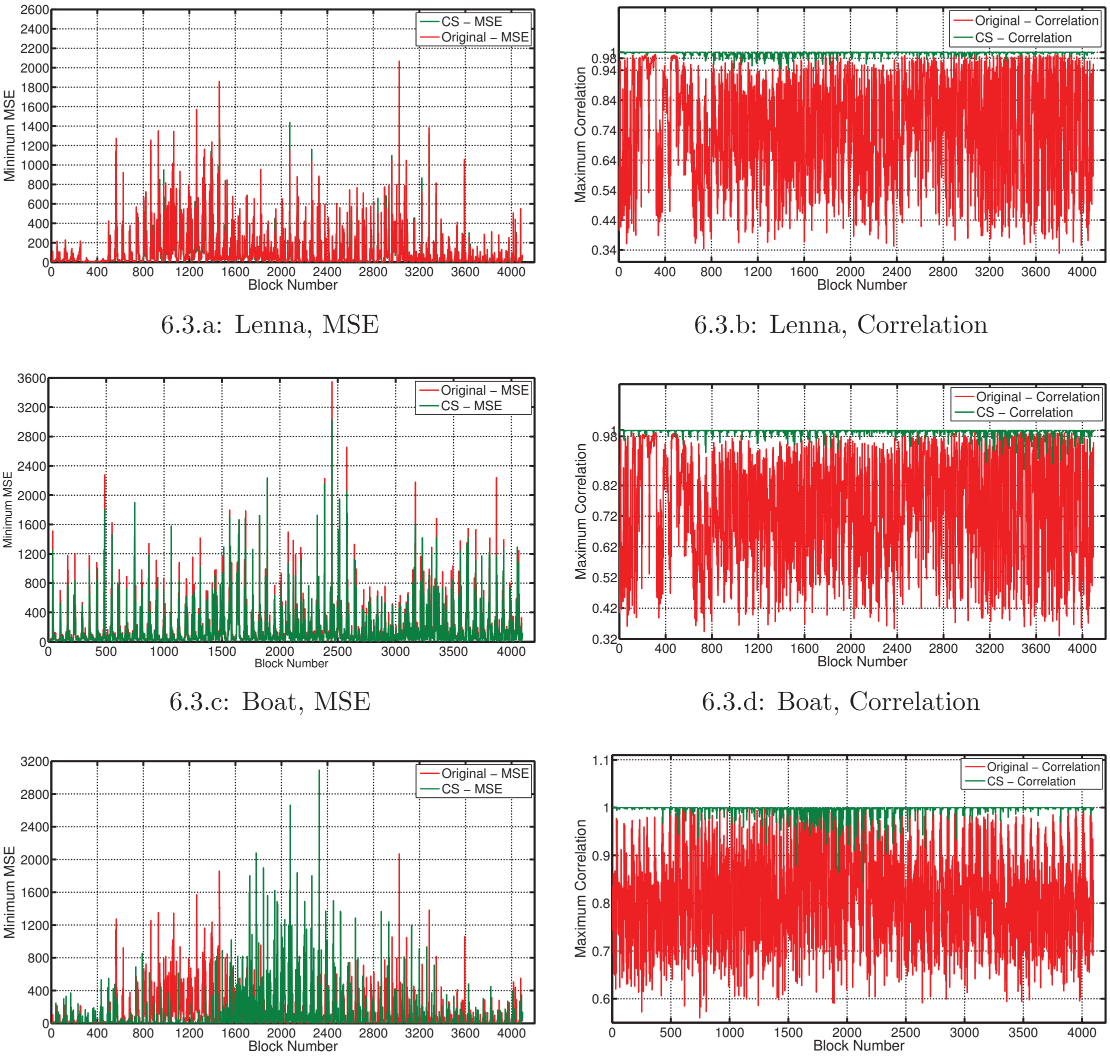
6.2.h: Goldhill, SSIM

Figure 6.2: Rate Distortion Performance of Block Size (64, 32, 16, and 8)

## 6.2 Block Similarity Analysis

Consider an image divided into non-overlapping rectangular blocks of pixels. In general, there are correlations between neighbouring image blocks. If the block size is small, this correlation is high. For image blocks that are correlated, their CS measurements are also correlated. An efficient CS encoding strategy can be formulated to reduce the measurement rate of correlated blocks. This is analogous to exploiting the correlation between video frames to improve performance in Chapter 5. In this section, image block correlation will be analysed quantitatively. Three standard $512 \times 512$ pixel test images – “Lena”, “Cameraman”, and “Boat”, are used. Each image is divided into non-overlapping blocks. The CS measurement rate for each block is fixed at 50%. Image block similarity is measured through the correlation coefficient and Mean Square Error (MSE).

The comparison between correlation and MSE for image blocks using a block size of $8 \times 8$ pixels is shown in Figure 6.3. This is computed for original pixel data and CS measurement of each block. The maximum correlation of each block with all other blocks and minimum MSE of each block with all other blocks is plotted. MSE values of CS data and original image data aligns very well with each other with most of the values being very close. This indicates that MSE of the CS measurements can be used as a reflection of the MSE of the original image data. Correlation between CS measurements are very high. This result suggests that similar image block has low MSE value of CS measurements. To evaluate the effect of block size on image block similarity, the percentage of correlated blocks exceeding an MSE threshold is computed for block sizes of $64 \times 64$, $32 \times 32$, $16 \times 16$, and $8 \times 8$. Figure 6.4 shows the results for the previous three images together with that for “Goldhill”. As expected, there are more correlated blocks for smaller block sizes. Blocks with a size of $64 \times 64$ can virtually be considered independent. The percentage of blocks that can be considered correlated depends on the MSE threshold. If the threshold is set too high, then most of the blocks in the image will be considered correlated. An appropriate threshold will need to be determine when block correlation is applied to the proposed distributed image codec.



6.3.a: Lenna, MSE

6.3.b: Lenna, Correlation

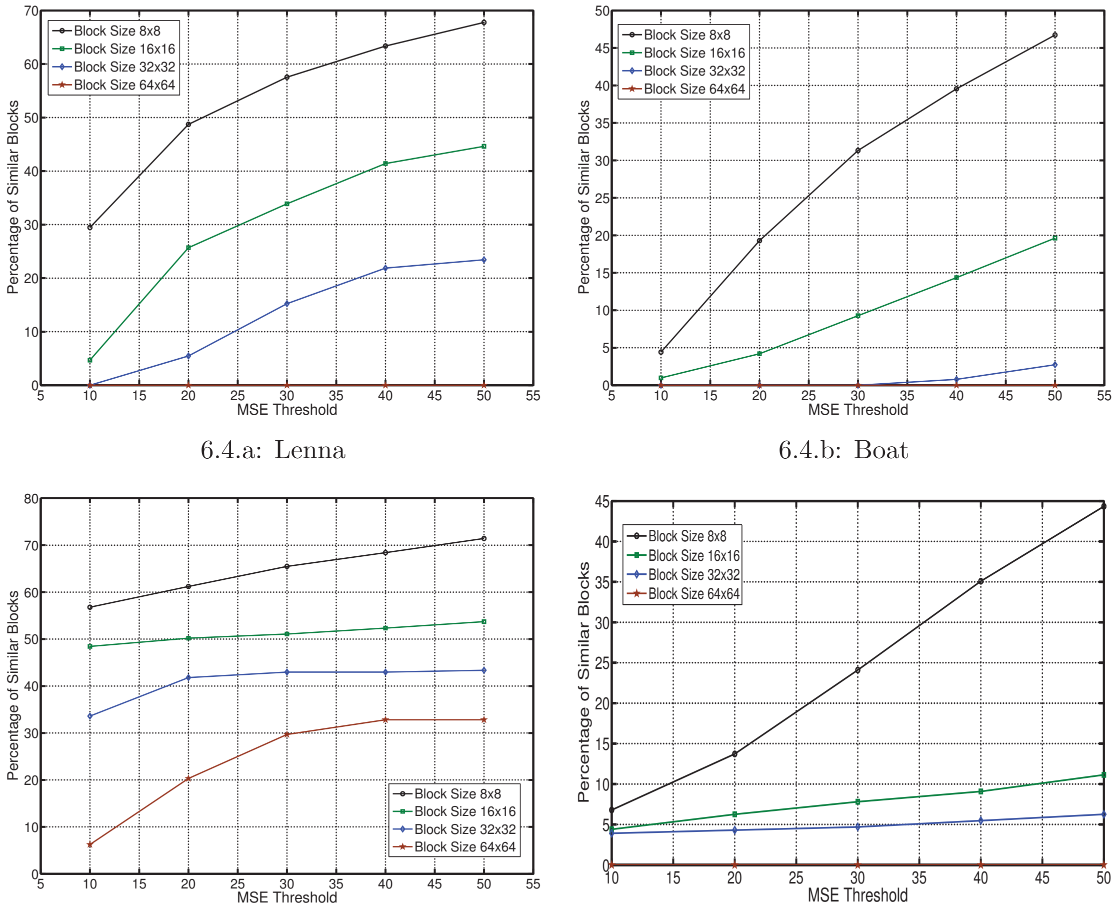
6.3.c: Boat, MSE

6.3.d: Boat, Correlation

6.3.e: Cameraman, MSE

6.3.f: Cameraman, Correlation

Figure 6.3: Block Similarity Analysis of Original Pixel Data and CS measurements with Correlation Coefficient and MSE



6.4.a: Lenna

6.4.b: Boat

6.4.c: Camerman

6.4.d: Goldhill

Figure 6.4: Percentage of Similar Blocks for Block Size (64, 32, 16 and 8)

## 6.3 Proposed Distributed Image Codec

A distributed image codec is proposed in this section that makes use of the correlation between neighbouring blocks. Image blocks are classified either as key blocks or non-key (WZ) blocks. Key blocks are encoded with a higher measurement rate than non-key blocks similar to the way key frames and non-key frames of a video sequence are treated in Chapter 5. Two block classification strategies – adaptive and non-adaptive, are used. They are described in detail in Section 6.3.1 respectively. Side information (SI) is generated for WZ blocks at the decoder using the CS measurements of Key blocks and this process is discussed in more detail in Section 6.3.2. Each Key block is reconstructed from its own measurements. For each non-key block, reconstruction is performed using its own CS measurements together with the SI generated. The block diagram of this codec is shown in Figure 6.5.

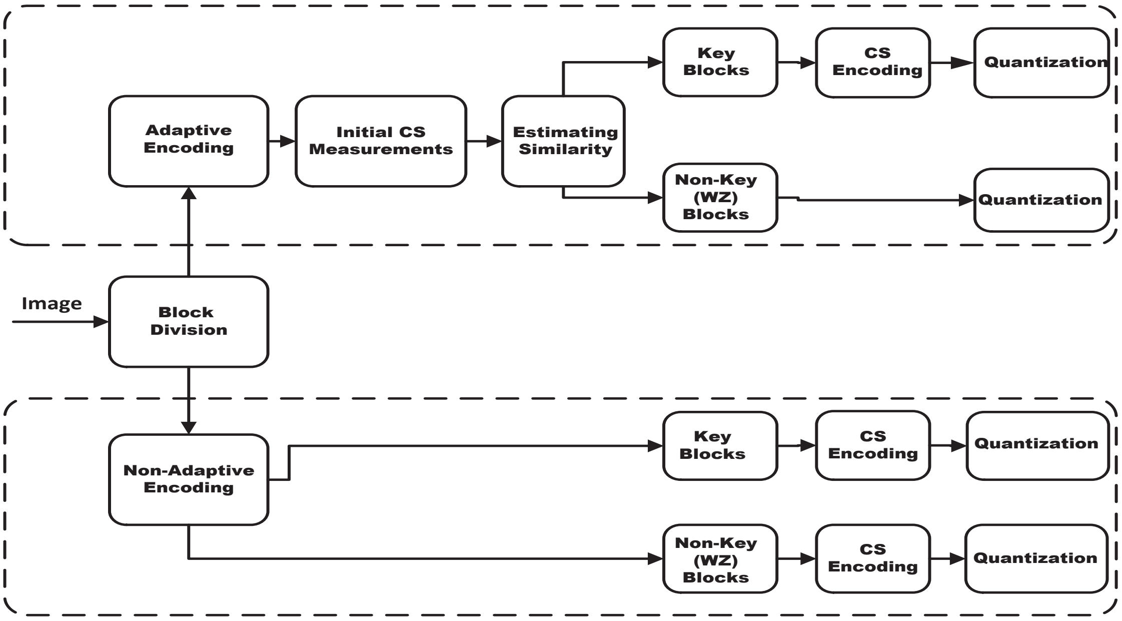
### 6.3.1 Encoder

At the encoder, an image with dimension $N \times N$ pixels is divided into non-overlapping blocks of $B \times B$ pixels. Let $x_i$ represent the data for i-th image block with CS measurements $y_i = \Phi_B x_i$. Each image block is classified either as a key block or a non-key (WZ) block. They are encoded using CS measurement rates of $M_k$ and $M_w$ respectively, where $M_k \gg M_w$. The CS measurements are then quantized using the quantization scheme proposed in Section 4.3.2 for transmission.

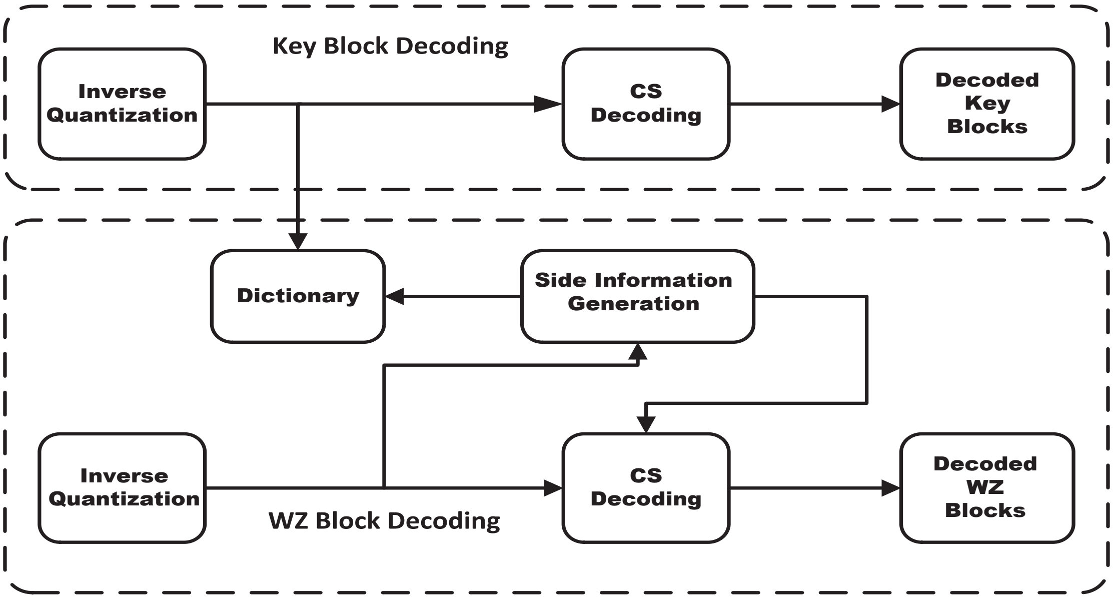
The way by which the image blocks are classified affects the compression ratio as well as the quality of the reconstructed image. The number of CS measurements required to reconstruct a signal is dependent on the sparsity of the signal. If a block has high sparsity then it can be reconstructed with fewer measurements compared with a block with low sparsity. If a low sparsity block is encoded with a low measurement rate, the reconstruction quality will be poor. Determining the sparity of an image block is a challenging problem. Two different block classification schemes are proposed here.

#### Non-Adaptive Block Classification

The non-adaptive block classification scheme is designed for very low complexity encoding with minimum computation. In this scheme, the blocks are classified sequentially according to order position in which they are acquired. A group of consecutive $M$ blocks is referred to as Group of Blocks (GOB). The first block in a GOB is a key block. The remaining $M-1$ blocks are WZ blocks. Block sparsity is not considered in this scheme. It is similar to the concept of Group of Picture (GOP) used in video coding. If $M = 1$, then every block is a key block and is encoded at the same rate. The average measurement rate ($M_{av} = \frac{M_k+(M-1)\times M_w}{M}$)decreases as $M$ increases.



6.5.a: Encoder



6.5.b: Decoder

Figure 6.5: Proposed Distributed Intra Image Codec

### Adaptive Block Classification

The adaptive block classification scheme can be deployed for encoders with more computing resources. This scheme requires the creation of a dictionary of CS measurements of key blocks. This dictionary is initially empty. The first block of the image is always a key block. The CS measurements of this block become the first entry of the dictionary. For each subsequent image block, the MSE between its CS measurements and the entries in the dictionary is computed. If any of these MSE's computed falls below a threshold $\lambda$, then this block is classified as a WZ block and will be encoded at the lower rate. Otherwise, it is a key block and its measurements are added to the dictionary as a new entry. In order to reduce the resources required, the number of entries in the dictionary can be limited to a certain fixed value. When the dictionary is full and a new entry is needed, the oldest entry will be discarded.

Determining an appropriate value for $\lambda$ is a tricky problem. In Figure 6.4, it has been shown that different images has different percentage of similar blocks for the same threshold value. A common approaches are based on standard deviation (STD) [70, 71]. The main disadvantage of these approaches is that the measurements of all the image blocks need to be obtained before the total STD can be computed. Here, a more effective adaptive threshold method is proposed. For each image block $i$, the threshold is obtained by

$$\lambda_i = C \cdot median(|y_i|) \tag{6.2}$$

where $y_i$ are the CS measurements of block $i$ and $C$ is a constant. The value of $C$ controls the relative number of key blocks and WZ blocks. If $C > 1$, the number of WZ blocks will be increased. Choosing $C < 1$ will increase the number of key blocks. This method is easy to compute and experimental results in Section 6.4 will show that it is very effective in determining the correlation of blocks. Furthermore, it reduces the encoding delay compared with STD based methods and therefore is more practical for real world applications.

The entire encoding process is summarized in Algorithm 2.

**Algorithm 2** Adaptive CS Encoding

```
Input:$y_w, D$
for each column $i$ $inD$ do
  Calculate $r[i] = MSE(y_w, D)$
end for
Calculate $\tau = min[r]$
Estimate $\lambda = C.median(|y_w|)$
if $\tau > \lambda$ then
  $y_k$ (Key Block)
else
  if $\tau < \lambda$ then
    $y_w$ (Wz Block)
  end if
end if
```

### 6.3.2 Decoder

At the decoder, after inverse quantization, the key blocks are reconstructed from their own CS measurements. For WZ blocks, reconstruction is performed with the help of side information which is generated through a dictionary. A similar SI generation method to the one described in Chapter 5 is used here.

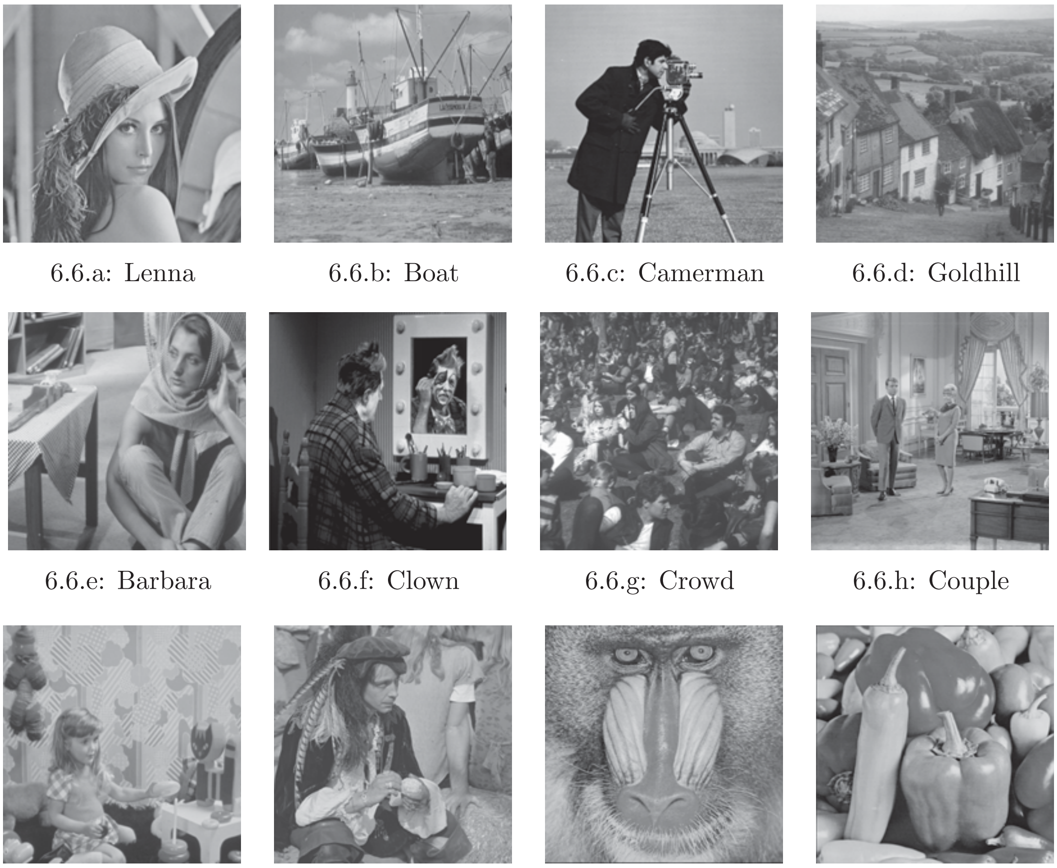
It has been shown in Section 6.2 that MSE values between CS measurements of image blocks are highly correlated. Therefore we can directly make use of the CS measurements of the key blocks as SI. Starting with an empty dictionary $D$, it is populated with the (inverse quantized) CS measurements of the key blocks as they are received. With non-adaptive block classification, each key block is followed by one or more WZ blocks. For the WZ blocks in the same GOB, $D$ has only one entry $D_1$. So $D_1$ is used as the SI to reconstruct this WZ block. For the WZ blocks in the next GOB, the dictionary will have two entries from the two key blocks received so far. The entry in the dictionary that is chosen as SI is the one that has the minimum MSE with the current WZ block. This process continues until all the WZ blocks are reconstructed. With adaptive block classification, the WZ blocks are reconstructed in a similar way. The only difference is that the number of WZ blocks following each key block is not fixed.

There are several advantages with this SI generation method. Firstly, it does not require the key blocks to be decoded first. Secondly, the dictionary only consists of CS measurements of key blocks, which are already available for decoding. So it does not need to be computed or learnt. Most importantly, SI obtained in this way can be directly used by the CS reconstruction algorithm without any further processing.

## 6.4 Experimental Results

Twelve test images [129] are used to evaluate the effectiveness of the proposed distributed CS image codec. They are shown in Figure 6.6. These images are 512 × 512 pixels in size and contain different content and textures. Visual reconstruction quality of the reconstructed images is evaluated by the peak signal to noise ratio (PSNR) and the structural similarity index(SSIM). Compression efficiency of the codec is evaluated by bits per pixel (bpp). The average measurement rate is the average including both key and WZ blocks. In Section 6.2, it has been shown that smaller block sizes yield a larger percentage of correlated blocks. It has also been shown in Section 6.1.2 that in general block size does not effect performance in CS coding. Therefore, in these experiments a block size of 8 × 8) pixels is used.

Scrambled Block Hadamard Ensemble (SBHE) sensing matrix is used to obtain the CS measurements and Daubechies 9/7 wavelets are used as the sparsifying matrix. The SpaRSA [36] algorithm is used as the reconstruction algorithm at the decoder. The proposed adaptive and non-adaptive approach with SI and without SI at the decoder is compared with two other techniques. The first one is the encoding of image blocks with the same measurement rate, i.e. no distinction of key and WZ blocks. It is labelled as "Fixed Rate" in the results. The second technique is first proposed in [70] and recently used to classify image blocks as compressible or incompressible [71]. It is labelled as "STD approach" in the results. The idea is that if the STD of the CS measurements of an image block is higher than STD of the full image, then the block is classified as incompressible and more measurements are required for successful reconstruction. [70] used the original image data to compute the STD, which is not possible in real world



6.6.a: Lenna 6.6.b: Boat 6.6.c: Camerman 6.6.d: Goldhill

6.6.e: Barbara 6.6.f: Clown 6.6.g: Crowd 6.6.h: Couple

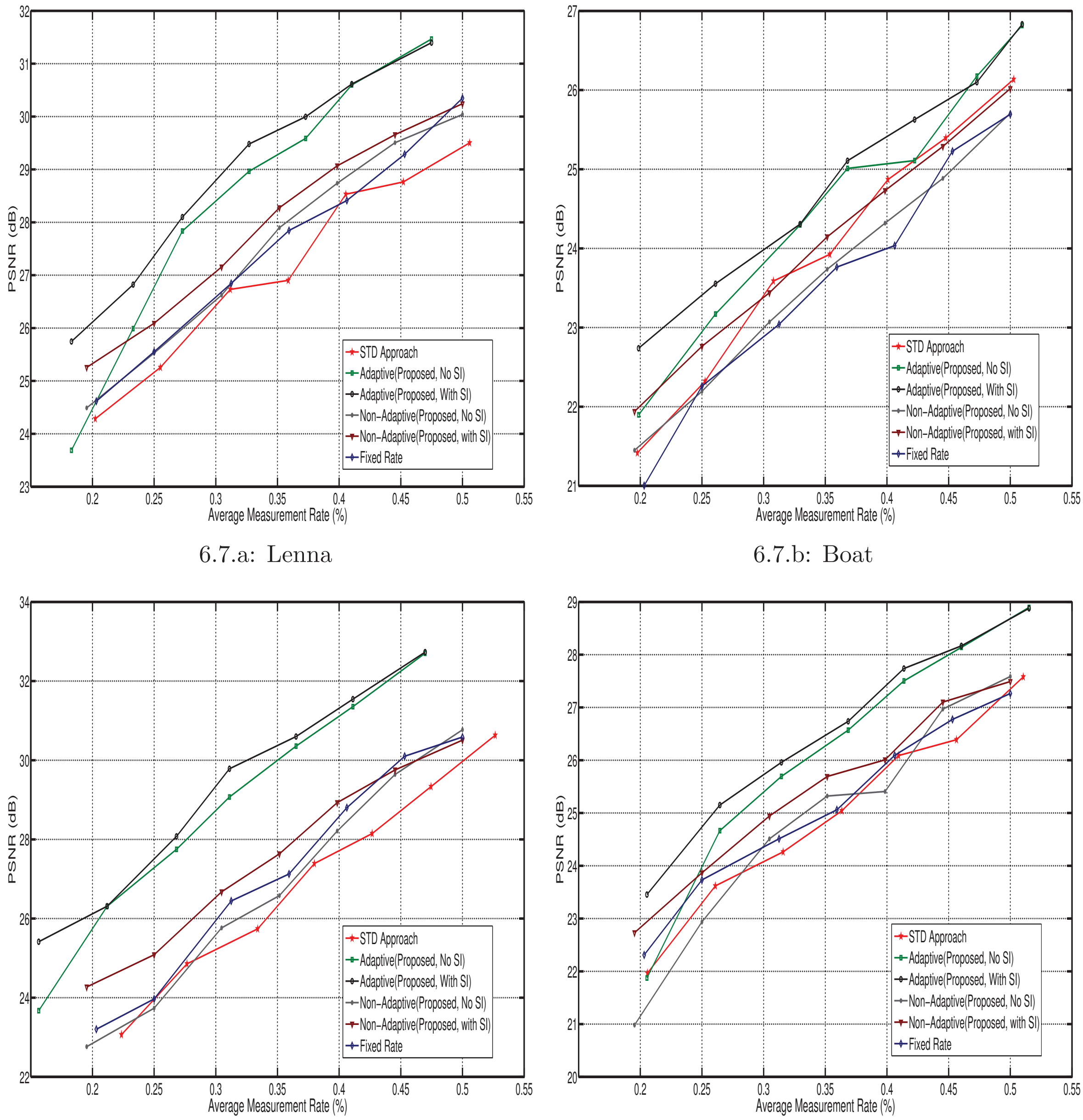
6.6.i: Girl 6.6.j: Man 6.6.k: Mandrill 6.6.l: Peppers

Figure 6.6: Test Images used.

applications. Details of this technique have been discussed in detail in Section 3.1.4.For adaptive block selection, $C = 1$ is used. All codecs are coded in MATLAB R2012b and simulations run on an Intel i5 3.6GHz, Windows 7 Enterprise Edition, 64-bit Operating System with 4GB RAM. For each measurement rate per image, the experiment is run 5 times and then the average is reported. Figure 6.7 shows the R-D curves for four test images (“Lenna”, ”Boat”, “Cameraman” and “Couple”). The proposed adaptive sensing scheme with and without SI at decoder outperforms all other algorithms for all test images. The non-adaptive scheme without SI is comparable with “Fixed Rate” and “STD Approach”. The non-adaptive scheme with SI improves the reconstruction quality and produces better results than “Fixed Rate” and “STD Approach”. Visual reconstruction quality evaluation for above four images on the basis of structure similarity index [124]

Table 6.1: Performance Evaluation of Test Images

| **Image** | | **STD Approach** | **Fixed Rate** | **Non-Adaptive (No SI)** | **Non-Adaptive (SI)** | **Adaptive (No SI)** | **Adaptive (SI)** |
|---|---|---|---|---|---|---|---|
| **Boat** | MR | 38% | 35% | 35% | 35% | 35% | 35% |
| | PSNR | 24.87 | 25.36 | 25.14 | 25.60 | 26.27 | **26.62** |
| | SSIM | 0.81 | 0.82 | 0.81 | 0.83 | 0.83 | **0.84** |
| **Barbara** | MR | 35% | 35% | 35% | 35% | 37% | 37% |
| | PSNR | 23.95 | 23.57 | 23.62 | 24.05 | 24.64 | **24.89** |
| | SSIM | 0.81 | 0.81 | 0.81 | 0.83 | 0.84 | **0.85** |
| **Camera man** | MR | 38% | 35% | 35% | 35% | 31% | 31% |
| | PSNR | 27.02 | 27.18 | 26.78 | 27.55 | 28.74 | **29.21** |
| | SSIM | 0.87 | 0.88 | 0.87 | 0.89 | 0.89 | **0.90** |
| **Couple** | MR | 36% | 35% | 35% | 35% | 36% | 36% |
| | PSNR | 24.99 | 25.11 | 24.82 | 25.40 | 26.19 | **26.58** |
| | SSIM | 0.80 | 0.81 | 0.80 | 0.82 | 0.83 | **0.84** |
| **Clown** | MR | 33% | 35% | 35% | 35% | 37% | 37% |
| | PSNR | 26.84 | 26.01 | 26.06 | 26.89 | 26.66 | **28.09** |
| | SSIM | 0.82 | 0.84 | 0.85 | 0.87 | 0.86 | **0.89** |
| **Crowd** | MR | 32% | 35% | 35% | 35% | 37% | 37% |
| | PSNR | 25.83 | 25.00 | 24.51 | 25.24 | 26.20 | **26.79** |
| | SSIM | 0.84 | 0.85 | 0.84 | 0.86 | 0.88 | **0.89** |
| **Girl** | MR | 37% | 35% | 35% | 35% | 31% | 31% |
| | PSNR | 27.76 | 28.45 | 28.04 | 28.73 | 28.55 | **29.20** |
| | SSIM | 0.83 | 0.82 | 0.82 | 0.84 | 0.82 | **0.83** |
| **Goldhill** | MR | 34% | 35% | 35% | 35% | 37% | 37% |
| | PSNR | 26.68 | 27.01 | 26.76 | 27.07 | 26.53 | **27.89** |
| | SSIM | 0.79 | 0.81 | 0.81 | 0.82 | 0.82 | **0.84** |
| **Lena** | MR | 36% | 35% | 35% | 35% | 32% | 32% |
| | PSNR | 27.14 | 27.56 | 27.55 | 27.96 | 28.30 | **28.88** |
| | SSIM | 0.86 | 0.87 | 0.87 | 0.88 | 0.88 | **0.89** |
| **Man** | MR | 36% | 35% | 35% | 35% | 36% | 36% |
| | PSNR | 25.84 | 25.98 | 25.59 | 26.00 | 26.70 | **27.26** |
| | SSIM | 0.81 | 0.82 | 0.81 | 0.82 | 0.84 | **0.85** |
| **Mandrill** | MR | 35% | 35% | 35% | 35% | 41% | 41% |
| | PSNR | 21.01 | 21.07 | 21.10 | 21.12 | 21.96 | **22.18** |
| | SSIM | 0.68 | 0.68 | 0.69 | 0.69 | 0.75 | **0.75** |
| **Peppers** | MR | 33% | 35% | 35% | 35% | 33% | 33% |
| | PSNR | 26.83 | 26.45 | 26.60 | 27.48 | 27.59 | **28.29** |
| | SSIM | 0.88 | 0.87 | 0.88 | 0.90 | 0.89 | **0.90** |



6.7.a: Lenna

6.7.b: Boat

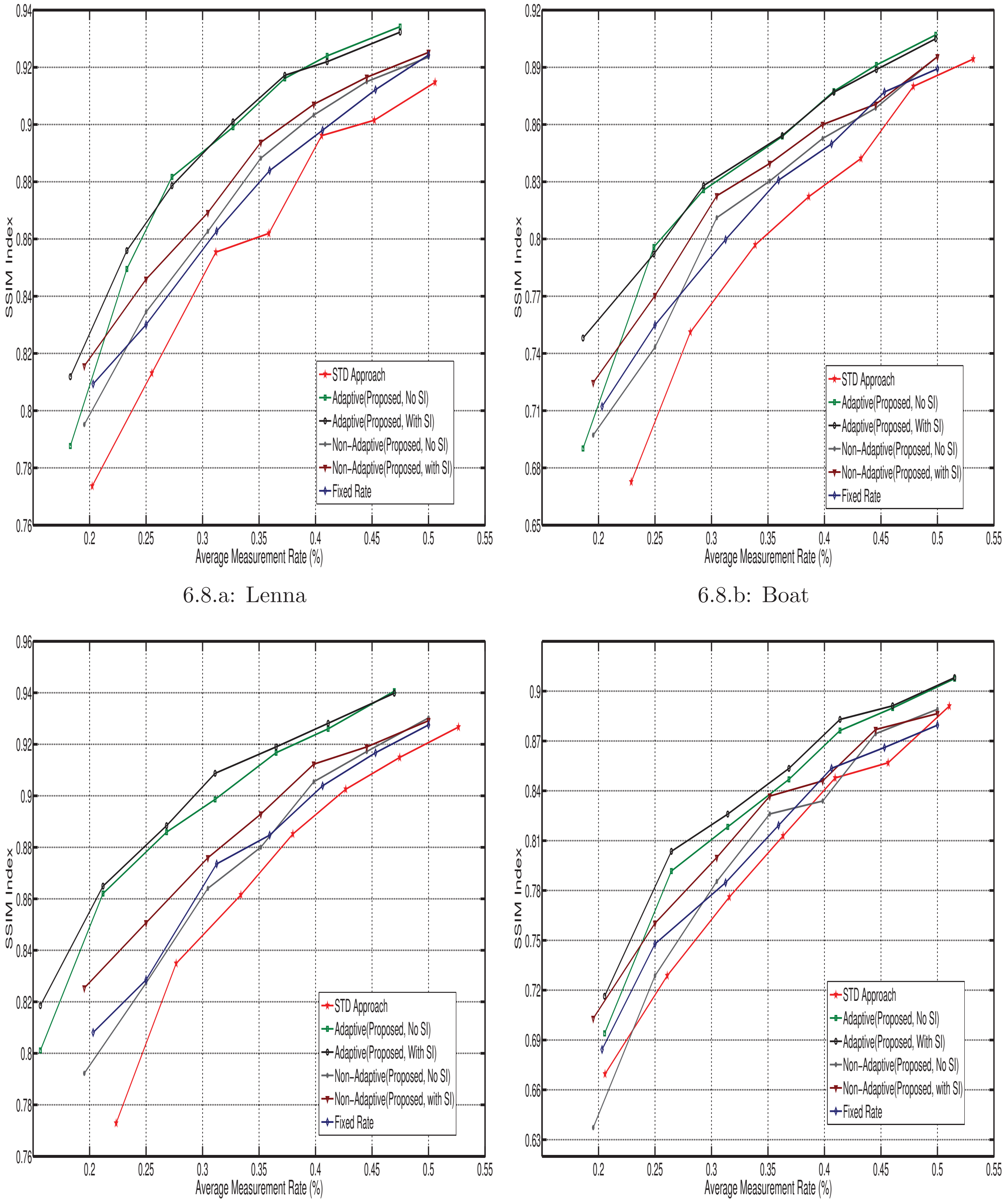
6.7.c: Camerman

6.7.d: Couple

Figure 6.7: Rate-Distortion Performance for Test Images

are shown in Figure 6.8. The visual quality of the adaptive scheme and the non-adaptive scheme with SI is superior for all test images.

Table 6.1 shows R-D performance of all twelve images averaged for all measurement rates. Again, the adaptive scheme outperforms all other schemes. The lowest average



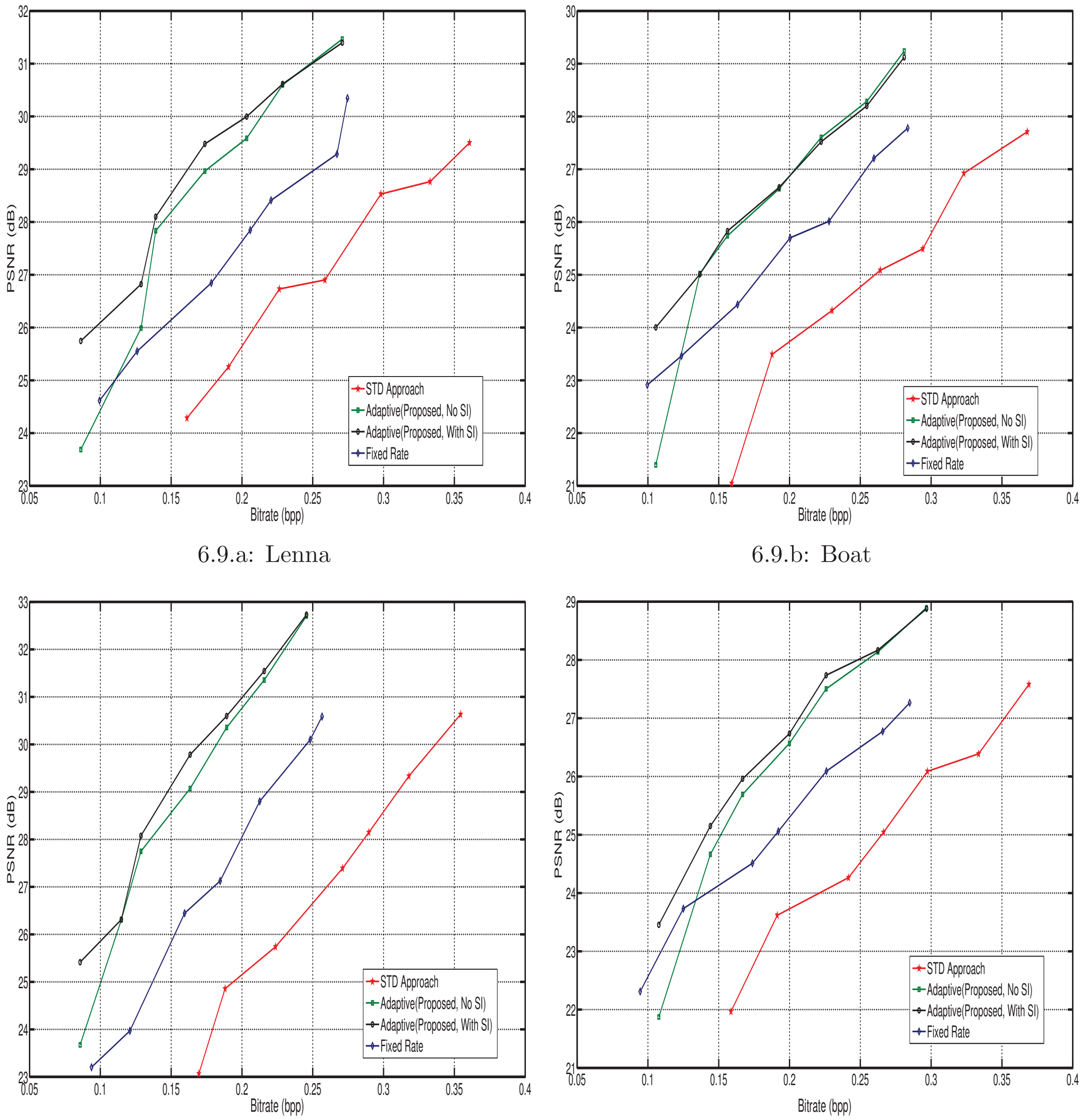
6.8.a: Lenna

6.8.b: Boat

6.8.c: Camerman

6.8.d: Couple

Figure 6.8: SSIM Index for Test Images



6.9.a: Lenna

6.9.b: Boat

6.9.c: Camerman

6.9.d: Couple

Figure 6.9: Compression Efficiency for Test Images

MR is obtained for "Cameraman", "Girl", "Lena" with the adaptive scheme; the PSNR is more than 1.5dB better than other schemes. However, for an image with less intra image similarity like "Mandrill", the adaptive schemes requires a higher number of measurements than other approaches. Using the proposed SI generation method improves reconstruction performance for both adaptive and non-adaptive encoding.

To evaluate the compression efficiency of the proposed distributed image codec, the CS measurements are quantized using the quantization scheme developed in Section 4.3.2. Each CS measurement is allotted 8 bits for quantization. The quantized CS measurements are then entropy coded using Huffman coding. Figure 6.9 shows the PSNR performance of four test images at different bit rates.

The compression efficiency of the adaptive scheme with SI is higher than "Fixed Rate" and "STD" schemes at the same reconstruction quality. At lower average measurement rates, the "Fixed Rate" scheme performs better than the adaptive scheme with no SI for images "Boat" and "Couple".

## 6.5 Summary

In this chapter, a new distributed CS image codec has been proposed. It includes two different block classification schemes. The non-adaptive block classification scheme is for low complexity, resource constrained encoders. The adaptive alternative is for more resourceful encoders as it involves block similarity comparisons. At the decoder, a simple side information generation technique is proposed that is based on the MSE comparison of CS measurements between image blocks. The proposed SI technique is simple to implement yet very effective in improving the reconstruction quality of WZ blocks. It can easily be integrated with any CS reconstruction algorithm without need of modification. The proposed codec has been shown empirically to outperform "Fixed Rate" and STD based approaches for all the test images.

# Chapter 7

# Distributed Block-based Video Compressed Sensing

In Chapter 5, a relationship between CS measurements of adjacent video frames has been established. It is then used to produce the side information for improving the coding efficiency of the proposed distributed video codec. In Chapter 6, this idea is extended to image compression with different sampling rates to blocks within an image. In this chapter, the idea developed in Chapter 5 for frame based CS measurements is extended to block-based CS measurements of a video frame with adaptive encoding. The proposed video codec exploits both intra-block and inter-block correlation in video frames. Section 7.1 presents the analysis of intra-frame and inter-frame block correlation. This analysis forms the basis for the block-based video codec proposed in Section 7.2. Section 7.3 presents some experimental evaluation of proposed video codec with comparisons with other CS-based conventional and distributed video coding methods discussed in Chapter 3 in terms of visual reconstruction quality, reconstruction time complexity and compression efficiency.

## 7.1 Block Correlation Analysis

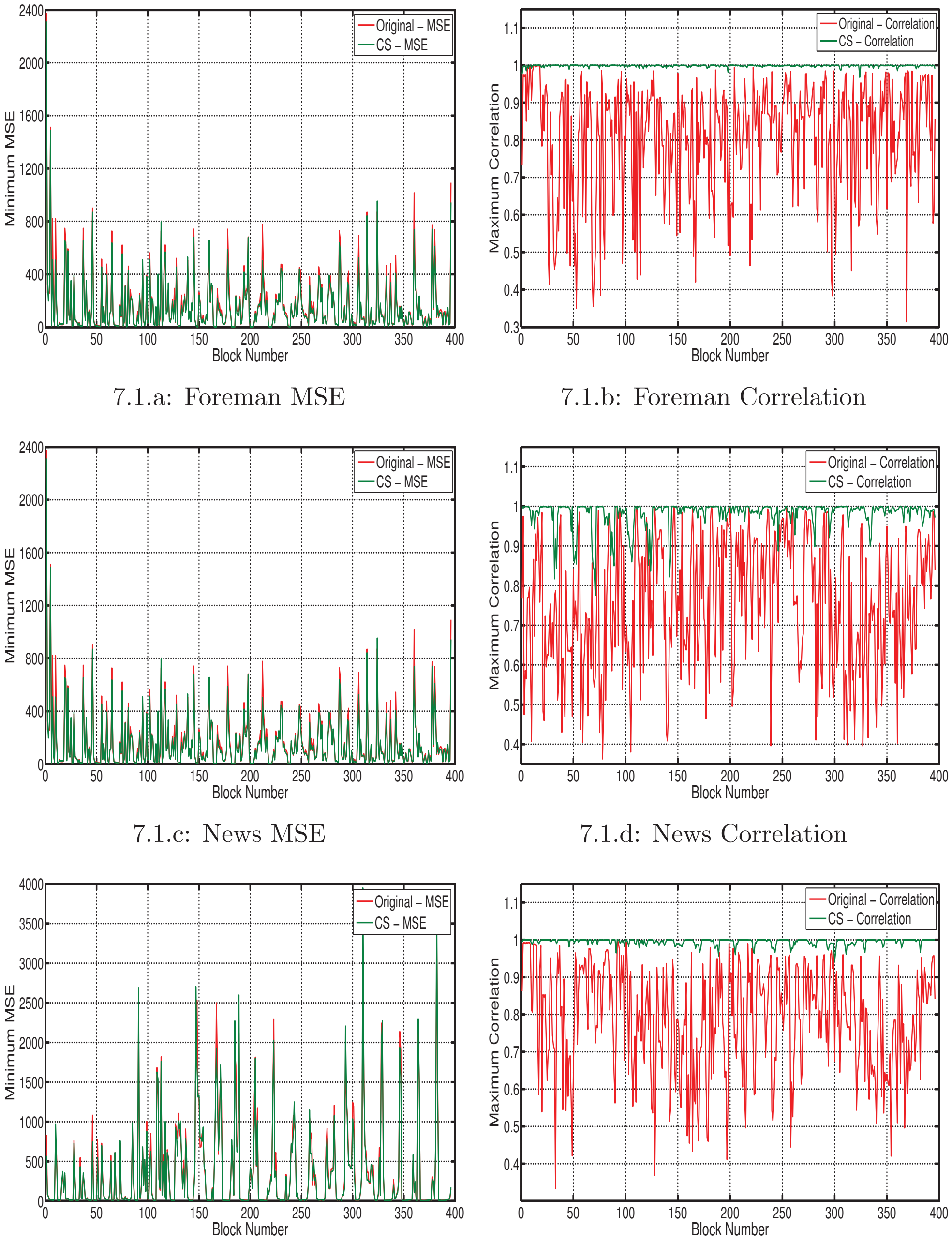
A study of the correlation between blocks both within a single frame and between adjacent frames of a video are presented here.

### 7.1.1 Intra-frame Block Correlation

When an image is divided into non-overlapping blocks, generally there is a high correlation between neighbouring image blocks. This high correlation can be exploited to estimate the similarity among image blocks. If the block size is small, this correlation is high. If the data of the image blocks are correlated, then their CS measurements will also be correlated. This correlation has been used to design an effective strategy to reduce the measurement rate of correlated blocks in Chapter 6. A similar analysis is performed for video frames here.

Three QCIF videos “Foreman”, “’News” and “Hall Monitor” are used for these experiments. Only the first frame of these videos are used in the following analysis. A video frame is divided into non-overlapping blocks and CS measurement of each block is obtained using SBHE. The measurement rate of each block is fixed at 50%. Intra-frame block similarity is measured with both the original pixel data and their CS measurement using the correlation coefficient and Mean Square Error (MSE). Correlation between each image block and all other image blocks in the frame are computed and then the maximum correlation coefficient and the minimum MSE values are plotted. Figure 7.1 shows the results for a block size of $8 \times 8$ pixels. In terms of MSE, CS measurements and original pixel data aligns very well with each other. This indicates that the CS measurements alone can be used to predict image block similarity. The correlation coefficients obtained using CS measurements are generally very high and do not relate well with those obtained using the original pixel data. These results suggest that the MSE values calculated using CS measurements should be used to predict the similarity between image blocks in a video frame.

To evaluate the effect of block size on block correlation, the percentage of correlated blocks exceeding an MSE threshold is computed for four different block sizes of $64 \times 64$, $32 \times 32$, $16 \times 16$ and $8 \times 8$ pixels. Figure 7.2 shows the results of the previous three videos plus “Coastguard”. The smallest block size of $8 \times 8$ produces more correlated blocks than all the other block sizes. Blocks with a size of $64 \times 64$ and $32 \times 32$ can virtually



7.1.a: Foreman MSE

7.1.b: Foreman Correlation

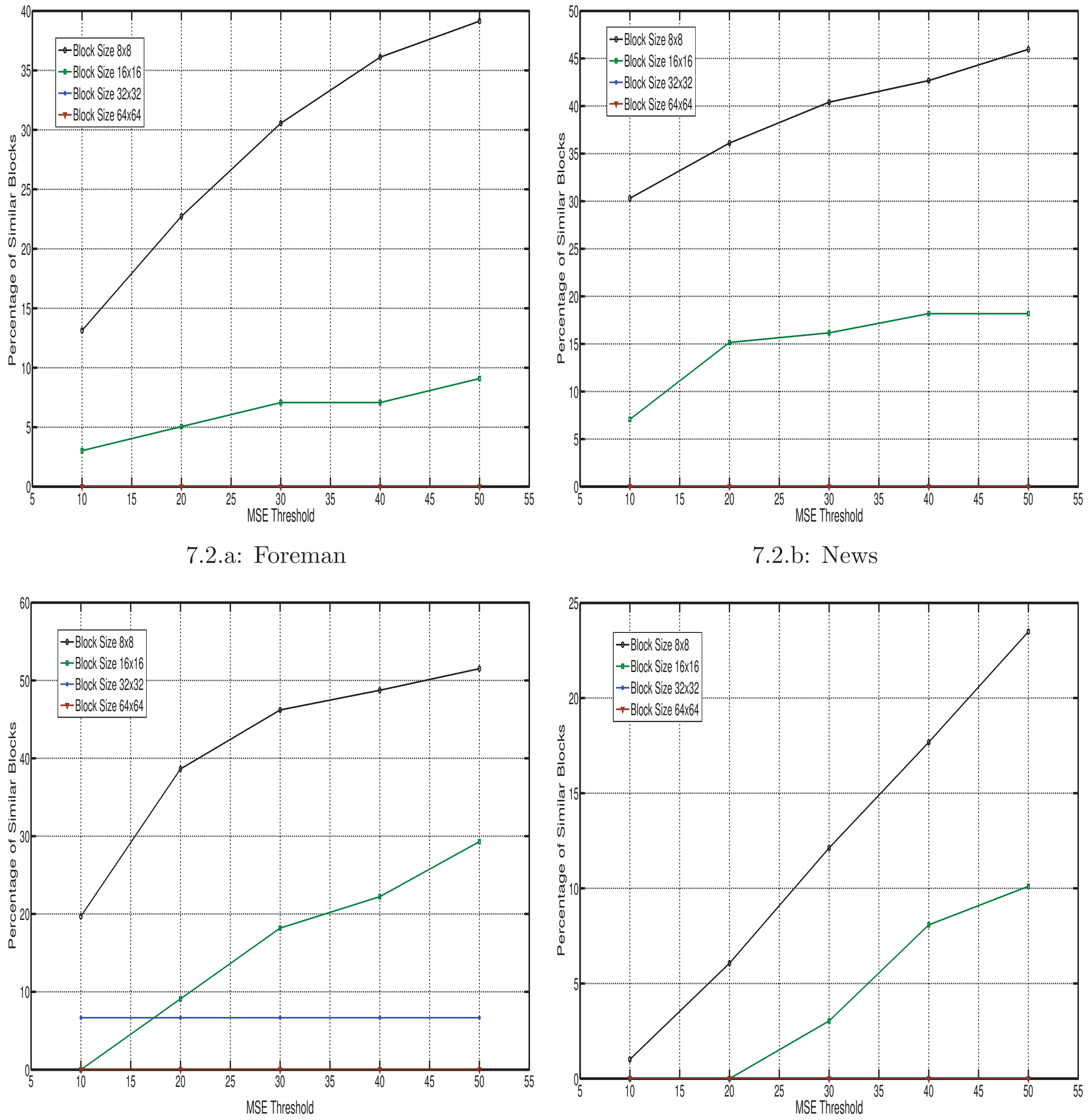
7.1.c: News MSE

7.1.d: News Correlation

7.1.e: Hall Monitor MSE

7.1.f: Hall Monitor Correlation

Figure 7.1: Intra Block Correlation of Original Pixel Data and CS measurements with MSE and Correlation Coefficient



7.2.a: Foreman

7.2.b: News

7.2.c: Hall Monitor

7.2.d: Coastguard

Figure 7.2: Percentage of Similar Blocks for Block Size (64, 32, 16 and 8)

be considered independent in all videos except "Hall Monitor". This result also suggests that the percentage of blocks that can be considered correlated depends on the MSE threshold. If the threshold is set too high, then most of the blocks in the video frame will be considered correlated. An appropriate threshold will need to be determined when

Intra-frame block correlation is applied. In the proposed block based video codec, an adaptive threshold is used to determine intra-frame block correlation.

### 7.1.2 Inter-frame Block Correlation

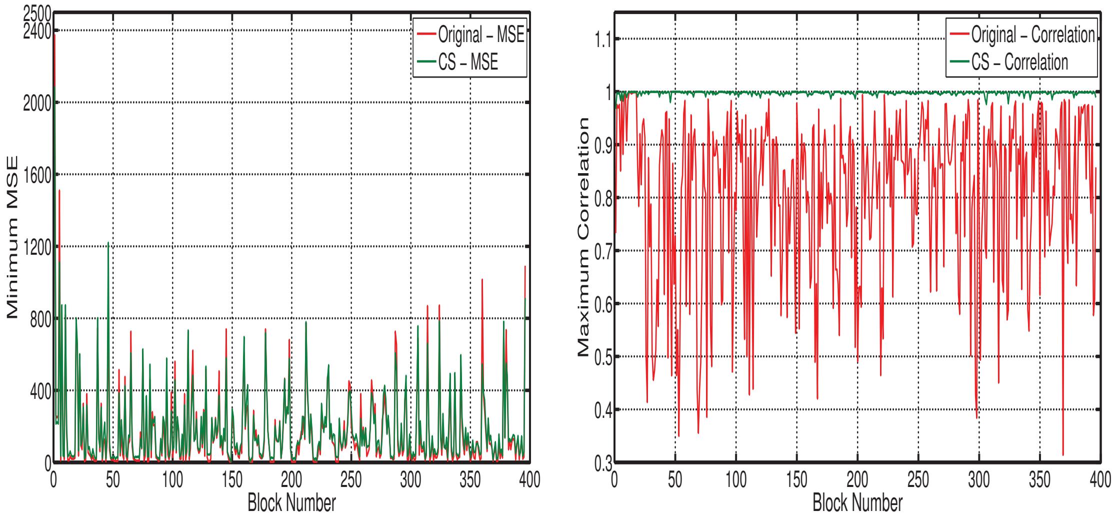
In a typical video sequences, adjacent frames are highly correlated and this fact has been exploited in traditional video codecs like MPEG and H.264 through motion estimation and motion compensation (ME/MC). In Chapter 5, this inter-frame correlation in the CS measurement domain using full video frame measurements was used in the video codec proposed there. Here, the correlation between blocks in adjacent video frames is analyzed. This correlation can be used to design a effective strategy to further reduce the sampling rate.

Two video sequences "Foreman" and "Coastguard" are used for the experiments in this Section. A GOP size of 3 is used. The first frame in a GOP is considered a key frame and its correlation with the two subsequent frames is computed. Each video frame is divided into non-overlapping blocks and CS measurements of each block are obtained using SBHE. The measurement rate is fixed at 50%. Correlation coefficients and MSE are computed with both the original pixel data and CS measurements.

Figure 7.3 and Figure 7.4 show the maximum correlation coefficients and minimum MSE values for the "Foreman" video. The results show that MSE values are better predictors of inter-frame block correlation than the correlation coefficients. This is quite different from the results obtained from frame-based CS measurements in Section 5.1.3 which shows that both MSE and correlation coefficient are good predictors. reason is that Figure 7.5 and Figure 7.6 show similar results for the "Coastguard" video sequence.

## 7.2 Proposed Adaptive Block-based Video Codec

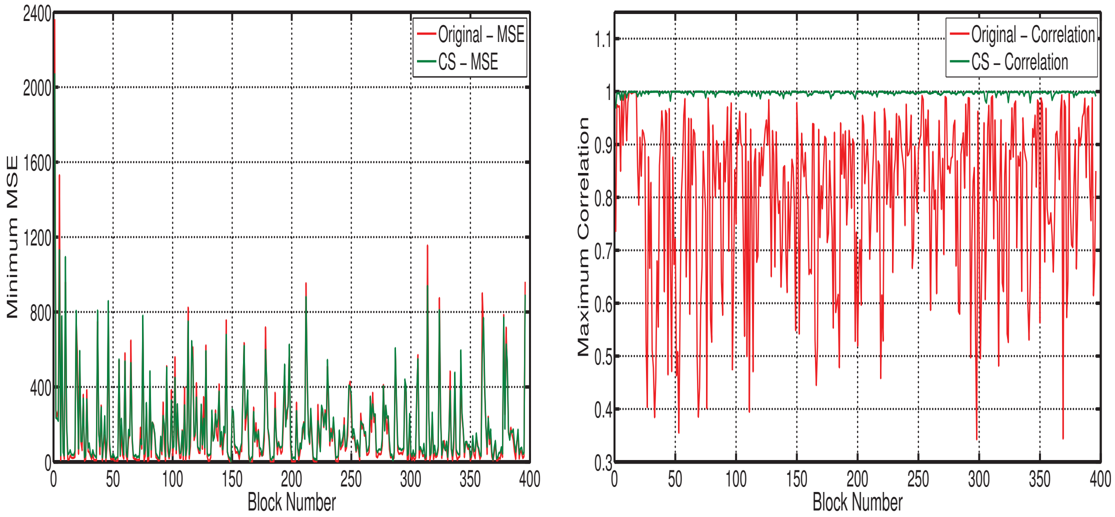
Based on the block correlations established in Section 7.1, a distributed block-based video coded is proposed. The block diagram of this codec is shown in Figure 7.7.



7.3.a: Foreman with MSE

7.3.b: Foreman with Correlation Coefficient

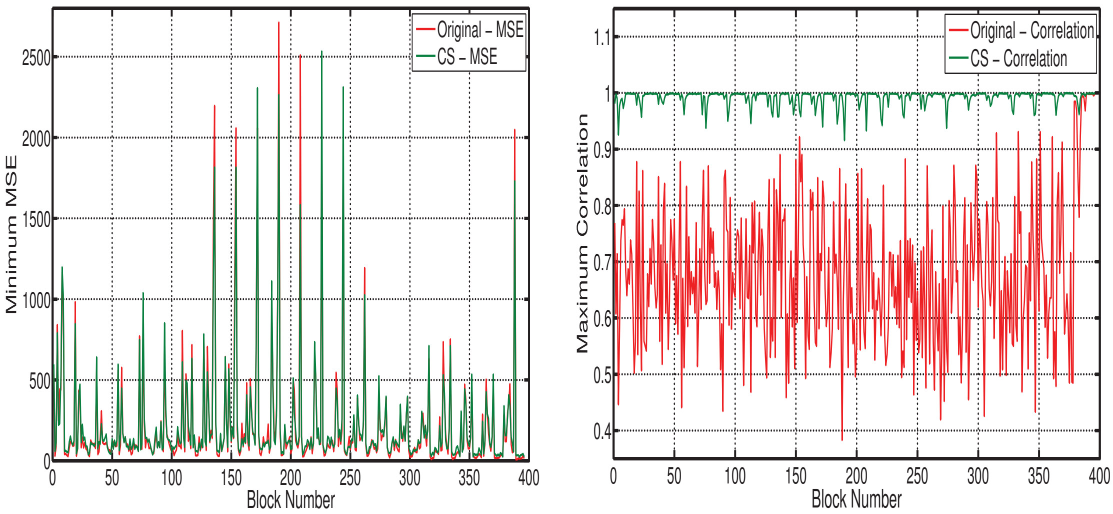
Figure 7.3: Inter Block Correlation of Original Pixel Data and CS measurements between 1st and 2nd frame of Foreman video with MSE and Correlation Coefficient



7.4.a: Foreman with MSE

7.4.b: Foreman with Correlation Coefficient

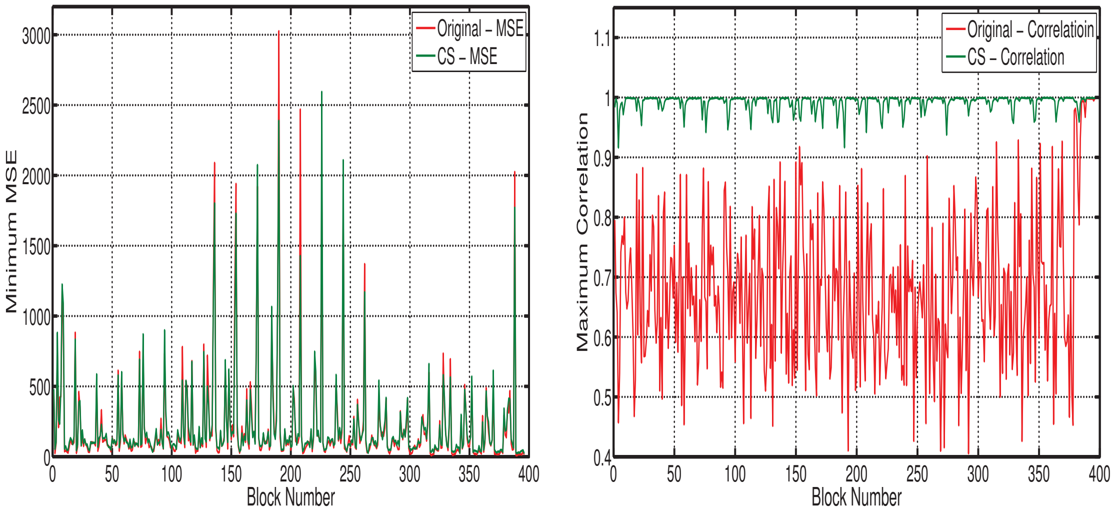
Figure 7.4: Inter Block Correlation of Original Pixel Data and CS measurements between 1st and 3rd frame of Foreman video with MSE and Correlation Coefficient



7.5.a: Coastguard with MSE

7.5.b: Coastguard with Correlation Coefficient

Figure 7.5: Inter Block Correlation of Original Pixel Data and CS measurements between 1st and 2nd frame of Coastguard video with MSE and Correlation Coefficient



7.6.a: Coastguard with MSE

7.6.b: Coastguard with Correlation Coefficient

Figure 7.6: Inter Block Correlation of Original Pixel Data and CS measurements between 1st and 3rd frame of Coastguard video with MSE and Correlation Coefficient

Each key frame is reconstructed from its block CS measurements. For each block in a Non-key frame, first a side information is generated for key blocks using Inter correlation and for Non-key blocks using both Inter and Intra correlation. Encoder and decoder details are described below.

### 7.2.1 Encoder

Video frames are grouped into GOPs consisting of a key frame followed by a number of non-key (WZ) frames. In Chapter 5 full frame measurements were obtained for both key and WZ frames. In the current scheme, a block-based sensing strategy is used to further lower the complexity of the encoder. Each video frame is divided into non-overlapping blocks of equal sizes. CS measurements are obtained using Structurally Random Matrices with SBHE as the sensing matrix. The blocks in a key frame are encoded with a higher measurement rate than those in WZ frames.

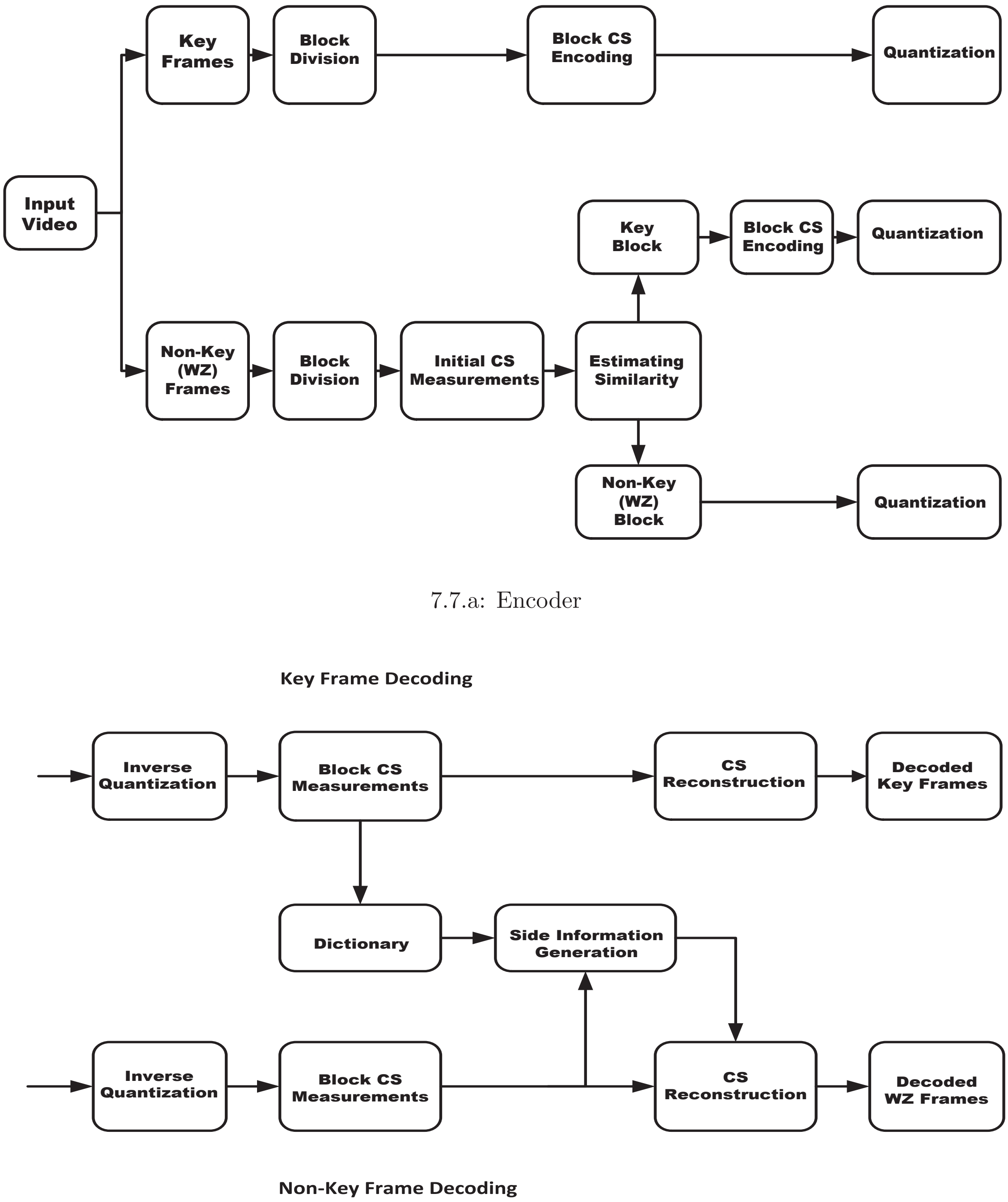
#### Key Frame Encoding

Each key frame $X_k$ is divided into $B$ non-overlapping blocks of $N_B \times N_B$ pixels. Let $x_i^k$ represents the $i$-th block in this key frame, then its CS measurements $y_i^k = \Phi_B x_i^k$ are obtained by using a block-based sensing matrix $\Phi_B$. Every block in a key frame is sampled at a fixed $M_k$ which is higher than that for the blocks in a WZ frame. After sampling, they are quantized using the quantization scheme proposed in Section 4.3.2.

#### WZ Frame Encoding

Each WZ frame $X_w$ is divided into non-overlapping blocks in the same way as key frames. As shown in Section 7.1.1, blocks within a video frame are correlated and this correlation can be used to reduce the sampling rate.

Blocks within a WZ frame are classified either as a key block or a non-key block. This decision is made depending on the correlation between the current block and a dictionary



7.7.a: Encoder

7.7.b: Decoder

Figure 7.7: Proposed Distributed Block-based Video Codec

of all key blocks processed. This dictionary $D$ is initially empty. The first block $x_1^w$ is always considered as a key block and $M_{kw}$ measurements are acquired. Each subsequent image block is initially sampled at a rate $M_{ww}$ such that $M_{ww} << M_{kw}$. Then the MSE values between current block and the entries in $D$ are computed. If the minimum computed MSE falls below a threshold $\lambda$, then this block is classified as a WZ block and no more processing is required for that block. Otherwise, it is considered a key block, and an additional $M_{kw} - M_{ww}$ measurements are acquired. These additional measurements can be obtained from the $(M_{ww}+1)$-th to the $(M_{kw})$-th rows of the sensing matrix. The measurements of this block are then added to $D$. To further lower the complexity, the maximum size of the dictionary can be limited. The CS measurements are then quantized using the quantization scheme proposed in Section 4.3.2.

A similar approach to that used in Section 6.3.1 is adopted to determine an effective threshold $\lambda$. This threshold is an adaptive one which depends on the median value of the CS measurements for that individual block. For each image block $i$, the threshold is given by

$$\lambda_i = C \cdot median(|y_i|) \tag{7.1}$$

where $y_i$ are the CS measurements of block $i$ and $C$ is a constant. The value of $C$ controls the relative number of key blocks and non-key blocks. If $C > 1$, the number of WZ blocks will be increased. Choosing $C < 1$ will increase the number of key blocks. This is a much simpler and more effective way to classify blocks compared to the STD approaches used in [70, 71]. Furthermore, it reduces the encoding delay and therefore is more practical for real world applications.

### 7.2.2 Decoder

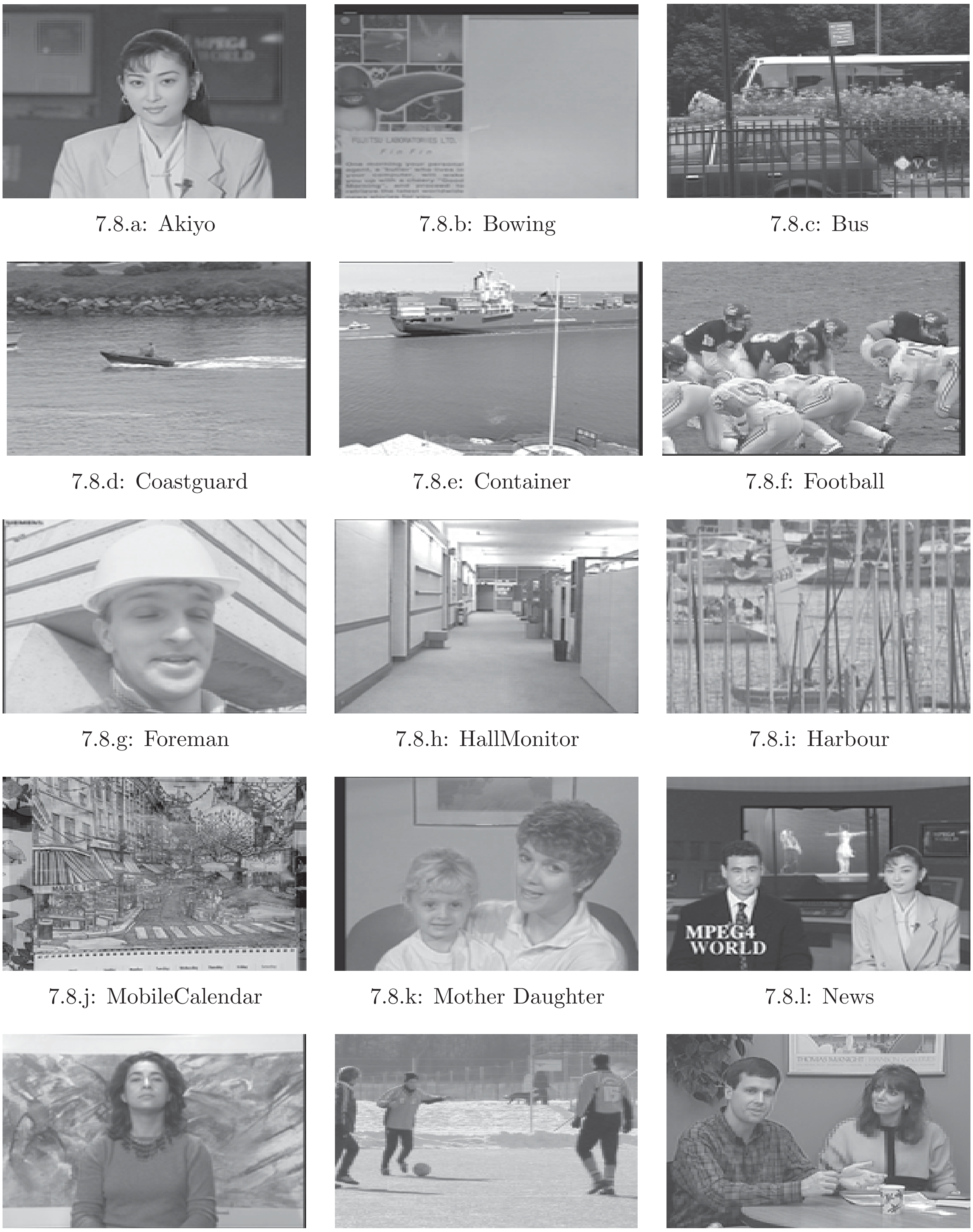
The blocks in a key frames are decoded entirely based on their own CS measurements. These CS measurements first inverse quantized and then reconstructed using a suitable CS reconstruction algorithm. For blocks in WZ frames, reconstruction is performed with the help of side information which is generated through a dictionary $D$. Initially, $D$ consists of the CS measurements of the blocks from the key frame, denoted by $y_i^k$, $i = 1, 2, \ldots, B$. To generate side information for a key block in a subsequent WZ frame with measurements $y_k^w$, the MSE values between $y_k^w$ and all the entries in $D$ that are within a search window of blocks that are adjacent to the current block are computed. The side information is the $M_{SI}$ measurements of the entry in $D$ which yields the minimum MSE, where $M_{SI} = M_k - M_{kw}$. The measurements that are used to reconstruct this key block are its own $M_{kw}$ measurements plus the $M_{SI}$ measurements of the side information. In this way, the measurement rate of block is increased to the same as that for a key frame. After that, the current key block measurements are added to $D$ to be used to account for intra-frame block correlation.

For a WZ block, the process is similar. By now $D$ would have the measurements of some of the key blocks in the current WZ frame. When the decoding of the current WZ frame is complete, existing key block measurements of this WZ frame will be discarded from $D$. This process is followed for all WZ frames in a GOP. $D$ can be limited to consists of only the key frame blocks in a single GOP, or those in some of the previous GOP can be maintained as well.

This is a simple and effective SI generation method which exploits both intra-frame and inter-frame block correlations. More importantly, side information is generated directly in the measurement domain. Hence it does not require the blocks in a key frame to be decoded first. Furthermore, it can easily be used in conjunction with any CS reconstruction algorithm without requiring changes.

## 7.3 Experimental Results

Figure 7.8 shows a first frame of the video sequences that are used for the evaluation of the distributed block-based video codec in the previous section. These videos are in QCIF format with a frame size of $172 \times 144$ pixels. There are both slow and fast motions in these videos.



7.8.a: Akiyo

7.8.b: Bowing

7.8.c: Bus

7.8.d: Coastguard

7.8.e: Container

7.8.f: Football

7.8.g: Foreman

7.8.h: HallMonitor

7.8.i: Harbour

7.8.j: MobileCalendar

7.8.k: Mother Daughter

7.8.l: News

7.8.m: Silent

7.8.n: Soccer

7.8.o: Students

Figure 7.8: Test Videos

CS measurements of each block in both key or non-key frames are obtained using Structurally Random Matrices with Scrambled Block Hadamard matrix as the sensing matrix. The sparsifying matrix is a wavelet transform matrix based on the Daubechies 9/7 wavelet. Only the luminance (Y) component is used in the experiments.

Three different GOP sizes of 3, 5 and 8 are used in the experiments. The first frame in each GOP is considered a key frame, followed by the respective number of WZ frames. Measurement rates (MR) of the blocks in the key frames range from 30% to 70%. MRs for key blocks in a WZ frame range from 10% to 50% and that for WZ blocks are 5% to 45%. In Section 7.1.1, it has been shown that smaller block sizes yield a larger percentage of correlated blocks. It has also been shown in Section 6.1.2 that in general block size does not effect performance in CS coding. Therefore, in these experiments a block size of $8 \times 8$) pixels is used.

The performance of the proposed codec is compared with three other techniques. All three techniques have been described in Section 3.2.4. The first one uses a multi-hypothesis approach to search for an optimal block in a key frame to represent a current block in a WZ frame [88]. This is achieved by finding multiple, distinct ME/MC predictions for a current block and then combined to yield a composite prediction by finding the optimal solution using Tikhonov regularization. The MATLAB source code of this codec is obtained from [130]. For comparable results, We have also used the same SPL transform in our proposed block based adaptive scheme reconstruction. The other two techniques are the Modified-CS-Residue [93] and the *k*-*t* FOCUSS [78, 94]. Modified-CS-Residue does not employ ME/MC at the decoder but it attempts to explicitly track the sparsity pattern of frame to frame. *k*-*t* FOCUSS uses an iterative recovery with ME/MC of non-key frames from the neighbouring key frames. The MATLAB source codes for Modified-CS-Residual and *k*-*t* FOCUSS are available from [131] and [132] respectively.

Visual reconstruction quality of the reconstructed video sequence is evaluated by the peak signal to noise ratio (PSNR) and the structural similarity index (SSIM). Compression efficiency of the codec is indicated by the bit rate. The average measurement rate for a video is the average including both key and WZ frames. All codecs are coded in MATLAB and simulations run on an Intel i5 3.4GHz, Windows 7 Enterprise Edition, 64-bit Operating System in MATLAB R2012b 64-bit with 4GB RAM. For fair analysis, no other programs were running on the simulation computer except the operating system and MATLAB.

### 7.3.1 Measurement Rate Reduction

In first set of experiment, the effect of constant $C$ in Equation 7.1 is examined. As mentioned in Section 7.2.1, $C$ controls the number of key and non-key blocks in a WZ frame. If $C > 1$, the threshold $\lambda$ will be increased and so there will be more non-key blocks. The reverse will happend for $C < 1$. A suitable threshold is highly dependent on the content of the video frame.

Table 7.1 shows the percentage of CS measurement rate reductions using adaptive encoding for three different values of $C$ and three different measurement rates for a GOP size of 3. It is observed that $C > 1$ generally provides higher measurement rate reductions as the measurement rate increases. $C = 1$ also provides good measurement reductions for all type of videos. For videos consisting of contents which are rich in spatial details ("Bus", "Mobile Calendar" and "Harbour"), measurement rate reductions are very low. This is because intra-frame block correlation is not high in these cases. Also, the higher the difference in measurement rates between key and non-key blocks, the higher will be the reduction. Finding an appropriate $C$ for all video content is a challenging problem. In subsequent experiments, $C = 1$ is used.

### 7.3.2 Reconstruction Complexity Evaluation

In the second set of experiments, reconstruction complexity for different schemes are evaluated by calculating the average reconstruction time (in seconds) for key frames and WZ frames. Figure 7.9 shows average reconstruction time of six video sequences with three different GOP sizes. The proposed codec outperforms all other techniques

Table 7.1: Percentage of Measurement Rate Reduction in Adaptive Encoding

| Video | $M_{kw} = 15\%, M_{ww} = 5\%$ | | | $M_{kw} = 25\%, M_{ww} = 15\%$ | | | $M_{kw} = 35\%, M_{ww} = 25\%$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | **C= 0.5** | **C= 1** | **C= 1.5** | **C= 0.5** | **C= 1** | **C= 1.5** | **C= 0.5** | **C= 1** | **C= 1.5** |
| **Akiyo** | 24.40 | 37.49 | **39.02** | 4.96 | 19.69 | **20.07** | 10.85 | 12.05 | **15.54** |
| **Bowing** | 22.68 | **36.31** | 26.10 | 13.05 | **16.08** | 15.12 | 11.21 | 12.73 | **13.63** |
| **Bus** | 1.14 | 1.55 | **5.85** | 1.69 | 1.93 | **2.73** | 2.85 | **3.98** | 3.15 |
| **Coast guard** | 14.01 | 16.78 | **25.48** | 7.17 | **11.05** | 9.41 | 6.11 | **9.81** | 7.43 |
| **Container** | 31.16 | **35.64** | 32.71 | 18.13 | 20.63 | **20.74** | 14.39 | **16.74** | 15.91 |
| **Football** | 8.39 | **31.52** | 23.60 | 4.40 | 12.56 | **16.10** | 7.02 | 9.19 | **11.68** |
| **Foreman** | 20.42 | 22.85 | **34.96** | 11.10 | 16.35 | **17.52** | 8.49 | 12.06 | **14.00** |
| **Hall Monitor** | 24.67 | **33.41** | 30.51 | 12.89 | 16.89 | **19.20** | 12.16 | 13.76 | **14.50** |
| **Harbour** | 1.65 | **8.12** | 2.39 | 0.12 | **3.79** | 1.36 | 1.93 | **2.47** | 2.22 |
| **Mobile Calendar** | 0.84 | **2.12** | 1.86 | 1.45 | 1.69 | **2.18** | 2.81 | 3.12 | **3.43** |
| **Mother Daughter** | 20.24 | 31.99 | **37.11** | 16.54 | 16.74 | **19.72** | 10.54 | **16.05** | 16.00 |
| **News** | 4.86 | 18.31 | **29.88** | 10.11 | 11.71 | **15.71** | 7.24 | 12.34 | **13.79** |
| **Silent** | 6.79 | 13.12 | **17.98** | 4.29 | 6.11 | **6.70** | 4.44 | 7.60 | **8.50** |
| **Students** | 9.54 | 11.39 | **17.10** | 6.05 | 9.16 | **9.91** | 5.41 | **8.05** | 7.96 |
| **Soccer** | 29.42 | **43.99** | 41.63 | 15.14 | **22.85** | 20.43 | 13.25 | 16.24 | **17.89** |

significantly. Techniques which explicitly uses ME/MC require more reconstruction time for all videos. The multi-hypothesis technique is the slowest due to its use of different ME/MC predictions instead of a single prediction. Reconstruction time for all test videos averaged for all measurement rates are tabulated numerically in Table 7.2.

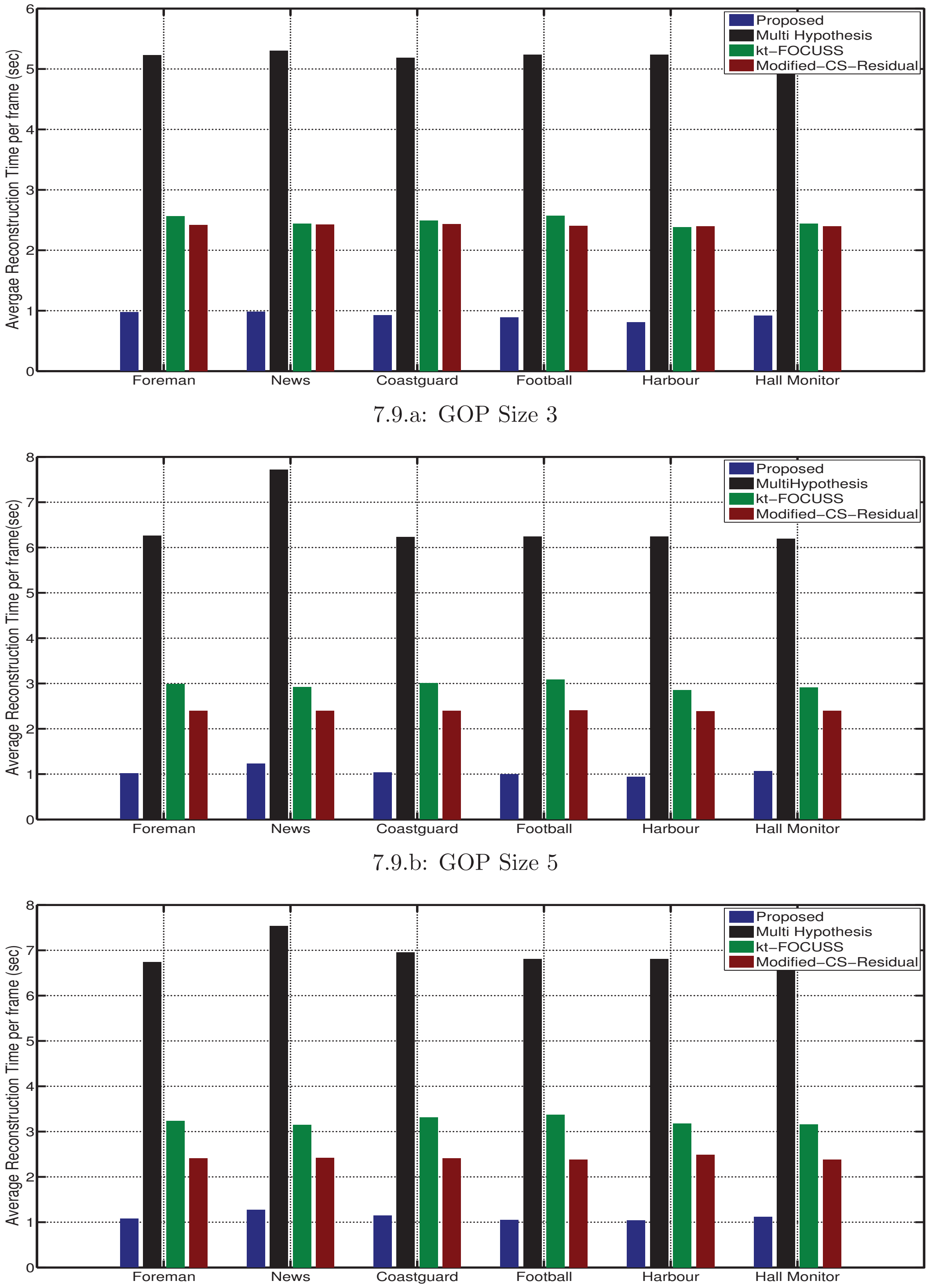
### 7.3.3 Rate Distortion Evaluation

Rate Distortion (R-D) analysis for different test videos are calculated in terms of the average CS measurement rate and PSNR. Figure 7.10 shows the R-D curves for six test video sequences (“Foreman”, “Akiyo”, “Container”, “Hall Monitor”, “Mother Daughter” and “Silent”) using a GOP size of 3. The proposed codec outperforms all the other three codecs for the videos “Akiyo”, “Container” and “Mother Daughter” which do not have fast motions. Its performance is comparable to the multi-hypothesis approach for medium motion videos “Hall Monitor” and “Silent”. At low measurement rates, its performance with the “Foreman” video is comparable with multi-hypothesis. But as the MR increases the difference increases. This is due to the better ME/MC with multiple predictions. Similar results are obtained for larger GOP sizes.

R-D curve for the same videos using GOP sizes of 5 and 8 are shown in Figures 7.11 and 7.12 respectively. In general, with increased GOP size, relative performance among the four codecs remain unchanged. Table 7.3 shows average R-D performance for lower measurements rates of all test video sequences for three GOP sizes. The proposed codec performance is better than all other techniques for all slow motion videos (”Akiyo”, “Bowing”,”Container” and “Mother Daughter”) for all GOP sizes. For medium motion videos, its performance is better for ”Students” at GOP sizes of 3 and 8 and for “Hall Monitor” at GOP sizes 5 and 8. For high motion videos such as “Football”and “Soccer”, the performance of the proposed codec is comparable to multi-hypothesis (average 1dB lower) for all GOP sizes. Modified-CS-Residue performance is better in “Coastguard”, “Bus” and “News” except for GOP size of 8. *k-t* FOCUSS performs better only for the “Harbour” video. These results show that even though proposed codec uses a much simpler side information generation technique, the reconstruction quality obtained is similar

Table 7.2: Average Reconstruction Time (in Seconds) of Video Sequences

| Video | GOP Size = 3 | | | | GOP Size = 5 | | | | GOP Size = 8 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | kt-FOCUSS | ModCS Residual | Multi Hypothesis | Proposed | kt-FOCUSS | ModCS Residual | Multi Hypothesis | Proposed | kt-FOCUSS | ModCS Residual | Multi Hypothesis | Proposed |
| **Akiyo** | 2.47 | 2.40 | 5.64 | **1.07** | 2.94 | 2.37 | 6.58 | **1.07** | 3.21 | 2.38 | 7.36 | **1.17** |
| **Bowing** | 2.54 | 2.41 | 5.48 | **0.94** | 3.04 | 2.38 | 6.61 | **1.03** | 3.27 | 2.43 | 7.22 | **1.12** |
| **Bus** | 2.58 | 2.33 | 5.29 | **0.98** | 3.07 | 2.32 | 6.33 | **1.02** | 3.36 | 2.32 | 6.79 | **1.09** |
| **Coast guard** | 2.53 | 2.36 | 5.43 | **0.96** | 3.04 | 2.33 | 6.35 | **1.04** | 3.31 | 2.35 | 7.01 | **1.18** |
| **Container** | 2.48 | 2.35 | 5.54 | **1.06** | 2.95 | 2.33 | 6.51 | **1.09** | 3.18 | 2.59 | 7.17 | **1.19** |
| **Football** | 2.60 | 2.34 | 5.33 | **1.01** | 3.10 | 2.33 | 6.29 | **1.01** | 3.40 | 2.31 | 6.88 | **1.11** |
| **Foreman** | 2.58 | 2.36 | 5.49 | **0.96** | 3.02 | 2.33 | 6.28 | **1.05** | 3.27 | 2.34 | 6.83 | **1.12** |
| **Hall Monitor** | 2.47 | 2.34 | 5.30 | **1.04** | 2.94 | 2.33 | 6.31 | **1.10** | 3.20 | 2.33 | 6.90 | **1.19** |
| **Harbour** | 2.43 | 2.34 | 5.26 | **0.90** | 2.92 | 2.32 | 6.92 | **1.06** | 3.23 | 2.34 | 6.87 | **1.11** |
| **Mobile Calendar** | 2.40 | 2.31 | 5.30 | **0.86** | 2.90 | 2.30 | 6.27 | **0.91** | 3.22 | 2.28 | 6.85 | **1.00** |
| **Mother Daughter** | 2.47 | 2.39 | 5.57 | **0.97** | 2.95 | 2.37 | 6.55 | **0.99** | 2.95 | 2.37 | 6.55 | **0.99** |
| **News** | 2.47 | 2.36 | 5.47 | **1.02** | 2.96 | 2.33 | 6.91 | **1.15** | 3.19 | 2.37 | 7.37 | **1.24** |
| **Silent** | 2.47 | 2.35 | 5.33 | **1.00** | 2.95 | 2.34 | 6.36 | **1.12** | 3.23 | 2.33 | 6.81 | **1.16** |
| **Students** | 2.40 | 2.35 | 5.39 | **0.96** | 2.90 | 2.34 | 6.45 | **1.05** | 3.22 | 2.33 | 6.87 | **1.13** |
| **Soccer** | 2.61 | 2.35 | 5.42 | **1.00** | 3.12 | 2.34 | 6.33 | **1.02** | 3.44 | 2.40 | 6.94 | **1.13** |

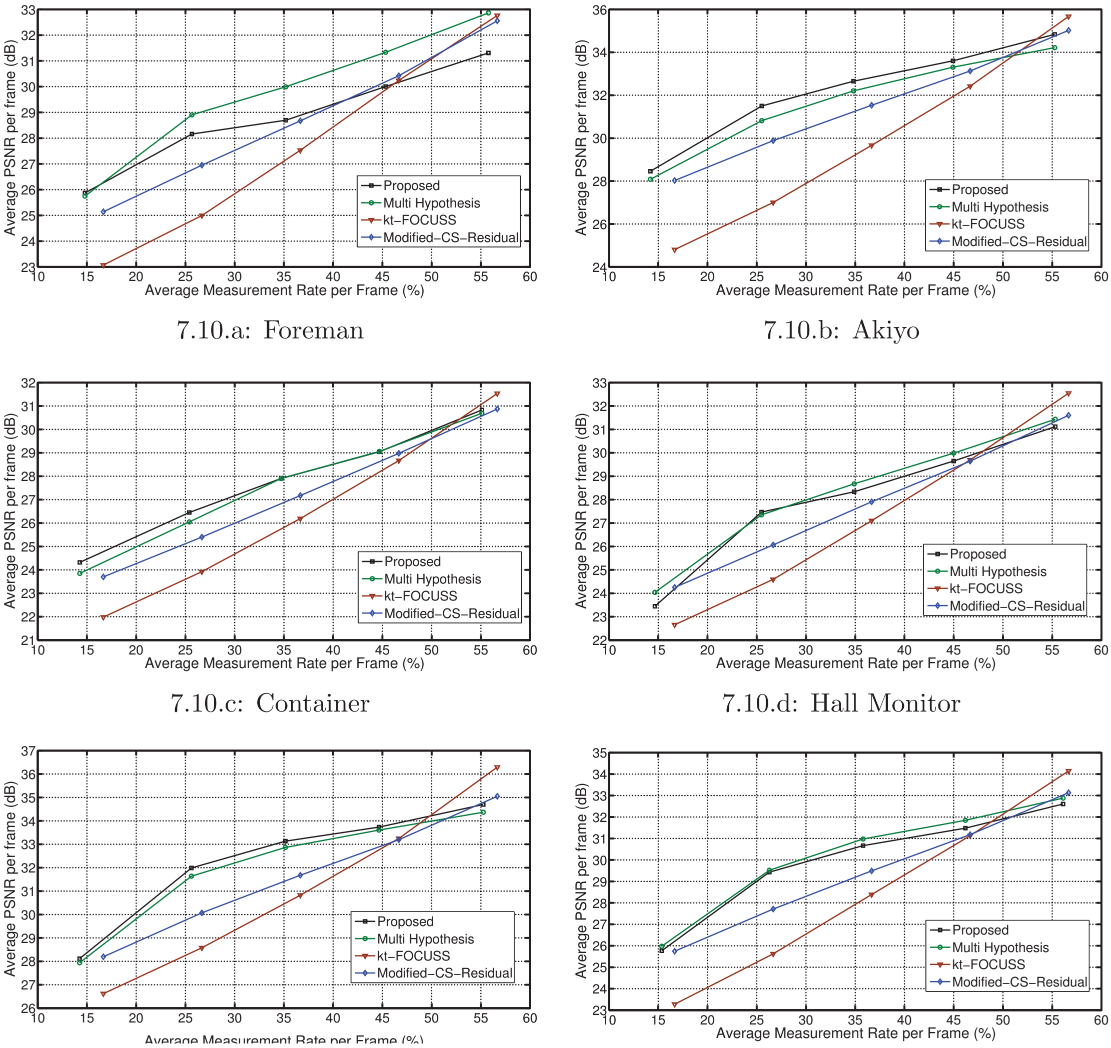


7.9.a: GOP Size 3

7.9.b: GOP Size 5

7.9.c: GOP Size 8

Figure 7.9: Reconstruction complexity comparison for GOP Size 3, 5 and 8



7.10.a: Foreman

7.10.b: Akiyo

7.10.c: Container

7.10.d: Hall Monitor

7.10.e: Mother Daughter

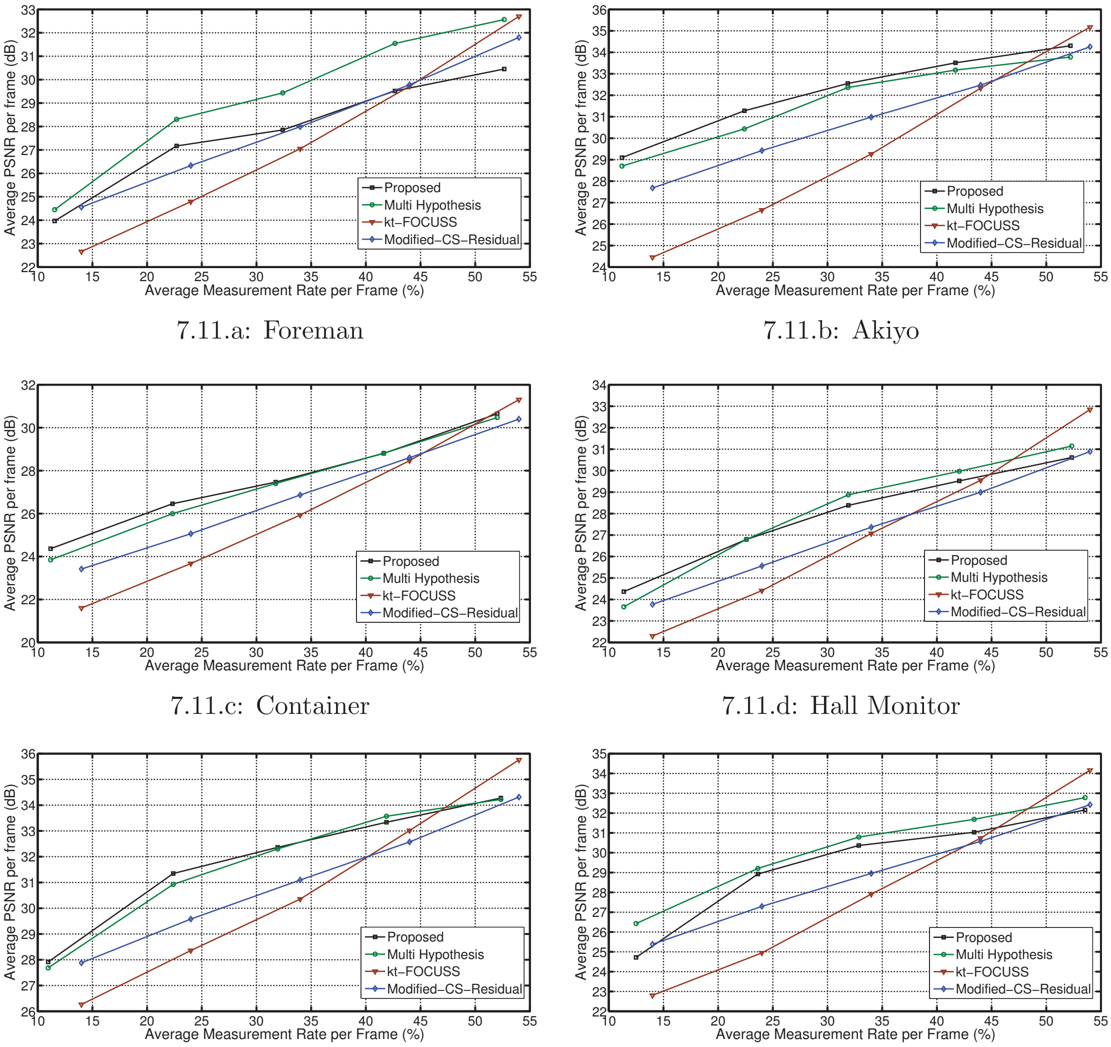
7.10.f: Silent

Figure 7.10: Rate Distortion Curve for GOP Size 3

to or better than ME/MC techniques. In terms of the SSIM, the results are similar to the PSNR results. They are shown in Figures 7.13, 7.14 and 7.15 for GOP sizes of 3, 5 and 8 respectively. For slow motion videos “Akiyo”, “Container” and “Mother Daughter”, the proposed codec outperforms all other techniques. The performance gap decreases for larger GOP sizes. For example, with GOP size 8, the performance becomes comparable to the multi-hypothesis approach. The proposed codec performance is better than Modified-CS-Residue and *k*-*t* FOCUSS for all GOP sizes. Table 7.4 shows average

Table 7.3: Average Rate Distortion Performance of Video Sequences for GOP Size 3, 5 and 8

| Video | GOP Size = 3 | | | | GOP Size = 5 | | | | GOP Size = 8 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | kt-FOCUSS | ModCS Residual | Multi Hypothesis | Proposed | kt-FOCUSS | ModCS Residual | Multi Hypothesis | Proposed | kt-FOCUSS | ModCS Residual | Multi Hypothesis | Proposed |
| **Akiyo** | 27.16 | 29.81 | 30.37 | **30.87** | 26.79 | 29.36 | 30.50 | **30.97** | 26.53 | 29.01 | 29.29 | **30.18** |
| **Bowing** | 24.69 | 29.29 | 30.08 | **30.16** | 24.42 | 28.73 | **29.50** | 29.42 | 24.25 | 28.26 | **29.61** | 29.50 |
| **Bus** | 22.25 | **23.61** | 23.18 | 22.05 | 21.52 | **23.01** | 22.70 | 21.46 | 21.14 | 22.43 | **23.01** | 21.29 |
| **Coast guard** | 23.23 | **25.62** | 25.61 | 24.98 | 22.92 | **25.14** | 24.83 | 23.91 | 22.76 | 24.77 | **25.01** | 23.77 |
| **Container** | 24.03 | 25.43 | 25.93 | **26.23** | 23.74 | 25.11 | 25.75 | **26.10** | 23.51 | 24.86 | 25.72 | **26.15** |
| **Football** | 25.31 | 25.75 | **26.34** | 25.38 | 24.76 | 25.17 | **25.97** | 24.82 | 24.41 | 24.80 | **25.48** | 24.26 |
| **Foreman** | 25.20 | 26.92 | **28.21** | 27.57 | 24.83 | 26.29 | **27.40** | 26.33 | 24.57 | 25.83 | **27.70** | 26.14 |
| **Hall Monitor** | 24.78 | 26.08 | **26.69** | 26.42 | 24.59 | 25.57 | 26.44 | **26.52** | 24.45 | 25.14 | 26.23 | **26.24** |
| **Harbour** | **24.79** | 24.34 | 23.54 | 22.96 | **24.39** | 23.72 | 23.32 | 22.45 | **24.08** | 23.24 | 24.07 | 22.60 |
| **Mobile Calendar** | 20.90 | **22.10** | 20.78 | 20.63 | 20.55 | **21.57** | 20.64 | 20.10 | 20.34 | **21.11** | 20.25 | 19.22 |
| **Mother Daughter** | 28.67 | 29.98 | 30.81 | **31.08** | 28.33 | 29.52 | 30.30 | **30.54** | 28.33 | 29.52 | 29.85 | **29.88** |
| **News** | 24.59 | **25.49** | 25.14 | 25.34 | 24.25 | **25.03** | 24.80 | 24.89 | 23.83 | 24.65 | 24.51 | **24.66** |
| **Silent** | 25.76 | 27.65 | **28.82** | 28.62 | 25.22 | 27.21 | **28.81** | 27.67 | 24.96 | 26.85 | **28.32** | 27.86 |
| **Students** | 20.90 | 27.28 | 27.10 | **27.33** | 20.55 | 26.81 | **27.82** | 26.55 | 20.34 | 26.43 | 26.67 | **26.90** |
| **Soccer** | 27.18 | 27.50 | **28.07** | 27.15 | 26.72 | 26.58 | 26.46 | **26.89** | 26.19 | 25.98 | **26.68** | 25.74 |



7.11.a: Foreman

7.11.b: Akiyo

7.11.c: Container

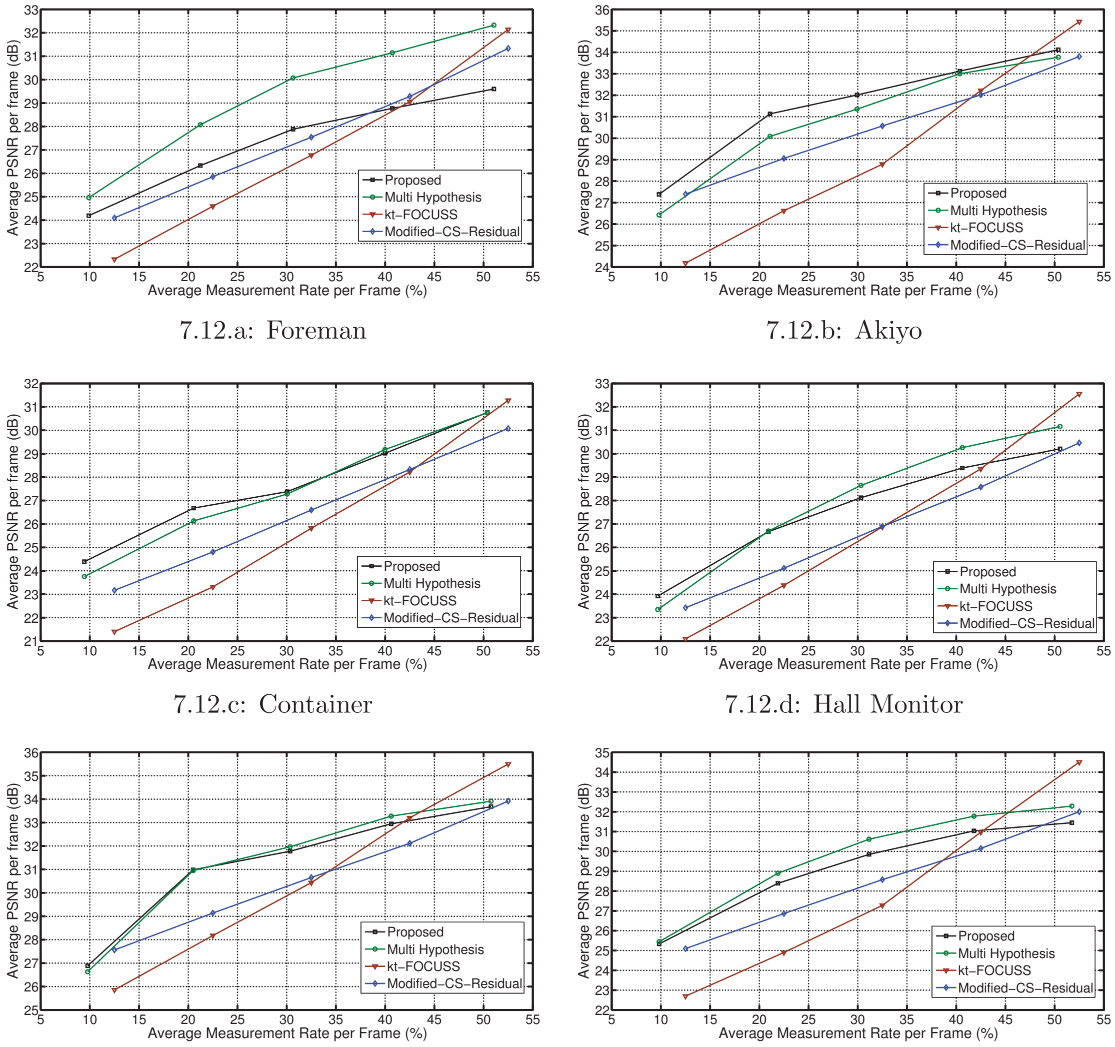
7.11.d: Hall Monitor

7.11.e: Mother Daughter

7.11.f: Silent

Figure 7.11: Rate Distortion Curve for GOP Size 5

SSIM values for all measurement rates for all test videos. Its performance is also not worse than multi-hypothesis for medium motion videos “News” and “Students”. Figure 7.16 shows the visual reconstruction quality for the 123rd frame (a WZ frame) of the “Akiyo” video with an adaptive WZ measurement rate (MR) of 27% (Proposed and Multi Hypothesis) and non-adaptive WZ MR of 30% (Modified-CS-Residue and *k*-*t* FOCUSS). It can be observed that the proposed codec produces better visual quality than the other techniques. Similar observations can be made from Figure 7.17 which shows



7.12.a: Foreman

7.12.b: Akiyo

7.12.c: Container

7.12.d: Hall Monitor

7.12.e: Mother Daughter

7.12.f: Silent

Figure 7.12: Rate Distortion Curve for GOP Size 8

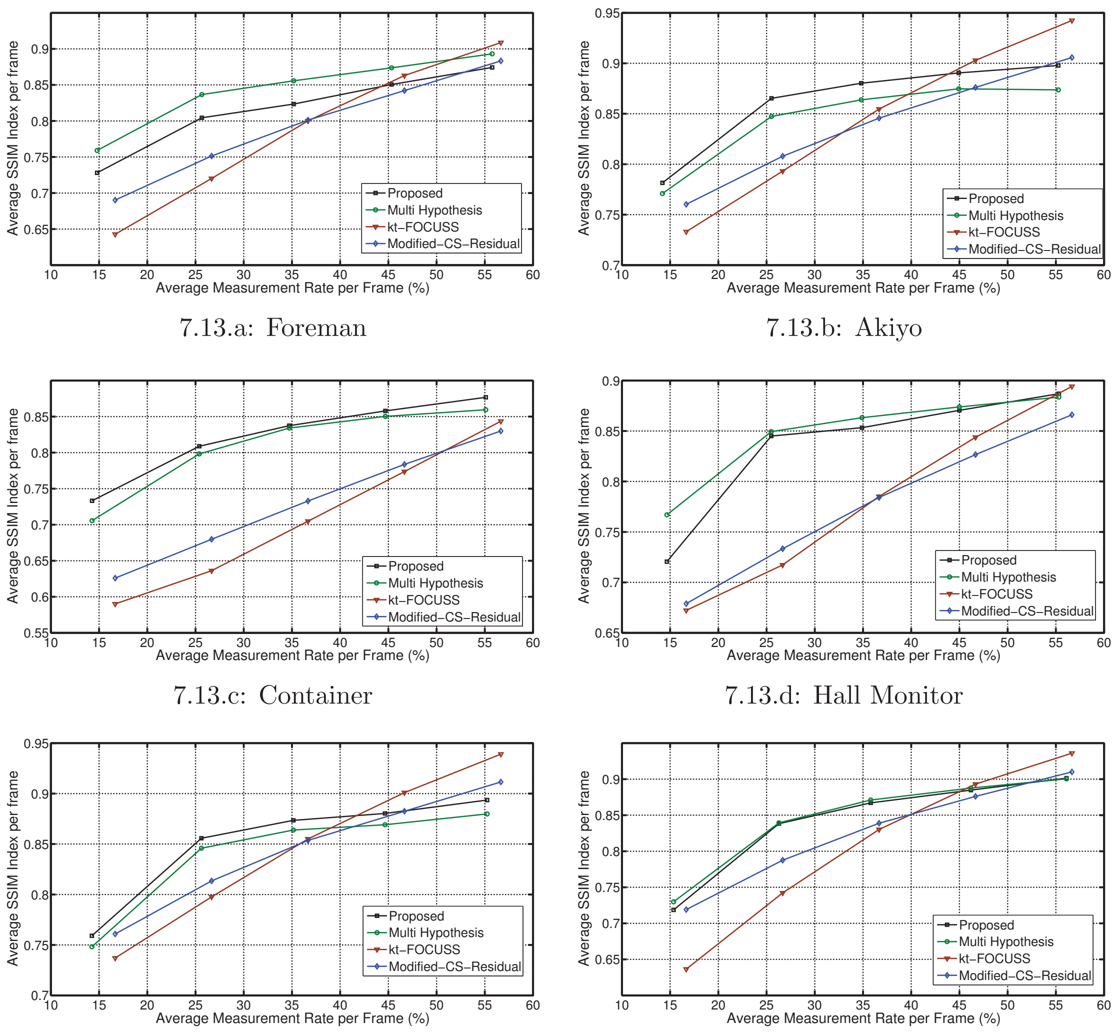
the 24th frame (a WZ frame) of the “Mother Daughter” video.

### 7.3.4 Performance Comparison with DISCOVER and Other Conventional Codecs

In the previous section, the proposed codec is compared with other CS-based video codecs. Here, it is compared against an efficient distributed video codec called “DISCOVER” which has been described in Section 3.3.3 and two existing conventional video

Table 7.4: Average SSIM Index Performance of Video Sequences for GOP Size 3, 5 and 8

| Video | GOP Size = 3 | | | | GOP Size = 5 | | | | GOP Size = 8 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | kt-FOCUSS | ModCS Residual | Multi Hypothesis | Proposed | kt-FOCUSS | ModCS Residual | Multi Hypothesis | Proposed | kt-FOCUSS | ModCS Residual | Multi Hypothesis | Proposed |
| **Akiyo** | 0.79 | 0.80 | 0.83 | **0.84** | 0.79 | 0.79 | 0.83 | **0.85** | 0.78 | 0.78 | 0.80 | **0.84** |
| **Bowing** | 0.70 | 0.78 | 0.83 | **0.83** | 0.69 | 0.76 | **0.81** | 0.80 | 0.68 | 0.74 | **0.81** | 0.80 |
| **Bus** | 0.68 | 0.66 | **0.70** | 0.63 | 0.65 | 0.63 | **0.68** | 0.60 | 0.63 | 0.60 | **0.68** | 0.59 |
| **Coast guard** | 0.59 | 0.69 | **0.73** | 0.69 | 0.57 | 0.67 | **0.70** | 0.64 | 0.57 | 0.65 | **0.71** | 0.63 |
| **Container** | 0.64 | 0.68 | 0.78 | **0.79** | 0.64 | 0.67 | 0.78 | **0.78** | 0.63 | 0.66 | 0.78 | **0.78** |
| **Football** | 0.71 | 0.69 | **0.74** | 0.69 | 0.68 | 0.66 | **0.72** | 0.66 | 0.66 | 0.64 | **0.70** | 0.63 |
| **Foreman** | 0.72 | 0.75 | **0.82** | 0.79 | 0.71 | 0.72 | **0.79** | 0.74 | 0.69 | 0.70 | **0.80** | 0.73 |
| **Hall Monitor** | 0.72 | 0.73 | **0.83** | 0.81 | 0.72 | 0.71 | 0.81 | **0.81** | 0.72 | 0.70 | **0.80** | 0.80 |
| **Harbour** | **0.80** | 0.73 | 0.71 | 0.68 | **0.78** | 0.70 | 0.71 | 0.66 | **0.77** | 0.67 | 0.74 | 0.66 |
| **Mobile Calendar** | 0.62 | **0.66** | 0.61 | 0.61 | 0.60 | **0.62** | 0.60 | 0.57 | 0.58 | **0.59** | 0.58 | 0.51 |
| **Mother Daughter** | 0.80 | 0.81 | 0.82 | **0.83** | 0.79 | 0.79 | 0.81 | **0.81** | 0.79 | 0.79 | 0.81 | **0.81** |
| **News** | 0.73 | 0.72 | 0.80 | **0.81** | 0.72 | 0.70 | **0.79** | 0.79 | 0.72 | 0.68 | **0.78** | 0.78 |
| **Silent** | 0.74 | 0.78 | **0.81** | 0.81 | 0.72 | 0.76 | **0.82** | 0.79 | 0.71 | 0.74 | **0.80** | 0.78 |
| **Students** | 0.62 | 0.76 | 0.78 | **0.79** | 0.60 | 0.74 | 0.76 | **0.78** | 0.58 | 0.72 | 0.77 | **0.77** |
| **Soccer** | 0.76 | 0.72 | **0.76** | 0.73 | 0.74 | 0.68 | **0.76** | 0.71 | 0.72 | 0.66 | **0.73** | 0.68 |



7.13.a: Foreman

7.13.b: Akiyo

7.13.c: Container

7.13.d: Hall Monitor

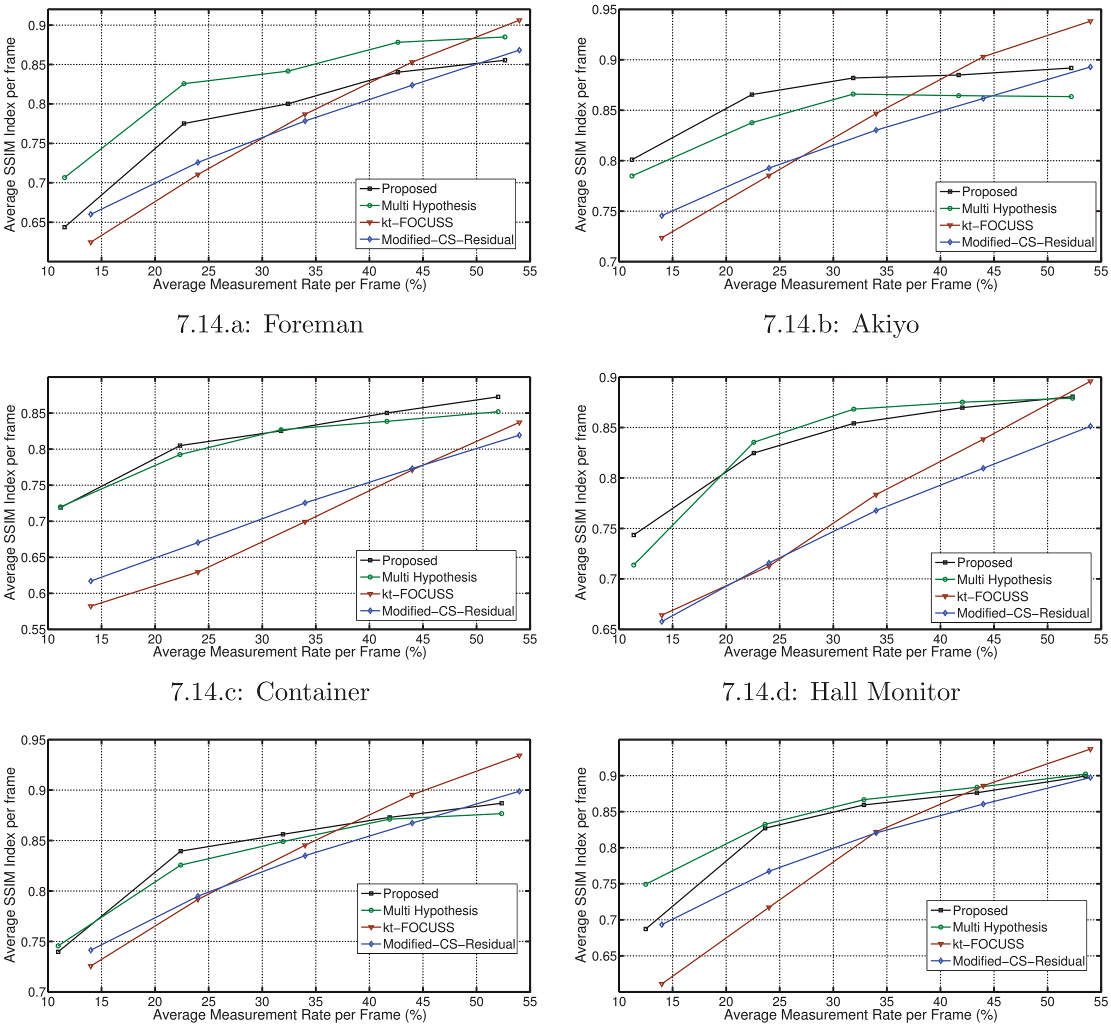
7.13.e: Mother Daughter

7.13.f: Silent

Figure 7.13: SSIM Index for GOP Size 3

coding standards (H.264 and H.263+) in terms of compression ratio and bit rates. H.264 is a video codec standard developed by ITU-T and is also known as H.264/AVC (Advanced Video Coding) [3]. Two different coding options are used in these experiments:

- H.264 (I-P-P) with GOP size of 3
- H.264 Intra with only intra coding of frames



7.14.a: Foreman

7.14.b: Akiyo

7.14.c: Container

7.14.d: Hall Monitor

7.14.e: Mother Daughter

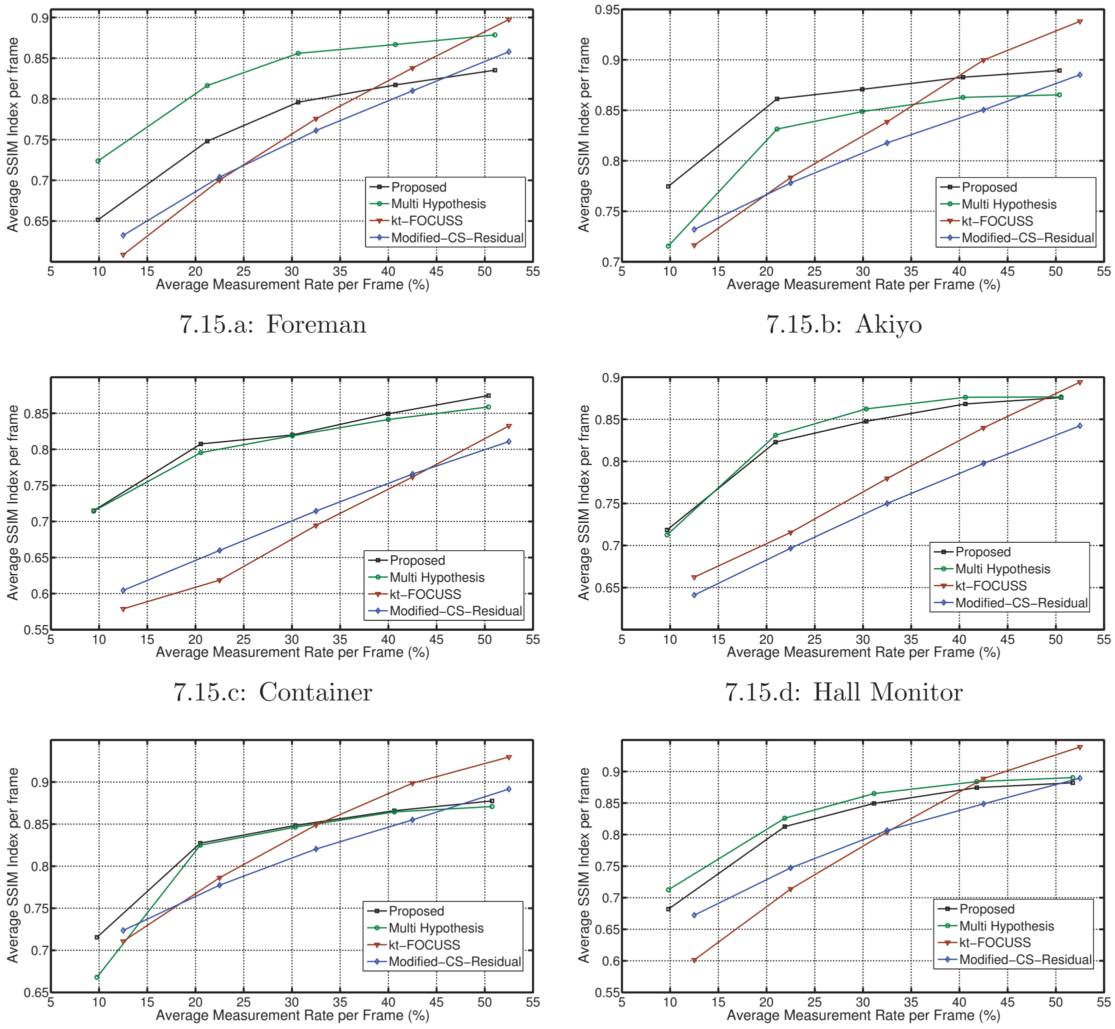
7.14.f: Silent

Figure 7.14: SSIM Index for GOP Size 5

H.263+ is an enhanced version of the H.263 standard developed by ITU-T [126] Two different coding options for H.263+ are chosen in these experiment:

- H.263+ (I-P-P) with GOP size 3

- H.263+ Intra which uses only Intra coding of frames under different quantization parameters. 8, 10, 12, 14, 16, 20, 24, 28 and 32 bit quantization are used to obtain different bit rates and quality.



7.15.a: Foreman

7.15.b: Akiyo

7.15.c: Container

7.15.d: Hall Monitor

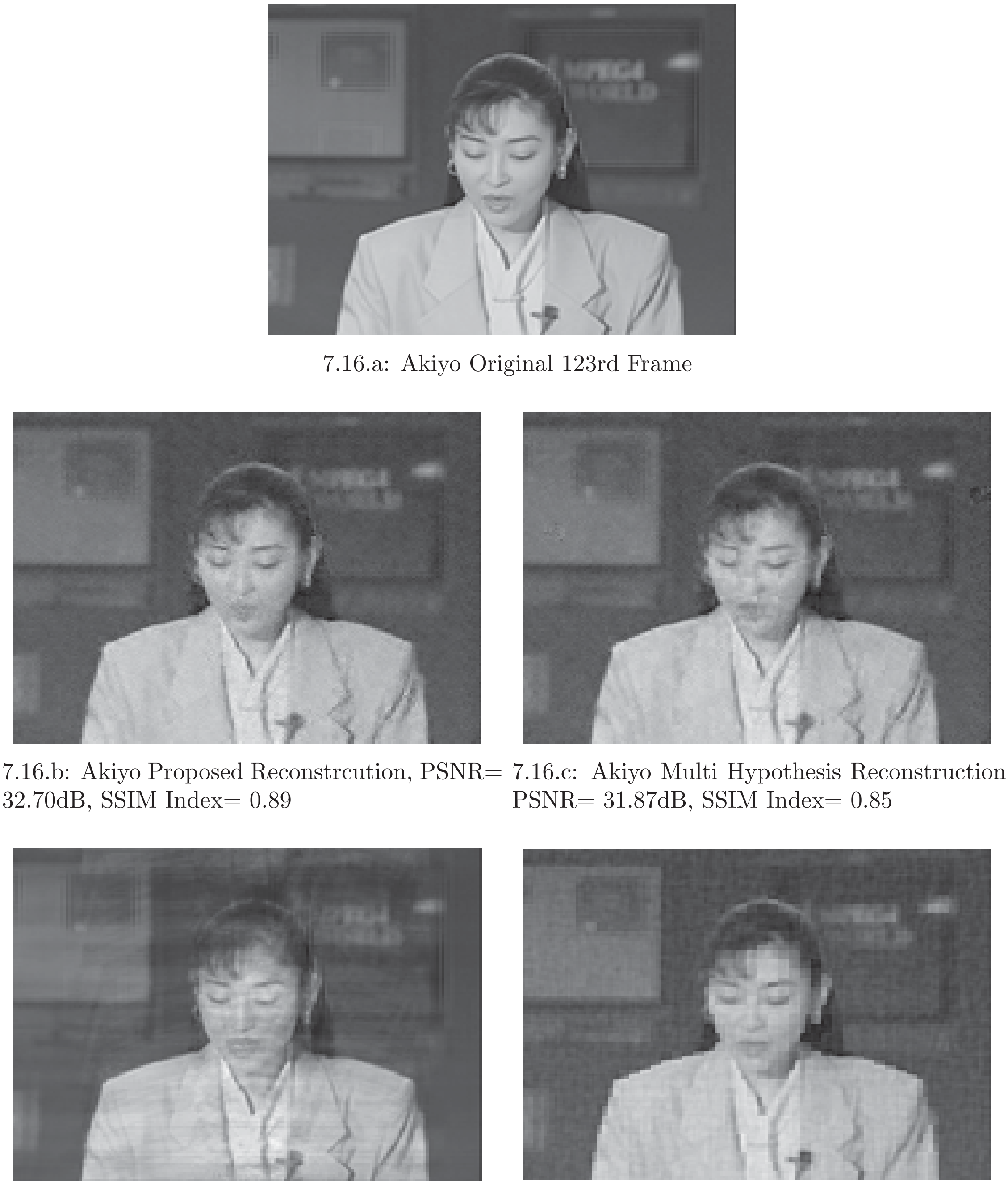
7.15.e: Mother Daughter

7.15.f: Silent

Figure 7.15: SSIM Index for GOP Size 8

The H.264 and H.263+ codecs used are those provided by the free software FFmpeg (Fast Forward MPEG) [127].

The CS measurements are quantized using the quantization scheme developed in Section 4.3.2. Each CS measurement is allotted 8 bits for quantization. The quantized CS measurements are then entropy coded using Huffman coding. Each raw (uncompressed) pixel requires 8 bits since only the luminance component is used.



7.16.a: Akiyo Original 123rd Frame

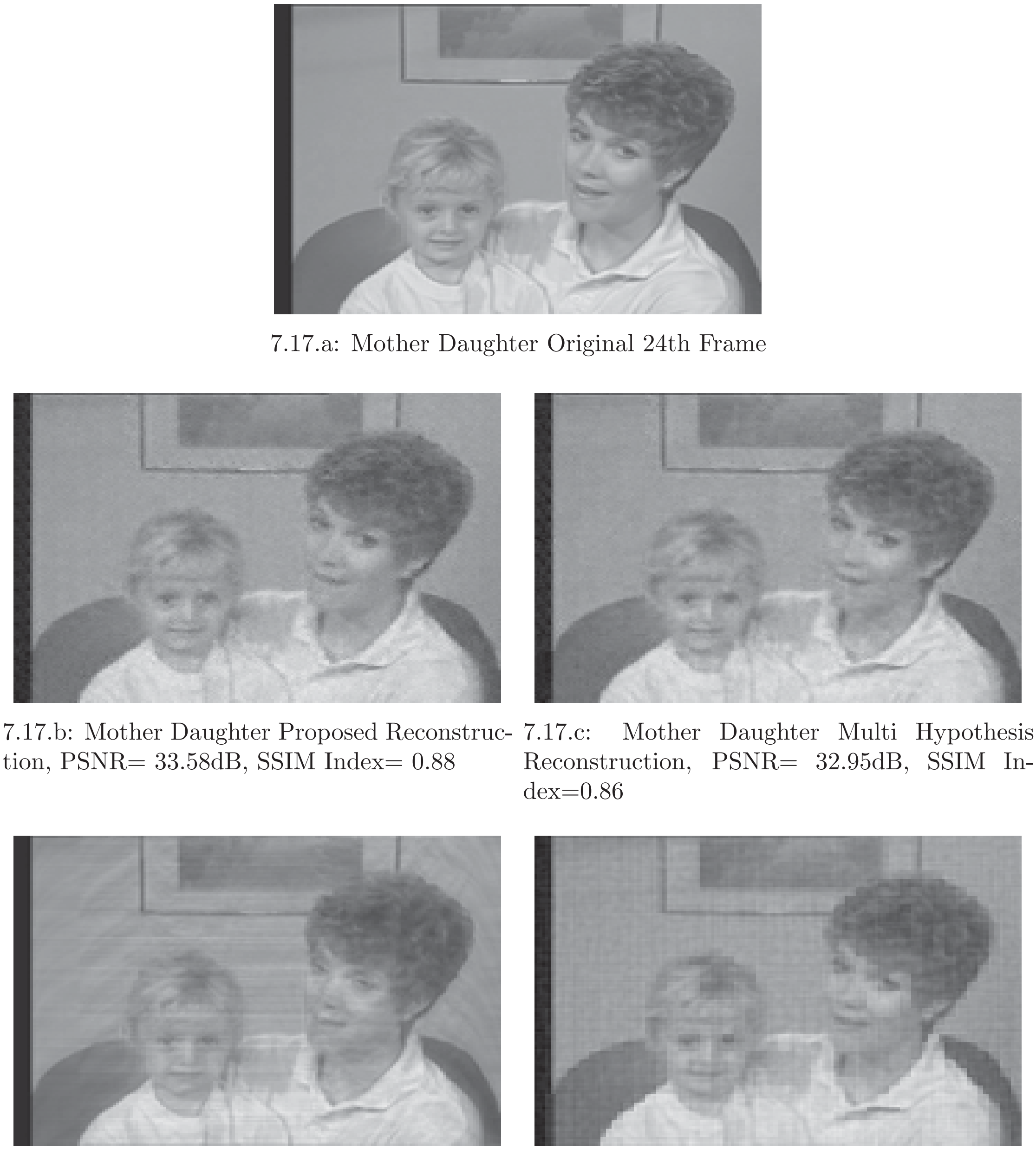
7.16.b: Akiyo Proposed Reconstrcution, PSNR= 32.70dB, SSIM Index= 0.89

7.16.c: Akiyo Multi Hypothesis Reconstruction, PSNR= 31.87dB, SSIM Index= 0.85

7.16.d: Akiyo kt-FOCUSS Reconstruction, PSNR= 27.42dB, SSIM Index= 0.82

7.16.e: Akiyo Modified CS Residual Reconstruction, PSNR= 30.59 dB, SSIM Index= 0.83

Figure 7.16: Visual Reconstruction Quality of Akiyo 123rd Frame for GOP Size 3



7.17.a: Mother Daughter Original 24th Frame

7.17.b: Mother Daughter Proposed Reconstruction, PSNR= 33.58dB, SSIM Index= 0.88

7.17.c: Mother Daughter Multi Hypothesis Reconstruction, PSNR= 32.95dB, SSIM Index=0.86

7.17.d: Mother Daughter kt-FOCUSS Reconstruction, PSNR= 30.49dB SSIM Index= 0.84

7.17.e: Mother Daughter Modified CS Residual Reconstruction, PSNR= 30.90dB, SSIM Index= 0.83

Figure 7.17: Visual Reconstruction Quality of Mother Daughter 24th Frame for GOP Size 3

Four QCIF videos – “Foreman”, “News”, “Coastguard” and “Hall Monitor” are used in these experiments. These videos have a frame rate of 30 frames per second. A GOP size of 3 is used.

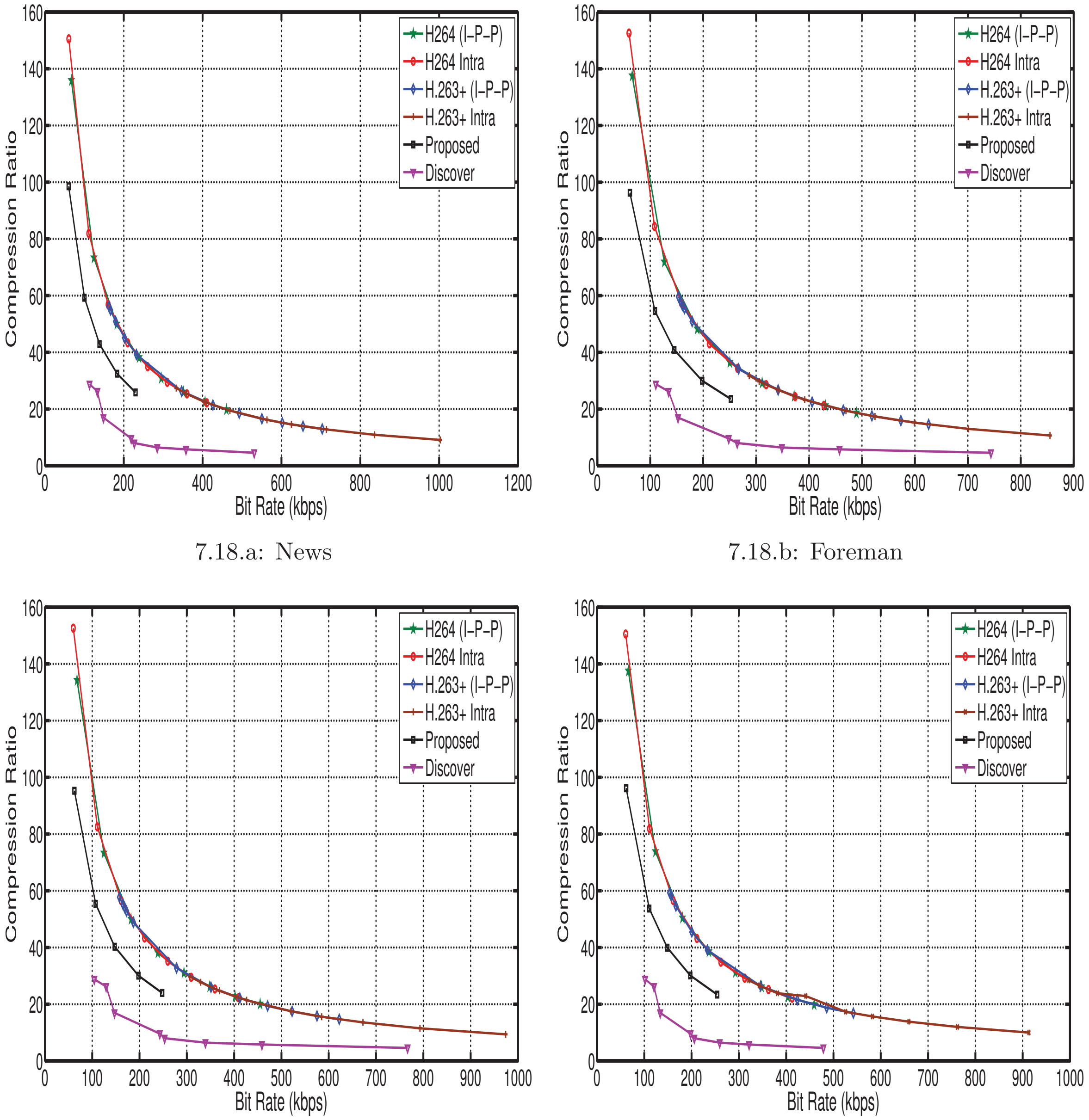
### Results

First, the compression ratio that are achieved by the codecs are compared. Figure 7.18 shows the compression ratio under different bit rates for four test video sequences. The proposed codec outperforms DISCOVER at all bit rates. However its performance against conventional video codecs is not as good due to the efficient intra coding used by H.264 and H.263+ which produces higher compression.

The reconstruction quality of different codecs at different bit rates are shown in Figure 7.19. The proposed codec performs better than H.264 Intra and H.263+ Intra for all four test videos at all bit rates. Its performance against H.263+ (I-P-P) is better for “Foreman” video and comparable for the other three videos.

The decoding times required by the proposed codec and by DISCOVER are shown in Figure 7.20. Different quantization parameters were used in DISCOVER to achieve different bit rates and quality [128], corresponding Bitrate points on $x - axis$ shows corresponding bitrates in proposed codec and DISCOVER. The decoding times required by the proposed block-based distributed CS video codec stays roughly the same at all bit rates. However, that required for DISCOVER increases as bit rate increases. Furthermore, the decoding times for the proposed codec are much shorter than that required for DISCOVER at all bit rates.

The decoding times for the two conventional coding standards will be much shorter. This is because conventional codecs are designed in such a way that the encoding process is complex and decoding process is simple whereas the proposed codec is designed with a different philosophy. Hence their decoding times are not shown here.



7.18.a: News

7.18.b: Foreman

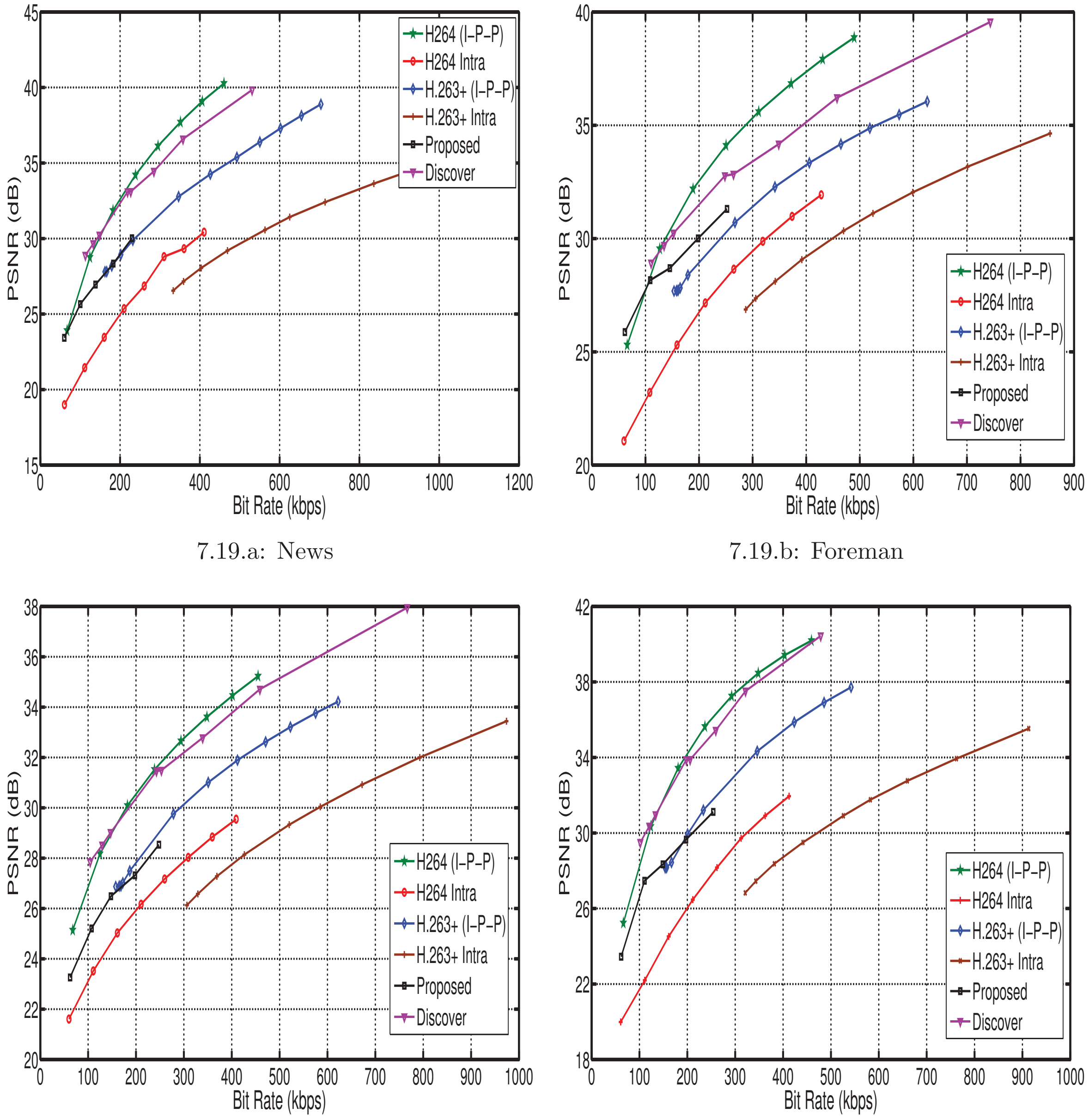
7.18.c: Coastguard

7.18.d: Hall Monitor

Figure 7.18: Bit Rate vs Compression Ratio for Video Sequences - GOP Size 3

## 7.4 Summary

In this chapter, a block-based distributed video CS codec is proposed. The encoding is performed on block by block basis using CS and can be implemented with reduced



7.19.a: News

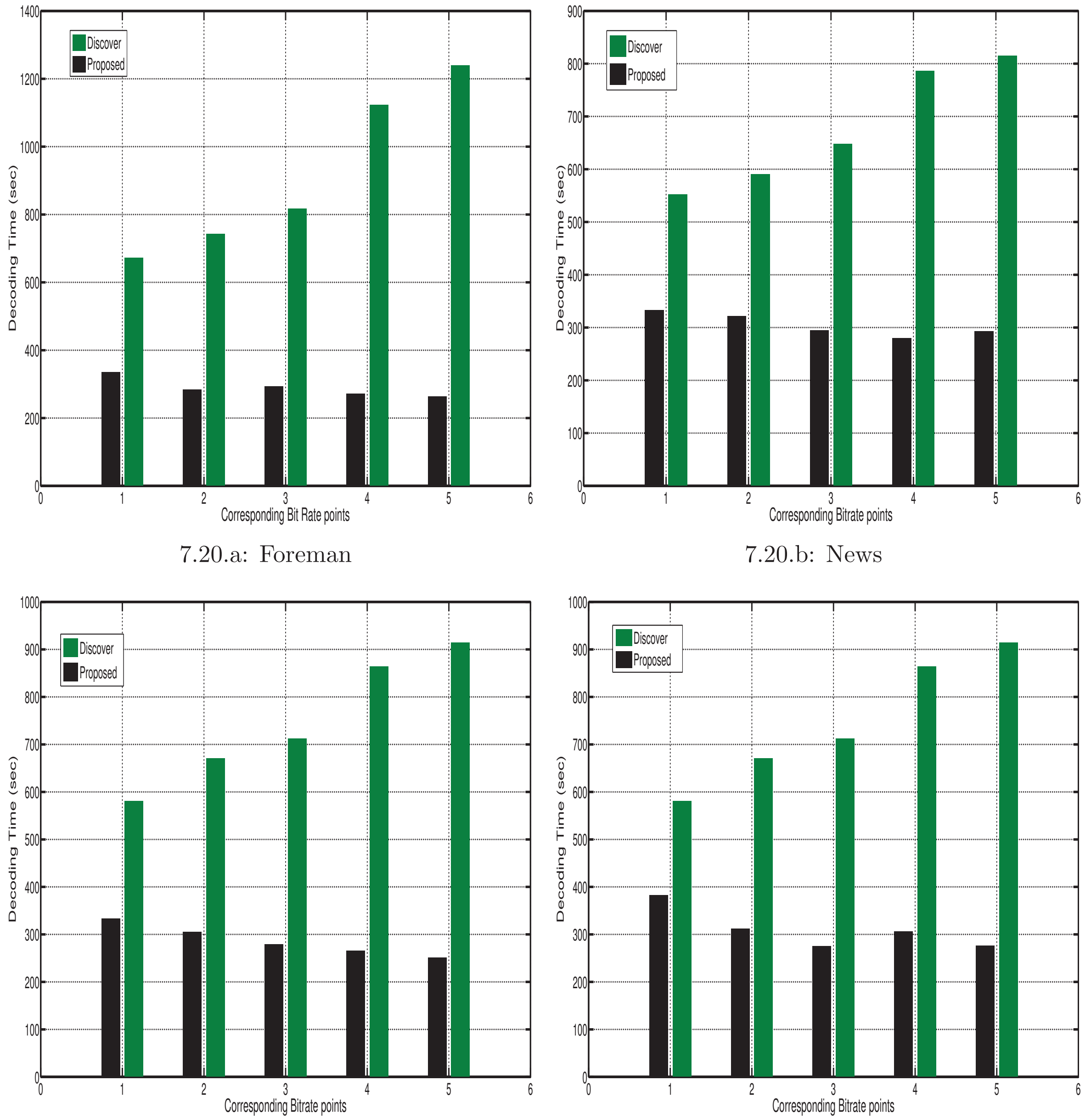
7.19.b: Foreman

7.19.c: Coastguard

7.19.d: Hall Monitor

Figure 7.19: Bit Rate vs PSNR for Video Sequences - GOP Size 3

complexity compared with full frame encoding. Furthermore, the sampling rate used to encode similar blocks in a WZ frame is adaptive, which increases the compression ratio. At the decoder, a simple side information generation technique is used which exploits both inter-frame and intra-frame block correlation. The proposed codec does not require



7.20.a: Foreman

7.20.b: News

7.20.c: Coastguard

7.20.d: Hall Monitor

Figure 7.20: Reconstruction Time Complexity Comparison with DISCOVER

a feedback channel nor motion estimation as are required by other codecs. It does not even need the key frames to be decoded before side information can be generated. The performance of the proposed codec is evaluated using videos with slow, medium and fast motions. The proposed codec performs comparably with the multi-hypothesis approach and outperforms other tested CS codecs. It is also better than H.264 Intra and H.263 Intra and comparable to H.263 (I-P-P) in terms of reconstructed video quality. Furthermore, it outperforms DISCOVER in terms of simplicity, compression ratio and reconstruction complexity.

# Chapter 8

# Conclusions and Future Work

## 8.1 Conclusions

In this thesis, a distributed framework for image and video coding that is based on compressed sensing has been presented. It enables low complexity, resource constraint encoding to be performed which will lower the cost of the encoders (cameras). Two new key techniques have been incorporated into the encoding process. The first one is a quantization scheme that is tailored for CS measurements. The second is a new side information generation scheme which is simple and effective and can easily be integrated with any CS reconstruction algorithm. The framework includes both whole frame (image) as well as block encoding. There are also options for using adaptive encoding rates which controls the quality of the image or video obtained. Frame and block correlations are exploited to increase the compression ratio without having to perform computationally expensive operations such as motion estimation either at the encoder or the decoder. The following conclusions can be drawn from the work presented in this thesis.

(i) The distributed image and video coding schemes presented are able to produce good quality reconstructed images and videos even though the encoders are extremely simple. The entire process is based on CS, without the need for motion compensation and estimation or a feedback channel. The quality of the reconstructed images and videos is enhanced by using side information. The side information generation schemes work in the CS measurement domain without the need to first decode any previously received frames. Minimal computation is needed for these SI generation schemes.

(ii) Structurally Random Matrices is the best type of sensing matrices for acquiring full-image CS measurements. It uses inherent block-based operation which makes it suitable for acquiring large dimension image and video signals. In particular, SBHE has been found to be the optimal sensing matrix in our experiments. It is hardware friendly and has low computational demand.

(iii) In our evaluation of CS reconstruction algorithms for image reconstruction, the performances of all algorithms tested perform similar in terms of reconstruction quality except for greedy algorithms. SpaRSA produces the best reconstruction quality, with the lowest number of iterations and reconstruction time.

(iv) The proposed quantization matrix based on the statistical properties of CS measurements yields better results than a uniform quantizer and those used for JPEG. This quantizer is also robust against different statistics of individual images.

(v) It is shown that CS measurements of adjacent video frames in frame based CS encoding are highly correlated. This correlation is used to design a simple and effective side information generation schemes. The performance of proposed SI scheme is found to be efficient than MCI. Experimental evaluation of proposed codec show that proposed codec out-perform existing techniques in simplicity and has comparable performance in quality.

(vi) The correlation between adjacent video frames can be estimated effectively using the MSE of their CS measurements.

(vii) MSE is more effective is determining intra-frame and inter-frame block correlation than correlation coefficient in block-based CS encoding.

(viii) For block-based encoding, block size generally does not affect the CS reconstruction quality.

## 8.2 Further Research

The distributed image and video coding techniques developed in this thesis can be extended to other imaging applications. The future possibilities for multi-view coding and hyperspectral imaging are particularly promising.

### 8.2.1 Multi-view Image/Video Coding

The most obvious avenue for further research is the extension of the present methods to multi-view image and video coding. In multi-view camera systems, the same scene is captured from several viewpoints. Not only is there a high correlation among consecutive video frames from a single camera but also those among from different cameras. This type of redundancy can potentially be exploited using the techniques presented in this thesis. CS has great potential for multi-view image and video acquisition by lowering the memory and computation requirements of the cameras. The proposed SI generation techniques in this thesis can be extended to lower the measurement rates and improve the reconstruction quality without putting extra burden on the decoding process.

### 8.2.2 Hyperspectral Imaging

In hyperspectral imaging, a number of images are acquired simultaneously at different wavelengths. The data collected essentially forms a three-dimensional data-cube, known as a voxel, with two spatial dimensions and one spectral dimension. Hyperspectral images usually have a large number of homogeneous regions. For each data-cube, its neighbouring pixel-vectors will likely share similar spectral characteristics. The difference in spectral information between two voxels is typically very small and thus they are highly correlated. This correlation could be exploited to compress hyperspectral data using the techniques presented here.

# Appendix A

# List of Publications

## Peer-reviewed international conferences

- Yousuf Baig, Edmund M-K. Lai and J.P. Lewis, ”Quantization effects on Compressed Sensing Video” in *Proceedings of IEEE 17th International Conference on Telecommunications (ICT)*, Doha, Qatar, April 4-7, 2010, pp.935-940.

- Yousuf Baig, Edmund M-K Lai and Amal Punchihewa. ”Distributed Video Coding Based on Compressed Sensing,” in Proceedings of IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Melbourne, Australia, July 9 - 13, 2012, pp.325-330.

- Yousuf Baig, Edmund M-K. Lai and Amal Punchihewa, ”Low complexity side information for distributed compressed video coding,” in *Proceedings of IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)*, Hong Kong, Aug 12-15, 2012, pp.436-441.

## Book chapter

- Muhammad Yousuf Baig, Edmund M-K. Lai and Amal Punchihewa, ”Compressive Video Coding: A Review of the State-Of-The-Art”, in *Video Compression*, ISBN: 978-953-51-0422-3, 2012 pp.3-14, Published by InTech

# References

[1] ITU, "T.81 : Information technology digital compression and coding of continuous-tone still images requirements and guidelines"," *ITU-T Recommendations*, 2004.

[2] P. Symes, *Digital Video Compression.* McGraw-Hill, 2004.

[3] ITU, "Advanced video coding for generic audiovisual services," *ITU-T Recommendations for H.264*, 2005.

[4] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[5] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury", "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921 – 960, 2007.

[6] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "Mesheye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *Proceedings of 6th International Conference on Information Processing in Sensor Networks (IPSN'07)*, 2007, pp. 360–369.

[7] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 765–776, jul 2005.

[8] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz, "High-speed videography using a dense camera array," in *Proceedings of 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, 2004, pp. 294–301.

[9] F. Pereira, “Distributed video coding: Basics, main solutions and trends,” in *Proceedings of IEEE International Conference on Multimedia and Expo*, 28 June-3 July 2009, pp. 1592–1595.

[10] D. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[11] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[12] R. Baraniuk, “Compressive sensing [lecture notes],” *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, Jul. 2007.

[13] T. Do, Y. Chen, D. Nguyen, N. Nguyen, L. Gan, and T. Tran, “Distributed compressed video sensing,” in *Proceedings of 43rd Annual Conference on Information Sciences and Systems (CISS09)*, Mar. 2009, pp. 1–2.

[14] L.-W. Kang and C.-S. Lu, “Distributed compressive video sensing,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP09)*, Apr. 2009, pp. 1169–1172.

[15] V. Stankovic, L. Stankovic, and S. Chencg, “Compressive video sampling,” in *Proceedings of 16th European Signal Processing Conference*, Lausanne, Switzerland, Aug. 2008.

[16] J. Y. Park and M. Wakin, “A multiscale framework for compressive sensing of video,” in *Proceedings of Picture Coding Symposium*, Chicago, IL, USA, 6-8 May 2009.

[17] J. Prades-Nebot, M. Yi, and T. Huang, “Distributed video coding using compressive sampling,” in *Proceedings of Picture Coding Symposium*, Chicago, IL, USA, 6-8 May 2009.

[18] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, “Compressive imaging for video representation and coding,” in *Proceedings of Picture Coding Symposium*, Beijing, China, 24-26 April 2006.

[19] E. Candes and M. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, pp. 21–30, Mar 2008.

[20] J. Ellenberg, “Fill in the blanks: Using math to turn lo-res datasets into hi-res samples,” *Wired Magazine*, vol. 18, no. 3, Mar. 2010.

[21] C. Shannon, “Communication in the presence of noise,” *Proceedings of IEEE*, vol. 37, pp. 10–21, Jan. 1949.

[22] ——, “Classic paper: Communication in the presence of noise,” *Proceedings of the IEEE*, vol. 86, no. 2, pp. 447–457, Feb. 1998.

[23] E. Candes and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007.

[24] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, “A simple proof of the restricted isometry principle for random matrices,” *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, Dec. 2008.

[25] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss, “Combining geometry and combinatorics: A unified approach to sparse signal recovery,” in *Proceedings of 46th Annual Allerton Conference on Communication, Control, and Computing*, sept 2008, pp. 798 –805.

[26] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, “Model-based compressive sensing,” *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982 – 2001, april 2010.

[27] S. Chen and D. Donoho, “Basis pursuit,” in *Proceedings of IEEE Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov. 1994, pp. 41–44.

[28] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[29] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[30] S. Mallat and Z. Zhang, “Matching pursuit with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 3397–3415, Dec. 1993.

[31] J. Tropp and A. Gilbert, “Signal recovery from partial information via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[32] D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, “Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094 –1121, feb. 2012.

[33] D. Needell and J. Tropp, “Cosamp: Iterative signal recovery from incomplete and inaccurate samples,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301 – 321, 2009.

[34] M. Figueiredo, R. Nowak, and S. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[35] E. van den Berg and M. P. Friedlander, “Probing the pareto frontier for basis pursuit solutions,” *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[36] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, “Sparse Reconstruction by Separable Approximation,” *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[37] J. M. Bioucas-Dias and M. A. T. Figueiredo, “A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration,” *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, 2007.

[38] M. Lustig, D. Donoho, and J. Pauly, “Sparse mri: The application of compressed sensing for rapid mr imaging.” *Magn Reson Med*, vol. 58, no. 6, pp. 1182–95, 2007.

[39] M. Lustig, D. Donoho, J. Santos, and J. Pauly, “Compressed sensing mri,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72–82, 2008.

[40] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, “Single-pixel imaging via compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[41] http://www.inviewcorp.com.

[42] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk, “A new camera architecture based on optical-domain compression,” in *Proceedings of SPIE Symposium on Electronic Imaging: Computational Imaging*, vol. 6065, 2006.

[43] R. Fergus, A. Torralba, W. T. Freeman, R. Fergus, A. Torralba, and W. T. Freeman, “Random lens imaging,” MIT Computer Science and Artificial Intelligence Laboratory, Tech. Rep., 2006.

[44] W. Tang, J. Ma, and F. J. Herrmann, “Optimized compressed sensing for curvelet-based seismic data reconstruction,” *preprint*, 2009.

[45] R. M. Willett, M. E. Gehm, and D. J. Brady, “Multiscale reconstruction for computational spectral imaging,” *Proceedings of SPIE*, vol. 6498, 2007.

[46] M. E. Gehm, R. John, D. J. Brady, R. M. Willett, and T. J. Schulz, “Single-shot compressive spectral imaging with a dual-disperser architecture,” *Opt. Express*, vol. 15, no. 21, pp. 14 013–14 027, Oct 2007.

[47] M. Herman and T. Strohmer, “High-resolution radar via compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2275–2284, June.

[48] Y.-S. Yoon and M. G. Amin, “Compressed sensing technique for high-resolution radar imaging,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6968, May 2008.

[49] http://dsp.rice.edu/cs.

[50] E. J. Candes and J. Romberg, “Practical signal recovery from random projections,” in *Proceedings of SPIE Computational Imaging*, vol. 5674, 2005, pp. 76–86.

[51] T. Do, L. G. N. Nguyen, and T. D. Tran, “Sparsity adaptive matching pursuit algorithm for practical compressed sensing,” in *Proceedings of 42nd IEEE Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 27-29 October 2008.

[52] T. Do, L. Gan, and T. Tran, “Fast compressive sampling with structurally random matrices,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 30 March - 4 April 2008, pp. 3369–3372.

[53] T. D. Lu Gan and T. Tran, “Fast compressive imaging using scrambled block hadamard ensemble,” in *Proceedings of European Signal Processing Conference*, August 2008.

[54] L. Gan, “Block compressed sensing of natural images,” in *15th International Conference on Digital Signal Processing*, July 2007, pp. 403–406.

[55] S. Mun and J. E. Fowler, “Block compressed sensing of images using directional transforms,” in *Proceedings of IEEE International Conference on Image Processing*, November 2009, pp. 3021–3024.

[56] Y. Yang, O. Au, L. Fang, X. Wen, and W. Tang, “Perceptual compressive sensing for image signals,” in *IEEE International Conference on Multimedia and Expo (ICME’09)*, 2009, pp. 89–92.

[57] Z. Gao, C. Xiong, C. Zhou, and H. Wang, “Compressive sampling with coefficients random permutations for image compression,” in *International Conference on Multimedia and Signal Processing (CMSP’11)*, vol. 1, 2011, pp. 321–324.

[58] Y. Tsaig and D. L. Donoho, “Extensions of compressed sensing,” *Journal of Signal Processing, Elsevier*, vol. 86, no. 3, pp. 549–571, mar 2006.

[59] F. W. B. Han and X. Tian, “Image representation by compressive sensing for visual sensor networks,” *Journal of Visual Communication and Image Representation*, vol. 21, no. 4, pp. 325–333, May 2010.

[60] S. M. J. E. Fowler and E. W. Tramel, “Multiscale block compressed sensing with smoothed projected landweber reconstruction,” in *Proceedings of European Signal Processing Conference*, August 2011, pp. 564–568.

[61] S. Ji, Y. Xue, and L. Carin, “Bayesian compressive sensing,” *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346 –2356, june 2008.

[62] L. He and L. Carin, “Exploiting structure in wavelet-based bayesian compressive sensing,” *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3488–3497, 2009.

[63] L. He, H. Chen, and L. Carin, “Tree-structured compressive sensing with variational bayesian analysis,” *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 233–236, 2010.

[64] S. Babacan, R. Molina, and A. Katsaggelos, “Bayesian compressive sensing using laplace priors,” *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 53–63, 2010.

[65] J. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[66] A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[67] A. Wyner, “Recent results in the shannon theory,” *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 2–10, Jan. 1974.

[68] S. Xie, S. Rahardja, and Z. Li, “Wyner-ziv image coding from random projections,” in *IEEE International Conference on Multimedia and Expo (ICME’07)*, july 2007, pp. 136 –139.

[69] Z. Gan, L. Qi, and X. Zhu, “Wyner-ziv coding of image using compressed sensing,” in *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS’10)*, dec 2010, pp. 1 –4.

[70] L. Sun, X. Wen, M. Lei, H. Xu, J. Zhu, and Y. Wei, “Signal reconstruction based on block compressed sensing,” in *Artificial Intelligence and Computational Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 7003, pp. 312–319.

[71] X. Zhang, J. Chen, H. Meng, and X. Tian, “Self-adaptive structured image sensing,” *Optical Engineering*, vol. 51, no. 12, pp. 127 001–1 127 001–13, 2012.

[72] H.-W. Chen, L.-W. Kang, and C.-S. Lu, “Dictionay learning-based distributed compressive video sesning,” in *Proceedings of Picture Coding Symposium*, Dec. 2010, pp. 210–213.

[73] H. W. Chen, L. W. Kang, and C. S. Lu, “Dynamic measurement rate allocation for distributed compressive video sensing,” in *Proceedings of IEEE/SPIE Visual Communications and Image Processing (VCIP): special session on Random Projection and Compressive Sensing*, Jul. 2010, pp. 774 401–774 410.

[74] J. E. Fowler, S. Mun, and E. W. Tramel, "Block-based compressed sensing of images and video," *Foundations and Trends in Signal Process.*, vol. 4, no. 4, pp. 297–416, 2012.

[75] S. Mun and J. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *Data Compression Conference (DCC), 2011*, 2011.

[76] H. Xiaoran, Z. Bojin, and C. Anni, "Measurement compression in compressive sampling based distributed video coding," in *Proceedings of 2nd IEEE International Conference on Information Engineering and Computer Science (ICIECS10)*, 2010, pp. 1–4.

[77] T. Do, X. Lu, and J. Sole, "Compressive sensing with adaptive pixel domain reconstruction for block-based video coding," in *Proceedings of IEEE 17th International Conference on Image Processing (ICIP10)*, 2010, pp. 3377–3380.

[78] J. C. Y. H. Jung, "Motion estimated and compensated compressed sensing dynamic magnetic resonance imaging:what we can learn from video compression techniques," *Imaging Systems and Technology*, vol. 20, no. 2, pp. 81–98, June 2010.

[79] Z. Liu, H. Zhao, and A. Elezzabi, "Block-based adaptive compressed sensing for video," in *Proceedings of IEEE 17th International Conference on Image Processing (ICIP10)*, 2010, pp. 1649–1652.

[80] H. Mansour and O. Yilmaz, "Adaptive compressed sensing for video acquisition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'12)*, 2012, pp. 3465–3468.

[81] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic mri," *Magnetic Resonance in Medicine*, vol. 59, no. 2, pp. 365–373, 2008.

[82] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (limat) framework for highly scalable video compression," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1530–1542, Dec. 2003.

[83] D. Lam and D. Wunsch, "Video compressive sensing with 3-d wavelet and 3-d noiselet," in *Proceedings of 19th IEEE IEEE International Conference on Image Processing (ICIP12)*, 2012, pp. 893–896.

[84] Z. Gan, L. Qi, and X. Zhu, "Wyner-Ziv coding of video using compressive sensing without feedback channel," in *Proceedings of IEEE 10th International Conference on Signal Processing*, Oct. 2010, pp. 1129 –1132.

[85] Z. Chaozhu and L. Jing, "Distributed video coding based on compressive sensing," in *International Conference on Multimedia Technology (ICMT'11)*, 2011, pp. 3046–3049.

[86] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[87] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems.* V. H. Winston & Sons, 1977.

[88] C. Chen, E. Tramel, and J. Fowler, "Compressed-sensing recovery of images and video using multihypothesis predictions," in *Proceedings of 45th IEEE Asilomar Conference on Signals, Systems and Computers*, 2011, pp. 1193–1198.

[89] Y. Hou and F. Liu, "A low-complexity video coding scheme based on compressive sensing," in *Fourth International Symposium on Computational Intelligence and Design (ISCID'11)*, vol. 2, 2011, pp. 326–329.

[90] C. Qiu, W. Lu, and N. Vaswani, "Real-time dynamic mr image reconstruction using kalman filtered compressed sensing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*, 2009, pp. 393–396.

[91] N. Vaswani, "Ls-cs-residual (ls-cs): Compressive sensing on least squares residual," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4108–4120, 2010.

[92] W. Lu and N. Vaswani, "Modified compressive sensing for real-time dynamic mr imaging," in *16th IEEE International Conference on Image Processing (ICIP'09)*, 2009, pp. 3045–3048.

[93] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4595–4607, 2010.

[94] K. S. N. E. Y. K. H. Jung, K. Sung and J. C. Ye, "k-t focuss: A general compressed sensing framework for high resolution dynamic mri," *Magnetic Resonance in Medicine*, vol. 61, no. 1, pp. 103–116, January 2009.

[95] Y. Zhang, S. Mei, Q. Chen, and Z. Chen, “A novel image/video coding method based on compressed sensing theory,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’08)*, 2008, pp. 1361–1364.

[96] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the h.264/avc standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[97] V. Stankovic, L. Stankovic, and S. Cheng, “Scalable compressive video,” in *IEEE International Conference on Image Processing (ICIP’11)*, 2011, pp. 921–924.

[98] M. Mashud Hyder and K. Mahata, “A scalable distributed video coder using compressed sensing,” in *Annual IEEE India Conference (INDICON’09)*, 2009, pp. 1–4.

[99] A. Aaron, S. Rane, E. Setton, and B. Girod, “Transform-domain Wyner Ziv codec for video,” in *Proceedings of SPIE Visual Communication and Image Processing*, San Jose, CA, USA, Jan. 2004.

[100] A. Aaron, R. Zhang, and B. Girod, “Wyner Ziv coding of motion video,” in *Proceedings of IEEE Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 2002.

[101] A. M. R. Puri and K. Ramchandran, “PRISM: A video coding paradigm with motion estimation at the decoder,” in *IEEE Transactions on Image Processing*, vol. 16, no. 10, Oct. 2007, pp. 2436–2448.

[102] R. Puri and K. Ramchandran, “PRISM: A new robust video coding architecture based on distributed compression principles,” in *Proceedings of 40th Allerton Conference on Communication, Control and Computing*, Monticello, IL, USA, 2-4 October 2002.

[103] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, “The DISCOVER codec: Architecture, techniques and evaluation,” in *Proceedings of Picture Coding Symposium*, Lisbon, Portugal, 7-9 November 2007.

[104] E. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406 –5425, dec. 2006.

[105] E. Candes, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.

[106] F. Sebert, Y. M. Zou, and L. Ying, “Toeplitz block matrices in compressed sensing and their applications in imaging,” in *Proceedings of International Conference on Information Technology and Applications in Biomedicine (ITAB08)*, may 2008, pp. 47 –50.

[107] L. Yu, J. Barbot, G. Zheng, and H. Sun, “Toeplitz-structured chaotic sensing matrix for compressive sensing,” in *7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP10)*, july 2010, pp. 229 –233.

[108] W. M. B. Duarte, Marco F and R. G. Baraniuk, “Fast reconstruction of piecewise smooth signals from random projections,” in *Proceedings of Signal Processing with Adaptative Sparse Structured Representations(SPARS05)*, Nov 2005.

[109] A. Gilbert and P. Indyk, “Sparse recovery using sparse matrices,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 937 –947, june 2010.

[110] R. Berinde, P. Indyk, and M. Ruzic, “Practical near-optimal sparse recovery in the l1 norm,” in *Proceedings of 46th Annual Allerton Conference on Communication, Control, and Computing*, sept. 2008, pp. 198 –205.

[111] http://www.l1-magic.org.

[112] J. BARZILAI and J. M. BORWEIN, “Two-point step size gradient methods,” *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[113] http://www.lx.it.pt/~mtf/GPSR/.

[114] S. Becker, J. Bobin, and E. Candes, “Nesta: a fast and accurate first-order method for sparse recovery,” *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1 –39, 2011.

[115] Y. Nesterov, “Gradient methods for minimizing composite objective function,” *Mathematical Programming*, vol. 140, no. 1, 2007.

[116] http://www-stat.stanford.edu/~candes/nesta/.

[117] R. Gray and D. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.

[118] L. G. Y. C. N. N. Do, T.T. and T. Tran, "Fast and efficient dimensionality reduction using structurally random matrices," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP09)*, april 2009, pp. 1821 –1824.

[119] http://sipi.usc.edu/database/.

[120] R. Reininger and J. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Transactions on Communications*, vol. 31, no. 6, pp. 835 – 839, jun 1983.

[121] S. Hoggar, *Mathematics of Digital Images.* Cambridge, 2006.

[122] http://media.xiph.org/video/derf/.

[123] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the dct coefficient distributions for images," *IEEE Transactions on Image Processing,*, vol. 9, no. 10, pp. 1661 – 1666, oct 2000.

[124] H. R. S. Z. Wang, A. C. Bovik and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," in *IEEE Transactions on Image Processing*, vol. 13, April 2004, pp. 600–612.

[125] http://www.discoverdvc.org/.

[126] ITU, "H.263 : Video coding for low bit rate communication," *ITU-T Recommendations for H.263*, 2005.

[127] http://ffmpeg.org.

[128] http://www.img.lx.it.pt/~discover/home.html.

[129] http://decsai.ugr.es/cvg/dbimagenes/index.php.

[130] http://www.ece.msstate.edu/~fowler/BCSSPL/.

[131] http://home.engineering.iastate.edu/~luwei/modcs/.

[132] http://bisp.kaist.ac.kr/ktFOCUSS.htm.