COMPRESSIVE SENSING BASED MULTIVIEW IMAGE CODING WITH BELIEF PROPAGATION

by

Parmida Beigi

B.Sc., Sharif University of Technology, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in the
School of Engineering Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2011

APPROVAL

Name:	Parmida Beigi
Degree:	Master of Applied Science
Title of Thesis:	Compressive Sensing based Multiview Image Coding with Belief Propagation
Examining Committee:	Dr. Carlo Menon, Assistant Professor, School of Engineering Science, Simon Fraser University Chair
	Dr. Jie Liang, Associate Professor, School of Engineering Science, Simon Fraser University Senior Supervisor
	Dr. Daniel Lee, Associate Professor, School of Engineering Science, Simon Fraser University Supervisor
	Dr. Sami Muhaidat, Assistant Professor, School of Engineering Science, Simon Fraser University Supervisor
	Dr. Mirza Faisal Beg, Associate Professor, School of Engineering Science, Simon Fraser University Internal Examiner
Date Approved:	26 August 2011

Declaration of Partial Copyright Licence

Abstract

Multiview imaging technologies consist of multiple cameras which are usually highly related. In some network settings, it is possible to reduce the operational quality of some cameras yet still achieve high-quality image recovery. Employing low-resolution cameras can greatly decrease the acquisition costs and complexities. The idea of Compressive Sensing (CS) is introduced to accomplish the role of low-quality cameras by operating at a diminished sampling rate. CS imposes a prior distribution on the unknown variables, and allows sparse signal recovery from sub-Nyquist measurements. In this thesis, we investigate the applications of Compressive Sensing via Belief Propagation (CS-BP) theory for low-quality cameras.

In more detail, we take advantage of the side information from neighboring views, in improving the performance of BP-based multiview image recovery. The main issue in the original CS-BP is that all unknown variables have the same prior distribution, which is not true in many cases, especially in transformed data. In this thesis, we investigate the applications of multiview technology along with methods on the generalization of the CS-BP.

To further improve the CS-BP, we explore the role of larger coefficient of the signal in assigning the pdf sampling step-size. As large coefficient are dominant in step-size determination, the greater the large components are, the less accurate the small components detection is. Thus, we propose methods which deal with DC and other large coefficient to attenuate their influenc on the sampling step-size. The proposed method greatly improves the accuracy of signal recovery, as the sampling step-size is maintained at a reasonably small value.

In addition, we evaluate the number of large coefficient that are to be eliminated from BP iterations, by introducing an adaptive technique which determines the optimum number of coefficient according to the involving costs and complexities. Application of compressive sensing in multiview technology is relatively a new idea and the experimental results show that the generalized CS-BP can greatly outperform the original CS-BP technique.

Acknowledgments

Foremost, I would like to express my sincerest gratitude to my supervisor, Prof. Jie Liang, for his kindness, guidance and patience during my MASc studies. His guidance and support in my research not only made the research life smoother for me, but also inspired me in thinking of new research ideas. Besides my research supervisor, I would like to thank the rest of my defense committee, Prof. Mirza Faisal Beg, Prof. Daniel Lee, Prof. Carlo Menon and Prof. Sami Muhaidat.

I also thank my wonderful lab mates especially Xiaoyu Xiu, Yu Gao and Xiaozheng Huang for their helpful feedbacks and their friendships in making lab a fun for me. Their useful suggestions and positive inputs gave me encouragement in my research experiments.

Last but not least, my deepest gratitude goes to my lovely family for their continuing love and care throughout my life. I would like to specifically thank my mother, father and my two brothers, Bardia and Arshia, who supported me in every steps of my studies and provided me with helpful advices. My mom's support was indeed a key element in me deciding to pursue my MASc studies in Canada.

Contents

List of Tables

List of Figures

Acronyms and Abbreviations

Acronym	meaning
BP	Belief Propagation
BCS	Bayesian Compressive Sensing
CS	Compressive Sensing
CS-BP	Compressive Sensing via Belief Propagation
CS-MIC	Compressive Sensing Multiview Image Coding
DCT	Discrete Cosine Transform
EM	Expectation Maximization
KLT	Karhunen-Loeve Transform
LDPC	Low Density Parity-Check
MAP	Maximum A Posteriori
MDBP	Message Damped Belief Propagation
MIC	Multiview Image Coding
MMSE	Minimum Mean-Squared Error
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
MVC	Multiview Video Coding
NP-hard	Non-deterministic Polynomial-time hard
OMP	Orthogonal Matching Pursuit
PSNR	Peak Signal to Noise Ratio
RIP	Restricted Isometric Property
RVE	Rectification-base View Extrapolation
RVI	Rectification-base View Interpolation
RVM	Relevance Vector Machines
SI	Side Information

Chapter 1

Introduction

1.1 Introduction

With the advanced camera technologies and consumer demands for wider visual experiences, multiview images/videos, *i.e.*, multiple pictures captured by several cameras around a target scene, has drawn a lot of attention. This setup can be used in a variety of applications that may also be working with single view images, yet improving the performance and user satisfaction. These applications include: Free Viewpoint Television, 3DTV, virtual view synthesis, high performance imaging, object tracking and surveillance, *etc* [38, 39]. In multiview image/video applications, variety of factors are involved which requires consideration. Firstly, a system is to be designed, capable of taking and storing huge numbers of multiview image/video data; which can be both costly and complicated. In addition, some arrangements such as sufficien number of cameras and their suitable locations make this process further challenging. In general, better quality and larger navigation ranges is available by using higher quality cameras as well as large numbers of them, yet increasing the computational costs and complexities. Therefore, there is a trade-off between costs/complications of cameras, setup, accessories, and the system performance including image qualities and their spatial covered ranges.

One of the important factors as mentioned above is the storage, acquisition and transfer of the multiview data, which can be quite costly and inefficient *video conferencing* is one example. Therefore, multiview image/video compression techniques (MIC/MVC) can be used in order to improve these systems' efficien y.

1.2 Multiview Image/Video Compression

Since all of the cameras capture the same scene, the captured images/videos have considerable redundancy, using this redundancy at the encoding procedure is the thought behind almost all of the MIC/MVC methods. Traditional image compression methods exploit this redundancy based on the spatial distribution of pixels or the limitations of the human understanding structure, where multiview compression techniques can combine an additional dimension of redundancy because of the overlap between the multiple views in the scene. A common way to exploit statistical dependencies among multiple views is the disparity compensation method. In this approach, one view (some blocks of it) is predicted from the other views by findin the best matching block in the neighboring views; finall the error between the correct view and the prediction one is encoded. View synthesis prediction (VSP) is another method, in which the geometrical relationship between different views is used to produce a synthesized view for the target view. The synthesized view is then used as an additional reference of prediction for encoding the target view. Among the VSP methods, some of them use depth estimation, including information on cameras pattern [48], while some of them do view interpolation-based prediction without use of depth information [47, 46].

1.2.1 Rectification-based View Interpolation for MVC

In this thesis, we are interested in the view synthesis prediction-based multiview video coding based on rectification-base view interpolation (RVI) proposed in [46]. A robust method is used to adjust the two reference views (left and right) which results in a reduction of their vertical mismatches. The block diagram of this method is presented in Fig. 1.1.

In general, in most view synthesis methods, view interpolation is acquired based on the left view and the right view, which can only be applied to half of the views. In order to cover all of the views, in [46] RVI is modifie to rectification-base view extrapolation (RVE) method, which uses two left or two right views. At last, the RVE method is applied to MVC to encode all views after the firs two views. The procedure in RVE is almost similar to RVI; the extrapolation algorithm firs performs projective rectificatio and disparity estimation to the two left views (instead of a right and a left view). The difference is in the next step, as instead of interpolating the disparity to fin the corresponding pixel locations in the middle view, in RVE the disparity is extrapolated and the pixel locations in the right view is estimated after. The fina step of the process is still the same, and it will apply the RVE to MVC, which will encode all views after encoding the firs two views.

In addition to the above, the acquisition cost of multiview dataset can further be reduced if only

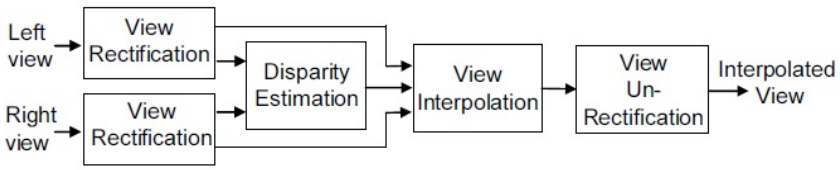


Figure 1.1: Block diagram of the RVI algorithm [46].

some of the views are sampled at high resolution, and other views are sampled at low quality. In this thesis, we have investigated the application of the Compressive Sensing (CS) theory in this hybrid multiview image acquisition scenario [12]. Compressive sensing which is closely related to the problem of solving an under-determined system of linear equation under a sparseness constraint is described next.

1.3 Compressive Sensing

In conventional applications, firs the N-dimensional data $\mathbf{x}$ is measured (encoded), then the encoded signal will be compressed (using wavelet, DCT, *etc*), and the compressed set of basis-function coefficient are stored. However, based on the signal pattern (such as sparsity), at this stage a large fraction of the transform coefficient may be useless and therefore thrown away, while still achieving accurate data reconstruction. This seems wasteful, since there are many applications for which data collection is expensive or time-consuming (such as MRI). The fact that after compression, a large fraction of data are possibly discarded, raised the following question: Why not use compressed measurements directly, *i.e.*, measure only the informative part of the data while reducing the measurement costs and complexity of the encoder system at the same time? This question introduced a new fiel named Compressive Sensing [12]. The idea is to develop methods by using lower number of required samples *i.e.*, perform compression exactly at the time of sampling, which results in an efficien and less complex system. Although signal recovery from the condensed measurements appears to be an ill-posed inverse problem because there are more unknowns than the observations, imposing a prior knowledge of sparsity on the signal, will make this setting reasonable and solvable even when having extrem underdetermined systems of equations.

The Shannon (Nyquist) sampling theorem states that to avoid losing information when sampling a signal, the number of the samples must be at least two times greater than the signal bandwidth. Compressive Sensing however, allows almost perfect recovery of *sparse* signals from far fewer

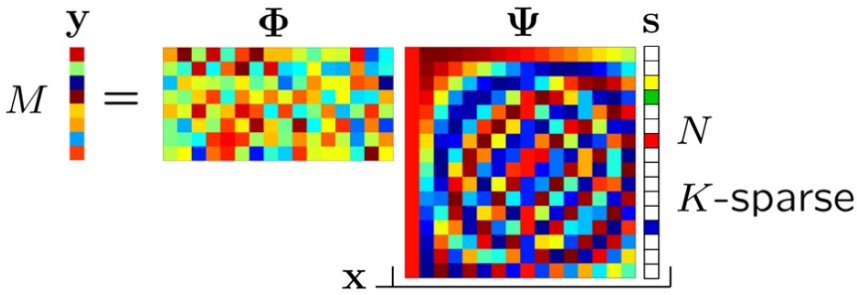


Figure 1.2: Compressive Sensing encoding process [2].

samples than that required by the Nyquist rate, under some conditions. It is important to note that, compressive sensing does not violate the Nyquist sampling theorem as CS reconstruction is based on the prior knowledge of the sparsity of the signal, and therefore, it is not able to reconstruct non-sparse signals. By CS, as described in Fig. 1.2 the N-length signal $\mathbf{x}$ can be reconstructed from the M-length observation $\mathbf{y}$ ($M \ll N$); where $\mathbf{y} = \mathbf{\Phi x}$, and $\mathbf{\Phi}_{M \times N}$ is the projection (measurement) matrix. If $\mathbf{x}$ itself is not sparse (a signal that most of its components are zero), a proper transform that represents $\mathbf{x}$ in a sparse format, should be used in order to CS theory. More specificall , applying the transform, will represent a general signal (in the spatial domain) into a sparse signal (in the transform domain), which can then be applied to the CS algorithm. Consider that the general signal is sparse in basis $\mathbf{\Psi}$, *i.e.*, $\theta = \mathbf{\Psi x}$, (where θ is a sparse vector), then a solution for θ can be found if $\mathbf{\Phi}$ and $\mathbf{\Psi}$ satisfy the incoherence condition, also known as *Restricted Isometric Property* (RIP) [8].

$$1 - \delta_k \leq \frac{||\mathbf{\Phi}.\,\theta||_2^2}{||\theta||_2^2} \leq 1 + \delta_k \tag{1.1}$$

where $0 \leq \delta_k \leq 1$ is the isometry constant.

RIP states the existence of robust algorithms to exactly reconstruct the sparse signals from their compressed measurements. More specificall , it guarantees that none of the K-sparse inputs, *i.e.*, signals with $K \ll N$ large coefficients fall in the null space of the sampling matrix.

There is a general criteria on the number of required samples (the number of columns of the measurement matrix) when sampling a sparse signal, which depends on the knowledge about the locations of non-zero coefficient in the transform domain. If the location of non-zero elements is known, then the number of samples in the space (time) domain is required to be at least equal to the number of non-zero coefficient (Nyquist rate). If the location of non-zero elements is unknown however, the number of needed samples is at least twice the number of non-zero coefficients For

both cases, depending on the type of sparsity (random, lowpass or bandpass) and the type of sampling (random, uniform or periodic nonuniform), we need to sample more than the minimum rate to guarantee stability (a characteristic of sampling matrix obtained by RIP). Another way to guarantee stability is to use combinations of samples rather than the direct sampling of the signal, which will ease the possibility of designing new methods to capture and represent compressible signals at a rate significantl below the Nyquist rate [25]. Considering that, a sparse signal can be reconstructed from its compressed samples with a probability of almost one, if:

$$M \geq c\, k \log(\frac{N}{k}) \tag{1.2}$$

where c is a positive constant [7].

The inequality in (1.2) can be further described as follows [5]:

$$M \geq \mu^2(\mathbf{\Psi}, \mathbf{\Phi})\, \frac{k \ln n}{c} \tag{1.3}$$

where $\mu(\mathbf{\Psi}, \mathbf{\Phi})$ is the maximum coherence between the rows of $\mathbf{\Psi}$ and $\mathbf{\Phi}$.

Hence, it is clear that in order to decrease the number of required samples, one should search for the matrix $\mathbf{\Phi}$ that have low coherence with $\mathbf{\Psi}$.

In [12], it is shown that the Gaussian random matrix is largely incoherent with any f xed basis $\mathbf{\Psi}$. More precisely, the inequality in (1.3) is closely related to RIP, as if $\mathbf{\Phi}$ is a Gaussian random matrix satisfying (1.3), $\mathbf{\Phi}.\mathbf{\Psi}$ is also a Gaussian random matrix with the same number of rows, thus satisfie RIP. Therefore random matrices (such as Gaussian, Binary, *etc*) can be a proper choice for the measurement matrix $\mathbf{\Phi}$ [33]; whereas, in many applications a more sophisticated sampling matrix is desired, such as structured matrices [6] or sparse matrices, which can further improve encoding and decoding [21].

Sparse coding, which accounts for the methods involved in findin the sparse representation of general signals, is widely used in areas such as signal processing, statistics and machine learning. Dictionary learning is a common approach accounts for designing the sparsifying according to the original non-sparse signal [20]. Considering the signal x in R_N, we can fin a sparse representation of it over a *dictionary* D in $R_{N\times K}$; where K $K \ll N$ columns referred to as *atoms*. Mallat was the firs one who used the wavelet in linear decomposition of signals using the atoms of a learned dictionary rather than a predefine one [23].

The CS reconstruction procedure is to fin the sparsest solution (with the smallest ℓ^0-norm) from the observed measurements:

$$\min_{\theta} ||\theta||_0 \qquad \text{subject to} \qquad \mathbf{y} = \mathbf{\Phi}'\theta \tag{1.4}$$

where $\mathbf{\Phi}' = \mathbf{\Phi}\mathbf{\Psi}$.

Since ℓ^0-norm optimization is NP-hard [14], various alternative methods have been proposed to solve the problem in a tractable way, ℓ^1-norm optimization based method is an example [20], [10]. In general, depending on several factors, such as the sparsity pattern, the type of sampling, complexity issue, and speed matter, different reconstruction algorithms can be used, which can be generally categorized into three groups [25]:

- **Geometric Methods**: The old methods which use ℓ^1-normminimization techniques for reconstructing a K-sparse signal from a set of $M = O(Klog(\frac{N}{K}))$ measurements [12] are geometric. It is good to mention that replacing ℓ^1-norm with the faster implemented ℓ^2-normwill not always produce reasonable answers, although there are iterative methods which have faster performance compare to ℓ^1-norm.

  Knowing the position and amplitude of samples as well as the location of sparsity in the transform domain, applying iterative techniques between these two domains will result in the original signal [13]. As mentioned before in the case of unknown sparsity positions, we need more samples in order to evaluate the number of sparse coefficients the position and the values of nonzero coefficients modifie iterative methods are used for this purpose, which are greedy algorithms; Matching pursuit that recovers the K-sparse N-dimensional signal after N steps [24] Basis pursuit which decompose signals into a linear expansion of waveforms that are chosen from a redundant dictionary and reconstructs the signal using ℓ^1-norm[36] are some examples.

- **Combinatorial**: This is another standard category in reconstructing signals from compressed samples. According to this section, the measurement matrix $\mathbf{\Phi}$ is generally formed by Binary entries and is define using *bipartite* graphs; a famous encoding system using this approach, is Low Density Parity Check (LDPC), which provides fast coding process. Message passing technique using graphical model is then used for reconstructing the signal, Belief Propagation (BP) decoder is commonly used for this purpose. BP basically identifie the column of $\mathbf{\Phi}$ that have maximum correlation with the measurement vector $\mathbf{y}$ [18].

- **Information Theoretic Methods**: This category contains more recent approaches in CS reconstruction. In this method, the signal $\mathbf{x}$ is assumed to be an instance of a vector random

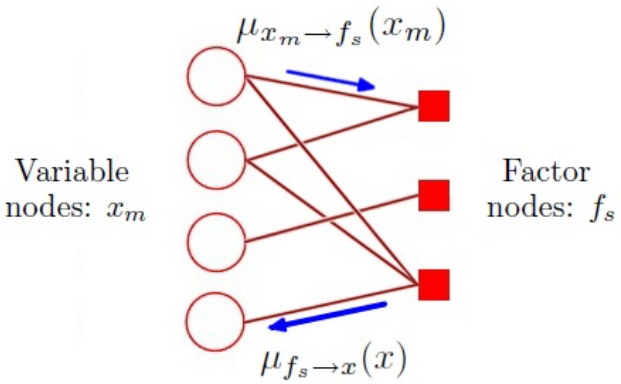


Figure 1.3: Description of a Bipartite graph with 4 variable nodes x and 3 factor nodes f_s

variable, and the m^{th} row of the measurement matrix $\mathbf{\Phi}$ is constructed using the value of the previous measurement y_{m-1}. The theory of Huffman coding is used to construct the binary sampling matrix such that it minimizes the average number of required measurements [1].

Thus, as mentioned, there are several encoding and decoding algorithms proposed in the literature, among them is *Belief Propagation* decoder along with sparse LDPC-based CS encoder which lead to a fast manipulation, is the area of our interest as well.

1.3.1 Belief Propagation

Belief Propagation is a message passing algorithm invented in 1982 to calculate the marginal distributions in Bayesian networks and Markov random fields It can be used to fin marginal distributions or to obtain estimates such as MAP, MMSE, *etc*. BP can exactly fin the marginal distributions if it is not loopy, *i.e.*, its corresponding graph does not contain any loops (tree, chain, *etc*), otherwise, it will provide estimates for the distributions. Belief Propagation technique is thus considered as a proper choice for signal reconstruction, and can be used as a fast decoder in Bayesian frameworks [3].

Belief Propagation operates on a factor graph named Bipartite graph, which is composed of two disjoint types of nodes: variable and factor nodes. The edges in this graph only exist between nodes of opposite type, *i.e.*, there is no edge between a node and the nodes in its own group.

Belief Propagation can be basically described as a message passing algorithm. Messages are

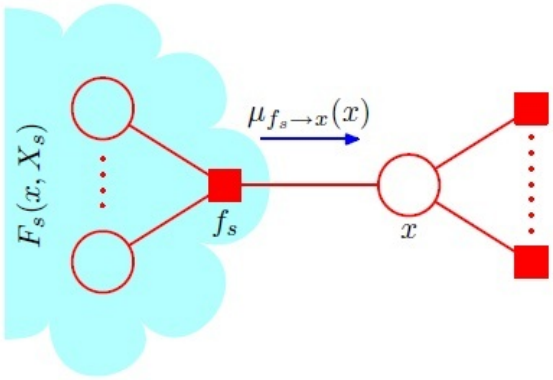


Figure 1.4: A section of a factor graph, illustrating the evaluation of marginal distribution f(x) [4].

actually real functions corresponding to probability distributions on the variable nodes, which are traveling among the nodes of different groups. According to Fig.1.3, there are basically two types of messages:

- $\mu_{x \to f_s}(x)$: the message sent from a variable node x to one of its neighbors in the bipartite graph, a factor node f_s. According to the "sum-product" algorithm [4], this message is computed by taking the product of all the received messages on node x (except the message coming from node f_s), this procedure is called *multiplication* of beliefs at variable nodes:

$$\mu_{x_m \to f_s}(x_m) = \prod_{l \in n(x_m) \setminus \{f_s\}} \left[\sum_{X_{ml}} F_l(x_m, X_{ml}) \right] = \prod_{l \in n(x_m) \setminus \{f_s\}} \mu_{f_l \to x_m}(x_m) \qquad (1.5)$$

- $\mu_{f_s \to x}(x)$: the reverse message is computed similarly, but the corresponding constraint of the factor node f_s should be considered and the result will be marginalized at the end, this is called *convolution* at check nodes.

$$\mu_{f_s \to x}(x) = \sum_{X_s} F_s(x, X_s) = \sum_{x_1} \dots \sum_{x_M} f_s(x, x_1, \dots x_M) \prod_{m \in n(f_s) \setminus \{x\}} \mu_{x_m \to f_s}(x_m) \qquad (1.6)$$

where $n(x)$ and $n(f_s)$ are sets of neighbors of x and x_s, respectively. As it is shown marginalization is taken over all the variable nodes connected to f_s except x.

Finally, as shown in Fig. 1.4 the marginal distribution $f(x)$ for each variable node is obtained from the product of all the incoming messages directed towards that node,

$$f(x) = \prod_{s \in n(x)} \left[\sum_{X_s} F_s(x, X_s) \right] = \prod_{s \in n(x)} \mu_{f_s \to x}(x) \tag{1.7}$$

This thesis is mainly focused on the applications of Compressive Sensing especially the Belief Propagation technique in multiview image systems. Being able to make some cameras to function at lower quality, can extremely decline the acquisition, storage and transmission costs. This can be achieved by applying compressive sensing framework to these views, *i.e.*, we capture the data of some views in a diminished sampling rate thus great amount of costs will be reduced, while other views are still being acquired in high quality.

1.4 Main Contribution

In this thesis, we apply Compressive Sensing using a sparse encoding and a Belief Propagation decoding scheme, in multiview image coding system. Employing CS in multiview applications is quite new and there are not a lot contributions on this topic in the literature [22, 9]. Although the idea of this compressive sensing framework was initially developed in [3] named (CS-BP), this method assumed the original sparse signals which is not a proper assumption for general images. In addition, according to CS-BP assumption, all unknown variables in the BP decoder have the same prior distribution, which is not true in many cases, especially images. In this thesis we propose a generalization algorithm which is capable of handling different prior distributions, general input data. In addition, we make use of side information at the belief propagation decoder in multiview scheme.

We propose an algorithm to improve CS-BP scheme, which deals with the largest coefficien (DC) of N-dimensional signal differently by removing its contribution on DC effected measurements, and updates the measurements after iteratively. At the last step, from the $N-1$ recovered coefficients the DC coefficien is calculated. Moreover, we further improve CS-BP scheme, by dealing with larger signal coefficient one after another in an iterative manner. In this case by diminishing the larger coefficient effects on the whole signal coefficient recovery, we can even further improve the performance of the CS-BP systems.

In addition, we develop a decoding scheme, which can take advantage of the side information (SI) of left and right views around a target view. By this scheme, only the left and right views need

to be gathered in high quality and the target view exploits side information from them and can be obtained using lower quality sampling schemes. Experimental results show that these modification greatly improve the performance of the original CS-BP algorithm. Part of the algorithms and the results presented in this thesis have been published in [45].

1.5 Thesis Outline

Chapter 2 describes some fundamental methods on Bayesian compressive sensing techniques, especially the CS-BP algorithm. Chapter 3 introduces the firs step of the proposed method, which is application of CS-BP by utilizing side information in multiview image coding. In Chapter 4, two methods are proposed as the generalization of the CS-BP, which deals with DC component of data efficientl and significantl outperforms the recovery results. Chapter 5 presents two further techniques and adaptive algorithm which deal with larger coefficient profoundly. Finally, in Chapter 6 conclusions and discussions on the future works are explained.

Chapter 2

Background

2.1 Introduction to Bayesian Compressive Sensing

The Bayesian approach to the statistical problems is a probabilistic technique that jointly describes the unknown parameters as well as the observed data. Generally, the probability distribution on the correctness of a "hypothesis" conditioned on the observed data is used to interpret a correct choice amongst different choices. More specificall , the probability of a hypothesis is evaluated by definin some prior probabilities, which eventually is updated to form a "posterior" probability in obtaining the best choice. Bayesian model selection is a technique which selects the statistical models of an observed data based on "Bayes" factors. In Bayesian modeling, all unknown coefficient are assigned a prior probability distribution conditioned on the observations. In a simple model selection scenario for example, the best choice should be obtained out of two models based on the observed data (D). The comparison between the two models M_1 and M_2 is define as follows:

$$\text{Bayes factor} = \frac{P(D|M_1)}{P(D|M_2)} = \frac{\int P(\theta_1|M_1)P(D|\theta_1, M_1)d\theta_1}{\int P(\theta_2|M_2)P(D|\theta_2, M_2)d\theta_2} \tag{2.1}$$

where θ_1 and θ_2 are model parameter vectors.

Compressive Sensing algorithms rely on the sparsity of the target signal $\mathbf{x}$, whether in its own or another basis ($\mathbf{x} = \mathbf{\Psi}\theta$). In many applications the statistical characteristics of the signal $\mathbf{x}$ is available in advance. In these cases, Bayesian inference can provide more precise estimation of x and reduce number of CS measurements. Ji *et al.* [35] proposed a Bayesian compressive sensing technique, where signal estimation is done using the *Relevance Vector Machines* (RVM). In RVM, rather than imposing a Laplace prior on θ, the sparse representation of $\mathbf{x}$, a hierarchical prior is

used, which has similar properties as the Laplace prior yet allows convenient conjugate-exponential analysis, an important factor in Bayesian framework. Based on this method, in addition to reconstructing the sparse signal as the estimation of the original signal, "error bars" are also provided; hence, it provides us a measure of confidenc in the reconstructed signal. In addition, by using the knowledge of the error bars one can determine when a sufficien number of compressive-sensing measurements have been used. Babacan *et al.* [34] also proposed a hierarchical Bayesian approach, which uses Laplace prior to model sparsity of the unknown data. Compared to most of the methods in the literature, the advantage of this method is that model parameters are not required to be tuned to the data and they can be estimated along with the unknown signal coefficients The method of sparse Bayesian learning [11] adopts a Bayesian framework, where x_i are independent, and zero-mean Gaussian distributions with unknown variance σ_i^2 are assigned to them. The unknown variances are then given the Gamma conjugate prior, and expectation maximization (EM) will iteratively compute a MAP estimate of $\mathbf{x}$. The primary focus in the literature can be seen on the detection of the few significan entries of the sparse signal known as basic selection, in [29] however, Schniter *et al.* have proposed a method which adopts a minimum mean-squared error (MMSE) estimation framework and maximum likelihood approach by a weighted mixture of the most likely models. *Multiuser decoding* techniques have also used Bayesian theory in signal estimation. As most of the users are inactive in this framework, the setup is similar to the compressive sensing decoding structure. In [17] for example, multiuser decoding is developed using sparse spreading sequences and belief propagation decoder. Baron *et al.* [3] also make use of sparse encoding matrices belief propagation decoding, which is our interest in this thesis.

2.2 Bayesian Compressive Sensing (BCS) Using Hierarchical Prior

In this technique, by the knowledge of $\mathbf{\Phi}$, the quantities to be estimated based on the CS measurements are the coefficient of θ and the noise variance σ_n^2 (in the case of noisy measurements $\mathbf{y} = \mathbf{\Phi}\theta + \mathbf{n}$). As mentioned before, the advantage of using hierarchical priors than the Laplace prior for example, is in that hierarchical prior is conjugate[1] to the Gaussian likelihood, thus the corresponding Bayesian inference can be derived in closed form [35]. To better discuss the algorithm, consider the graphical representation of the model as shown in Fig. 2.1. Whereas the accuracy of this method is not as well as some other techniques, the main advantage of using RVM techniques

[1] In Bayesian theory, a prior probability distribution $p(\theta)$ is said to be conjugate with a likelihood function $p(x|\theta)$ if the resulting posterior distribution $p(\theta|x)$ is in the same family as $p(\theta)$ [35]

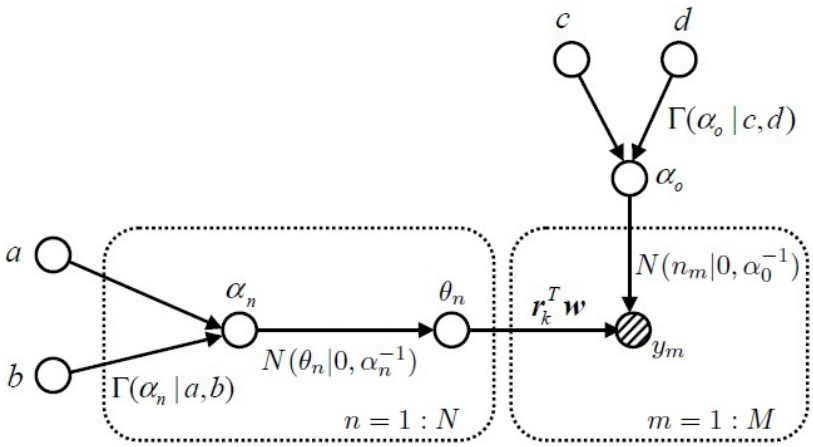


Figure 2.1: Graphical model of the Bayesian CS scenario [35]

is its efficien computation as described bellow.

2.2.1 Bayesian CS inversion via RVM

As described in Fig. 2.1, there are parameters $\alpha_i = 1/\sigma_i^2$, the inverse-variance of signal coeffi cients prior and α_0, the inverse of noise variance are known. A zero mean Gaussian distribution is assigned as the prior for each signal coefficien θ_i. In addition, Gamma distribution is generally used as the prior for α, since Gamma distribution is the conjugate prior for Gaussian distribution (for the existing noise). Finally by marginalizing over all N coefficient of α, the overall prior of θ conditioned on a and b can simply be obtained. Consider the CS measurements $\mathbf{y}$ and the projection matrix $\mathbf{\Phi}$, the posterior for θ is then expressed as a multivariate Gaussian distribution with mean $(\mu = \alpha_0 \sum \mathbf{\Phi}^T y)$ and covariance $(\sum = \alpha_0 \mathbf{\Phi}^T \mathbf{\Phi} + \mathbf{A})^{-1}$, where $\mathbf{A} = \text{diag}(\alpha_1, ..., \alpha_n)$.

In this method, it can be seen that most of the α_i tend to infinit which correspond to the small coefficients therefore only a few numbers of θ_i for which the corresponding α_i remains relatively small, contribute for representation of $\mathbf{y}$ and the level of sparseness (size of k) is determined automatically. Also, it is shown that there is no need to set a, b, c and d on the Gamma hyperpriors, thus uniform hyperpriors on α and α_0 have been used [35]. Since calculation via RVM is slow for large dimension $\mathbf{x}$, a fast RVM algorithm has been developed which achieves highly efficien computations. This technique operates in a constructive manner, *i.e.*, it eventually adds (or deletes) candidate basis function to the model until all the "relevant" basis functions (for which the corresponding weights are nonzero) have been included. Thus, the complexity of the algorithm is more

related to k than N (which makes it more efficient)

2.2.2 Selecting Projections to Reduce Signal Uncertainty

Previous CS algorithms focused on estimating θ (and hence $\mathbf{x}$) using point estimation. These approaches do not provide a measure of uncertainty in $\mathbf{x}$; therefore, adaptive design of $\mathbf{\Phi}$ was not possible. However, the BCS algorithm, specificall the fast RVM algorithm allows efficien computation of $\mathbf{x}$ and associated error bars with the goal of reducing uncertainty, and determining the enough number of measurements for reliable signal reconstruction. In other words, in the case that the change in uncertainty is not significan it is possible that simply the noise is reconstructed and adaptive sensing may be stopped. There are also some comparisons between BCS and other CS methods (such as orthogonal matching pursuit (OMP)), which indicate that the adaptive CS may be one of the unique advantages of BCS over other CS algorithms.

2.3 Bayesian Compressive Sensing using Laplace Priors

In this model [34], a hierarchical form of the Laplace prior is assigned to model the sparsity of the unknown signal $\mathbf{x}$, which is shown as a generalized case of RVM. Two algorithms resulting from this model are presented; one global optimization algorithm and one constructive (greedy) algorithm designed for fast reconstruction which is useful in practical settings. Unlike most existing CS reconstruction methods, both algorithms are "fully-automated", *i.e.*, the unknown signal coefficient and the model parameters are estimated separately from the observations and therefore no user-intervention is needed. Additionally, these algorithms can provide estimates of the uncertainty of the reconstructions.

2.3.1 Bayesian Modeling Using Laplace Priors

In order to deal with the problem that the Laplace distribution is not conjugate to the observation model which is Gaussian (due to the existing noise), similar to [35], the solution is to model it in a hierarchical way by utilizing Gamma distribution and definin some additional prior distributions (hyperpriors). The dependencies in the joint probability model $p(\theta, \alpha, \lambda, \alpha_0, \mathbf{y})$ are shown in Fig. 2.2. Where λ is the Laplace parameter, and υ the hyperparameter to defin λ ($p(\lambda|\upsilon)$). Similar to other Bayesian techniques, the goal here is to fin the posterior distribution $p(\theta|\mathbf{y}, \alpha, \alpha_0, \lambda)$, which is multivariate Gaussian distribution $\mathcal{N}(\mu, \sum)$ with mean and covariance define as before,

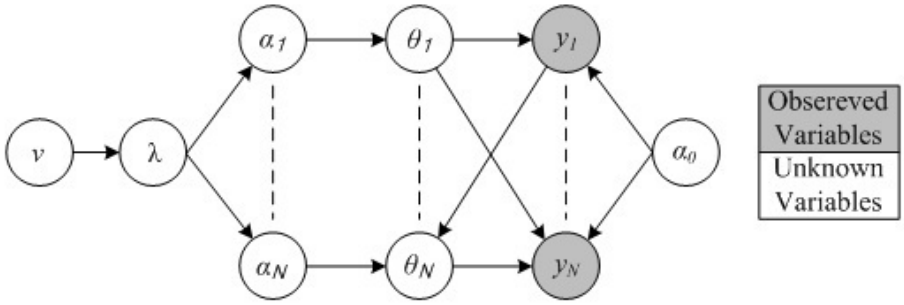


Figure 2.2: Directed acyclic graph representing the Bayesian model [34]

i.e., $(\mu = \alpha_0 \sum \mathbf{\Phi}^T \mathbf{y})$ and $(\sum = \alpha_0 \mathbf{\Phi}^T \mathbf{\Phi} + \mathbf{A})^{-1}$, where $\mathbf{A} = diag(\alpha_1, ..., \alpha_n)$. The idea here is to iteratively estimate the distribution of θ, given an estimate of α, α_0 and λ. Analysis shows, however, that this method has some disadvantages such as number of required calculations, numerical errors, *etc*. Hence, a practical algorithm has been presented, in which, to promote sparsity and to decrease the computational requirements, only a single α_i will be updated at each iteration of the algorithm instead of updating the whole vector α. Moreover, it is described that by starting with an empty model ($\alpha = 0$) and iteratively adding components to the model, the algorithm will become more efficient Compared to the separate Gaussian priors employed on the entries of θ in the RVM framework, Laplace priors enforce the sparsity constraint more heavily by distributing the posterior mass more on the axes so that signal coefficient close to zero are preferred. In addition, the Laplace distribution promotes sparsity to the largest extent due to being log-concave, providing the useful advantage of eliminating local minima while leading to unimodal posterior distributions as well. Based on the results obtained from a particular image, although BCS and Laplace have nearly the same error rate, Laplace is faster and the reconstructed image is sparser [34].

2.4 Sublinear Compressive Sensing Reconstruction via Belief Propagation Decoding

This compressive sensing scheme is based on codes of graphs, which allows for joint design of sensing matrices and low complexity reconstruction algorithms. The designed compressive sensing matrix can be shown to perform asymptotically optimal when used in combination with OMP methods [18]. As mentioned before, in CS theory, ℓ^0 optimization problem is NP-hard; therefore, other

methods such as Linear Programming have been considered so far, which do not have fast speed (the important factor in some applications even if more measurements are needed). The focus of attention in this work is for low computational complexity and faster algorithms. In this regard, Belief Propagation (BP) algorithm is used which operates on the columns of the measurement matrix $\mathbf{\Phi}$ which have the most correlation with the measurements $\mathbf{y}$. In this approach, the major challenge to reconstruct the sparse signal is to determine in which subspace, generated by not more than K (a define value based on the Restricted Isometry Property (RIP) method) columns of the matrix $\mathbf{\Phi}$, the measured signal lies in. Once the correct subspace is obtained, the non-zero signal coefficient are calculated by applying the pseudo-inversion technique. The basic idea here is to fin the support (set of indices of the non-zero elements of $\mathbf{x}$) of the unknown signal $\mathbf{x}$ sequentially; in each iteration, the candidate column indices are used as current estimates of the support set of $\mathbf{x}$, until the indices in the correct support set are included in the estimated support set [18].

2.4.1 Problem Analysis Using LDPC codes

Normalized code words of Low Density Parity-Check (LDPC) are used as the columns of the measurement matrix $\mathbf{\Phi}$. As mentioned before, this algorithm is combined with OMP to improve its performance. Note that standard OMP algorithm guarantees exact recovery of the signal $\mathbf{x}$ as long as $\mu \leq 1/2K$, where μ is the mutual coherence of the matrix $\mathbf{\Phi}$ and K is the parameter define by RIP for the columns of $\mathbf{\Phi}$ [41]. Hence, in order to satisfy this condition, for all signals, we need to identify LDPC codes under the condition $\frac{1}{2} - \frac{1}{4K} < \frac{d_H(c_i,c_j)}{n} < \frac{1}{2} + \frac{1}{4K}\ \forall\, i \neq j$. The most important part of the CS recovery based on LDPC sensing matrices is that the columns of maximum correlation correspond to the most likely codewords, which can then be efficientl determined by applying iterative methods such as BP. It is worth noting that the suitable BP decoders in this framework must be able to handle high interference noise (in identifying a column ϕ_k, the remaining columns act as interference) and determine sets of most likely codewords. For $0-1$ (binary) $\mathbf{x}$, BP decoder has the purpose to identify the column of $\mathbf{\Phi}$ that maximizes the correlation with the measurement vector $\mathbf{y}$. By adjusting different entries of $\mathbf{y}$, different BP decoder outputs may be produced; therefore, a list of potential columns will be basically obtained. Therefore, the idea is that in the case that the output list has more than K codewords, the algorithm will output only K codewords from the list that have largest correlations with $\mathbf{y}$. Otherwise, if the list has less than K entries, the whole list can be considered [18]. The algorithm summary of BP and BP-OMP is described in Table 2.1.

Table 2.1: BP algorithm in (a) and in combination with OMP in (b) [18]

Algorithm 1 List-based BP Algorithm

Input: K, $\mathbf{H}$, y, biasing list size L, biasing value B.

- Construct the list $\mathbf{RX} = (\mathrm{rx}^{(1)}, \mathrm{rx}^{(2)}, \ldots, \mathrm{rx}^{(L)})$, where
  - Set $\mathrm{rx}^{(1)} = \mathrm{y}$
  - Set $\mathrm{rx}^{(2)}$ is the vector y with entries $\pm K$ being biased to $\pm B$
  - Randomly generate an index set $\mathcal{J}$ of size $(L-2)/2$, where $rx_j \neq \pm K, \forall j \in \mathcal{J}$.
  - For each $j \in \mathcal{J}$, bias the j^{th} coordinate of the vector $\mathrm{rx}^{(2)}$ to $+B$ or $-B$ to get $\mathrm{rx}^{(2l+1)}$ and $\mathrm{rx}^{(2l+2)}$, respectively, $(l = 1, 2, \ldots, (L-1)/2)$.
- Run the BP algorithm for $\mathrm{rx}^{(i)}$ and output a binary word $\hat{\mathrm{v}}_i$ $(i = 1, \ldots, L)$. Delete all the words which do not satisfy $\mathbf{H}.\hat{\mathrm{v}}_i = 0$ and delete all the repetitions in the list.
- Output at most K words $\hat{\mathrm{v}}_i$ whose BPSK images have largest correlations with $\mathbf{y}$.

Output: The list of potential columns of $\mathbf{\Phi}$ in the superposition.

(a)

Algorithm 2 BP-OMP Algorithm

Input: K, $\mathbf{H}$, y, L, B.
Initialization: $\mathrm{rx} = \mathrm{y}$.
Iteration: Run the iteration K times

- Run the list-based BP algorithm and output a list of potential columns
- Pick the codeword $\mathbf{v}$ whose BPSK image has largest correlation with $\mathbf{rx}$
- Update: $K = K - 1$, $\mathrm{rx} = \mathrm{y} - \mathrm{BPSK}(\mathrm{v})$

Output: The list of K columns of $\mathbf{\Phi}$ in the superposition.

(b)

2.5 Bayesian Compressive Sensing via Belief Propagation (CS-BP)

In this Bayesian scheme, a sparse encoder matrix $\mathbf{\Phi}$ and Belief Propagation (BP) decoder is utilized with the aim of accelerating both the encoding and decoding procedures. The Bayesian inference is approximately accomplished using BP decoder which perform the encoding on graphical models. Using sparse encoding matrices can further increase the speed of computational analysis as the size of the graphical model, *i.e.*, the number of involved "edges" will decrease. A kind of a Low Density Parity Check (LDPC) code is used for the encoding matrix $\mathbf{\Phi}$, which is sparse and therefore results in fast computations. This matrix can be represented by a bipartite graph, where BP decoder operates on, using message passing technique. Due to sparsity, the number of loops in the graph decreases thus enhancing the convergence of the message passing technique. In order to decode a k-sparse N-dimensional signal, the CS-BP requires $M = O(K \log(N))$ measurements and $O(N \log^2(N))$ computations. It is worth noting that CS-BP does not necessarily ensure convergence, although the experimental results are satisfactory [3]. We will describe the stages of CS-BP process in more detail in the following.

2.5.1 Signal Prior Distribution Model

In this method, the signal to be recovered is K-sparse N-dimensional signal $\mathbf{x}$ which takes two-state mixture Gaussian model [26] as the prior distribution. Consider a random vector $\mathbf{X} = [X(1), \ldots, X(N)]$ in $\mathbb{R}^N$, and $x = [x(1), \ldots, x(N)]$ as an outcome of it. Next, the mixture Gaussian prior distribution will be assumed for each of X_i, as well as a state variable $Q(i)$. According to the signal model which consists of K large coefficient and $N - K$ small ones, large and small coefficient are classifie by assigning two different values 1 and 0 to the state variable, respectively. Therefore, the state random vector $\mathbf{Q} = [Q(1), \ldots, Q(N)]$, where $Q(i)$'s are iid with Bernoulli distribution ($Pr(Q(i) = 1) = S$ and $Pr(Q(i) = 1) = S$, where $S = K/N$ is the *sparsity rate*). In addition, smaller entries are mostly located around zero, therefore smaller variance will be assigned for their Gaussian prior, whereas, larger for coefficient larger variance is a proper choice. Next, as shown in Fig. 2.3, the prior distribution on all of the signal coefficient will be define similarly as follows:

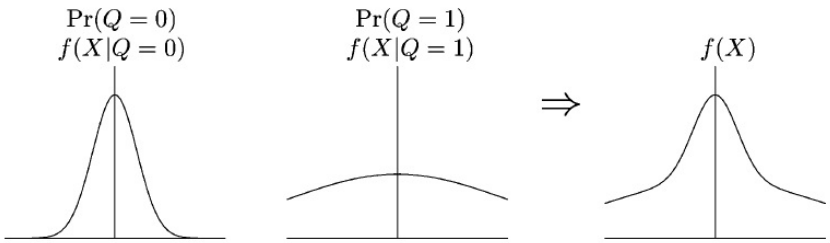


Figure 2.3: Mixture Gaussian model for signal coefficient [3].

$$f(X(i)) = \frac{K}{N} f(X(i)|Q(i)=1) + \frac{N-K}{N} f(X(i)|Q(i)=0)$$
$$= \frac{K}{N} \mathcal{N}(0,\sigma_1^2) + \left(1 - \frac{K}{N}\right) \mathcal{N}(0,\sigma_0^2) \quad (2.2)$$

2.5.2 Low Density Parity Check Codes

As mentioned before, in order to increase the speed of both the encoding and decoding, a sparse *LDPC-like* matrix called CS-LDPC is assigned. LDPC codes are specifie by matrices that contain mostly $0's$ and only a few $1's$. Low Density Parity Check codes are linear error correcting codes, which perform by transmitting messages over noisy channels [16]. The main advantage of these codes is that they perform very closely to the capacity of a lot of different channels in a higher speed. LDPC codes can be represented by matrices or graphical models.

- **Matrix representation**

  LDPC matrix is sparse, i.e., most of its coefficient are zero. To defin this type of matrix, firs the matrix should be a parity check matrix, ***i.e.***, it must consists of only 1 and 0 entries. Second, the matrix should be low density, ***i.e.***, the number of $1's$ must be a lot fewer than the matrix dimension. In more detail, for a $(m \times n)$ matrix, we defin the number of $1's$ in each row and column by w_r and w_c, respectively. If $w_r \ll n$ and $w_c \ll m$, then this matrix is called low-density. The following matrix is a parity-check (4×8) matrix, but as it is obvious it is not called a low-density matrix. In general, parity check matrix should be relatively large in order to be eligible to classifie as low-density.

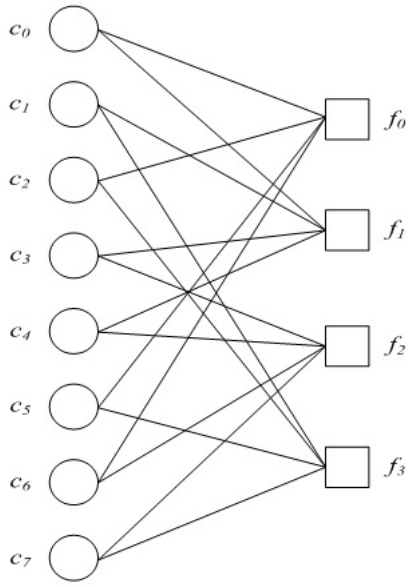


Figure 2.4: Tanner graph corresponding to the equation (2.3)

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{2.3}$$

- **Graphical representation**

  Tanner in [40] introduced an efficien graphical representation of LDPC codes, known as *Tanner graphs*. Tanner graphs are bipartite graphs which especially is used to represent analysis related to error correcting codes. More specificall , in coding theory, Tanner graphs are used to construct long error-correcting codes from smaller ones. The two types of nodes in this bipartite graph are called *variable* nodes (v-nodes) and *check* nodes (c-nodes). For instance, in Fig. 2.4 the tanner graph for the matrix described in 2.3 is shown. This graph consists of m check nodes (the number of parity bits) and n variable nodes (the number of bits in a codeword). Check node c_i is connected to variable node v_j if the ij'th entry of the corresponding matrix is 1.

An LDPC code is called regular if w_c and $w_r = w_c(n/m)$ are constant for every column and row, respectively. The example matrix in the equation (2.3) is regular with $w_c = 2$ and $w_r = 4$. It

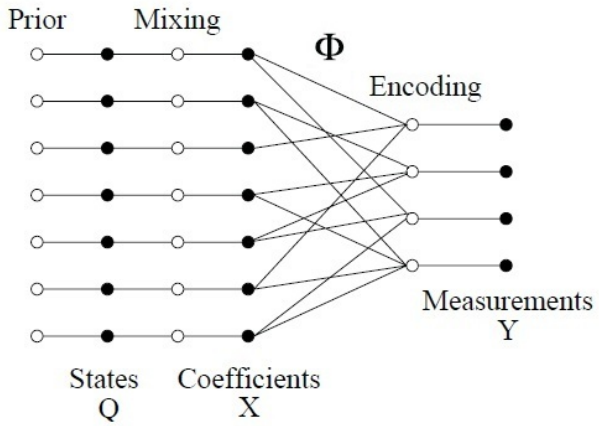


Figure 2.5: Factor graph of the CS-BP encoder [3].

is also possible to observe the regularity of the code from the graphical model. In a regular LDPC code, there is the same number of incoming edges to every v-node and also to every the c-node. If the numbers of 1s in each row or column are not constant in an LDPC code, the code is called an irregular LDPC code.

2.5.3 Sparse Encoding

The sparse encoding matrix used in this method is CS-LDPC matrix, which is slightly different from the LDPC matrix, as CS-LDPC consists of -1 entries as well as 0's and 1's. Baron *et al.* in [3] states that adding negative elements to the sensing matrix will result in performance improvement. As the entries of $\mathbf{\Phi}$ are only $-1, 0$ and 1, basic additions and subtractions are needed to produce the measurements. In addition, constant row, column weight constraints (L, R) on $\mathbf{\Phi}$ makes it a regular LDPC matrix. The number of measurements is calculated as sample rate times n, *i.e.*, $M = [n \times R/L]$. Significan assumption in this method is sparsity of the original signal, *i.e.*, the sparsifying basis $\mathbf{\Psi} = I$, which is not an applicable assumption. The factor graph describing the CS encoder is shown in Fig. 2.5.

2.5.4 CS-BP Decoding

Decoding procedure can be treated as a Bayesian inference problem using Belief Propagation. The goal is to recover $\mathbf{x}$ in $\mathbf{y} = \mathbf{\Phi x}$, knowing $\mathbf{\Phi}$ and $\mathbf{y}$, where $\mathbf{x}$ has a mixture Gaussian prior model.

Next, we need to calculate the "beliefs" or "messages" that are to pass the graphical model. The idea is to sample the pdf of each coefficien (variable node) and then send the samples as a P-dimensional message vector. Based on the Belief Propagation technique described in Chap. 1, the marginal distribution for a given variable node x_i is calculated (1.7). Using the marginal distribution in order to solve the ill-posed problem $\mathbf{y} = \mathbf{\Phi x}$, the idea in [3] is to choose the solutions that match the prior signal best. More specificall , signal is estimated using Minimum Mean Square Error (MMSE) and Miniumum a Posteriori (MAP) estimates:

$$\begin{aligned} \hat{\mathbf{x}}_{MMSE} &= arg\ \min_{x'} E||\mathbf{X} - x'||_2^2 \qquad \text{s.t.} \qquad \mathbf{y} = \mathbf{\Phi} x' \\ \hat{\mathbf{x}}_{MAP} &= arg\ \max_{x'} f(\mathbf{X} = x') \qquad \text{s.t.} \qquad \mathbf{y} = \mathbf{\Phi} x' \end{aligned} \tag{2.4}$$

where the expectation is taken over the prior distribution of $\mathbf{x}$.

It is worth noting that loops in the graph may violate the convergence to precise distributions or even cause BP algorithm to diverge. Although using sparse $\mathbf{\Phi}$ can reduce the number of loops and help in developing less complex message passing methods, some stabilizing techniques are further required. In [3], the message damped belief propagation (MDBP) [31] is utilized, where old and new estimates of messages are weighted averaged to produce new messages. Using MDBP will ensure the convergence, which is not feasible without it. The summarized CS-BP algorithm is described in Table 2.2.

2.6 Summary

According to Bayesian Compressive Sensing viewpoint, having a prior knowledge of sparsity of the signal in some bases $\mathbf{\Psi}$, one can reconstruct the signal from compressed measurements, with the objective to provide posterior distribution for the recovered signal as well. In this chapter, we have investigated wide variety of compressive sensing Bayesian inference problems, especially Bayesian compressive sensing using Belief Propagation which is essentially in connection with the area of this thesis.

Table 2.2: CS-BP Decoding algorithm [3].

- Step 1: Initialization.
  1. Initialize the iteration counter $i = 1$.
  2. Initialize messages from variable to check nodes with the signal prior.
- Step 2: Convolution.
  1. For each measurement, compute the convolution in (1.6) considering all neighboring variable nodes. Convolve further with a noise prior in the presence of noise.
  2. Apply damping methods such as MDBP [32].
- Step 3: Multiplication.
  1. For each coefficient compute the multiplication in (1.5) considering all neighboring check nodes.
  2. Apply damping methods as needed.
  3. If the iteration counter has yet to reach its maximal value, go to step 2.
- Step 4: Output.
  1. For each coefficient compute MMSE or MAP estimates based on the marginal distribution $f(v)$.

Chapter 3

Proposed SI-Assisted CS-Based Multiview Imaging

3.1 Introduction

In the following chapters, we are going to analyze the connection between Bayesian Compressive Sensing approach and Multiview Image coding scenario. Multiview imaging technology is an emerging fiel and is progressing by development of cost-reducing techniques in storage and acquisition. Compressive sensing, a framework which allows reliable sparse signal recovery using few samples can accomplish this goal, and may bring up highly efficien methods in multiview imaging field We have specificall investigated the application of the Bayesian compressive sensing via belief propagation (CS-BP) theory to Multiview image framework. The reason is that using CS-BP algorithm provides us with a proper environment to add prior information, by facilitating the use of side information from neighboring views. As mentioned before, according to this Bayesian inference technique, a prior distribution is define for the signal coefficients which imposes sparsity on the signal. As discussed earlier, the original CS-BP algorithm [3] assumes that all unknown variables have the same prior distribution, *i.e.*, zero mean Mixture Gaussian. This assumption is not true in many cases, especially images as the transform coefficient in images have different means and variances; therefore, some specialized adaptations should be taken into account. In this chapter, we utilize the information obtained of view interpolation on multiview images, in providing better prior distribution for signal coefficients The side information-assisted CS decoding can converge much faster and give better reconstruction performance. In addition, the CS-BP technique only

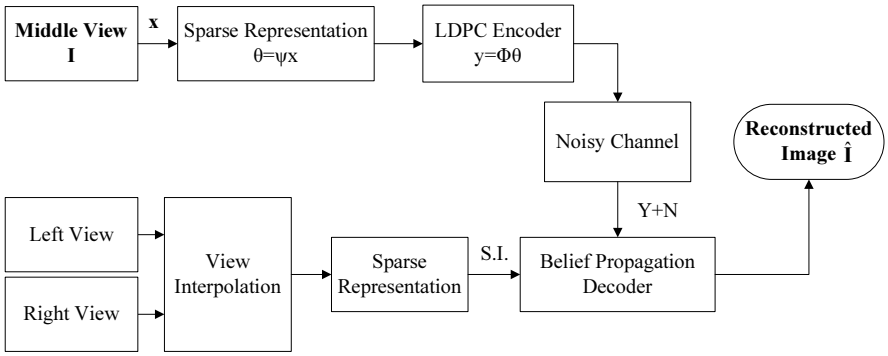


Figure 3.1: Block diagram of the proposed algorithm.

considers original sparse signals, which is not the case in most applications; instead, we generalize the algorithm for non-sparse signals by adding the sparsifying bases at the encoder side. We have therefore, introduced a block-based scheme which is able to take in any general non-sparse signal. The presented experimental results will discuss the effect of considering these assumptions and will depict how much improvement CS-MIC has compared to the original CS-BP method.

3.2 Side Information Incorporation in Multiview Image Recovery

In multiview imaging scheme, several views from a scene are captured by the surrounding cameras. The storage, acquisition and transmission cost of high quality images is a critical issue when dealing with applications such as surveillance. As the number of views increases, these costs increase considerably as well. There are some ways mentioned in the literature in order to reduce the cost, "virtual image" is an example. The virtual images can be obtained by utilizing the real views taken from other directions as well as disparity information. In this thesis the following scenario is considered: if some views are coded using the existing high-quality methods, and others are coded via compressive sensing encoder, how to achieve the best CS performance by taking advantage of the side information? In this regard, the assumption of the proposed setup is that CS camera would have lower cost and complexity compared to the conventional cameras. For simplicity we focus on the case with 3 views, where the left and the right views are coded using existing methods such as H.264 multiview video coding, whereas the middle view $\mathbf{I}$ is CS encoded. The general block diagram of this system is shown in Fig. 3.1.

3.2.1 Block-Based CS Encoder

At the CS-BP encoder, we use the same sparse CS-LDPC matrix as in [3]. However, since natural images are not sparse, we use the 2D discrete cosine transform (DCT) to generate a sparse representation of the image.

DCT is the approximation of the KLT ($Karhunen - Loéve\ Transform$) of $AR(1)$ signal when its correlation coefficien (ρ) is close to 1. KLT is used to express a signal as a linear combination of orthogonal functions; DCT represents a sequence of finitel many signal coefficient in terms of a sum of cosine functions at different frequencies, and it can be considered as a sparsifying basis for images. 1D DCT is calculated as bellow:

$$\theta(k) = \sum_{n=0}^{N-1} x_n \cos[\frac{\pi}{N}(n + \frac{1}{2}k)] \qquad k = 0, \ldots, N-1 \tag{3.1}$$

The equation above can be treated as a simple matrix multiplication as well $\theta = \mathbf{c}\mathbf{x}^T$, where $\mathbf{c}$ and θ are the 1D DCT and the transformed data, respectively. For large images, block-based DCT can be used to reduce the computational time and complexity. Typical block sizes used in the CS literature include 32×32 and 64×64. More information regarding DCT map is presented in Chapter 4.

Let $\mathbf{X}$ be a $N \times N$ block of the input image, and $\mathbf{C}$ represents the $N \times N$ 1D DCT. The 2D DCT on the $\mathbf{X}$ is the result of a 1D Row DCT followed by a 1D Column DCT. The transformed block is then given by,

$$\mathbf{\Theta} = \mathbf{C}(\mathbf{C}\mathbf{X}^T)^T = \mathbf{C}\mathbf{X}\mathbf{C}^T. \tag{3.2}$$

In addition, in order to apply the CS theory, we also need to convert 2D image data into 1D. This can be achieved by concatenating all columns of a 2D block into one vector. In this case, the whole CS-MIC scheme can then be established in a block-based format. Let $\mathbf{x}$ and θ be the 1D stacked vector of $\mathbf{X}$ and $\mathbf{\Theta}$, respectively; the equation above can be written as follows:

$$\theta = (\mathbf{C} \otimes \mathbf{C})\mathbf{x} \triangleq \mathbf{\Psi}\mathbf{x}, \tag{3.3}$$

where, $\mathbf{C} \otimes \mathbf{C}$ is the Kronecker product of $\mathbf{C}$.

After obtaining the sparse representation of each block of the input data, we apply it to the LDPC encoding system. The corresponding CS measurements are then,

$$\mathbf{y} = \mathbf{\Phi}\theta = (\mathbf{\Phi}\mathbf{\Psi})\mathbf{x}. \tag{3.4}$$

3.2.2 Side Information Assistance in BP Decoder

One useful property of the CS-BP algorithm is that it can easily utilize prior information to help the decoding. As discussed in Section 2.5, in CS-BP, a simple two-state mixture Gaussian model is used as the prior knowledge, which is based on the sparsity of the signal. In more detail, signal coefficient are assigned prior distributions of zero-mean mixture Gaussian, having two alternate variances (σ_1 or σ_0). As mentioned before, state variable $\mathbf{Q}$ is Bernoulli with probability S and $1 - S$, for large and small coefficients respectively. Therefore, the CS-BP prior model for each coefficien is calculated as follows:

$$f(x_i) = S\mathcal{N}(0, \sigma_1^2) + (1 - S)\mathcal{N}(0, \sigma_0^2). \tag{3.5}$$

As discussed earlier, this simple prior model is not suitable for many applications, such as image/video coding, where different coefficient usually have different pdfs. Some coefficient might have non-zero mean values, as well. In particular, if some side information is available, such as the interpolated view as in [46], we properly make use of the side information in definin prior distributions. The proposed prior pdf is such that it is centered around that side information, instead of 0. Hence, we defin the following prior distribution for signal coefficients

$$f(x_i) = \mathcal{N}(\hat{\mathbf{x}}(i), \sigma_i^2) \tag{3.6}$$

where $\hat{\mathbf{x}}$ is the side information, σ_i^2 is the variance of each unknown variable, which can be obtained from, *e.g.*, some training data. Specificall , we utilize the statistics of side information in definin the variance, as shown in Table 3.1.

Table 3.1: Obtaining variance for prior distribution

- Adjust threshold value TH for SI coefficient
- Find small SI coefficient (SI(i) $\leq$ "TH")
- Find standard deviation of them $\Longrightarrow \sigma_0$
- Skip DC
- Do similarly for other large components $\Longrightarrow \sigma_1$

In the following theorem we will determine the number of required measurements of the proposed algorithm and compare it with that of the original CS-BP. This theorem is analyzed based on the directions of the firs theorem in [3], and the proof appears in the Appendix A.

Theorem 1 Consider x as a Gaussian signal ($\mathcal{N}(\hat{\mathbf{x}}, \sigma^2)$), with sparsity rate $S = K/N$, and $\mathbf{\Phi}$ as a CS-LDPC matrix with $L = \frac{2\eta\sigma_1^2 \ln(SN^{1+\gamma})}{\sigma^2}$, where $\eta, \gamma > 0$. In order to decode x such that $\|x - \tilde{x}\|_\infty < \mu\sigma_1$ with probability $1 - 2N^{-\gamma}$, we need,

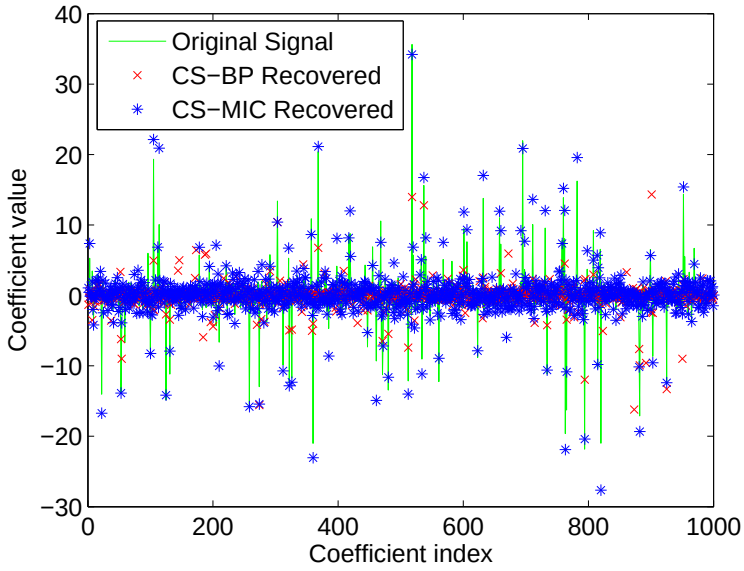
$$M = O\left(\frac{(1+\eta^{-1})(1+\gamma)}{\mu^2}(N+K)(\frac{\sigma}{\sigma_1})^2 \log N\right) \tag{3.7}$$

number of measurements.

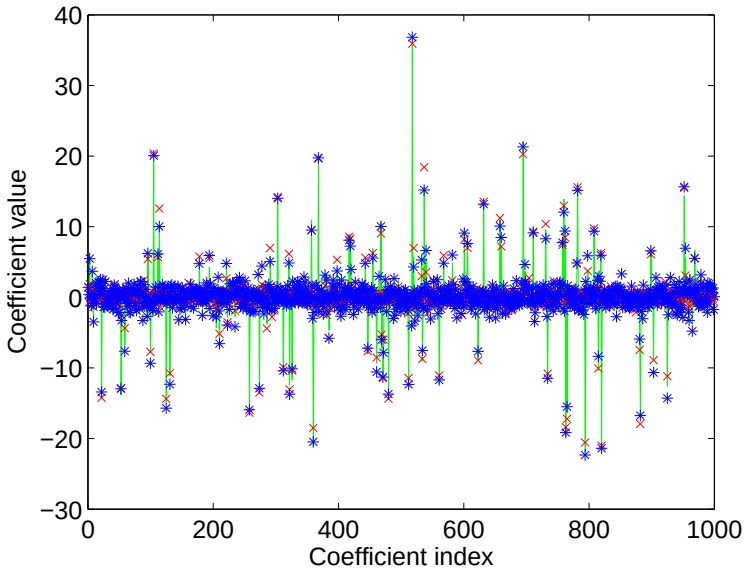
3.3 Experimental Results

For a fair comparison, we generate the 1D "sparse" signal with sparsity rate $K = 0.1$ as [3], and compare the result of the CS-MIC and CS-BP recovered signals with original signal $\mathbf{x}$ in Fig. 3.2. The model of the signal is the mixture of two randomly produced signal vectors, where K of the components have variance 10 and others have variance 1. $P = 243$ samples of pdf are set as messages in the BP decoder; this choice of P can provide fast FFT computation [3]. In this experiment, we use $L = 10$ and vary the sampling rate M/N from 0.2 to 0.7 for $N = 1000$ number of coefficients The side information in CS-MIC is generated by addition of a Gaussian noise. In order to check how similar are the generated SI and the signal $\mathbf{x}$, we calculate the relative error between them as the difference between the signal power and the noise power ($P_{\mathbf{x},dB} - P_{\mathbf{x}-SI,dB}$), which is $-0.48\,dB$ in this experiment. As the results show, adding the side information in the proposed method has great effect on the accuracy of the reconstructed signal, especially for larger coefficients

The Mean Squared Error as a function of the number of measurements for 3 different row weights L is presented in Fig. 3.3. The test signal is generated as before, with the same number of pdf samples as messages. As shown in the figure for both the CS-BP and CS-MIC, smaller row weights may cause loss of some of the large signal components, thus resulting in higher MSE. As we increase L, for both CS-BP and CS-MIC, fewer measurements are required to reconstruct the signal under the same recovery error. The MSE results for CS-MIC are obtained under the generation of SI with relative error $-0.52\,dB$. Results demonstrate up to 60% improvement over CS-BP, which is more sensed for lower sample rates.



(a) MSE(CS-BP):0.865, MSE(CS-MIC):0.391



(b) MSE(CS-BP):0.231, MSE(CS-MIC1):0.206

Figure 3.2: Comparison of the CS-BP method and the proposed one, at two different sampling rates: (a) 0.2 and (b) 0.7. $N = 1000$, $S = 0.1$, $\sigma_1 = 10$, $\sigma_0 = 1$, and $\sigma_z = 1$.

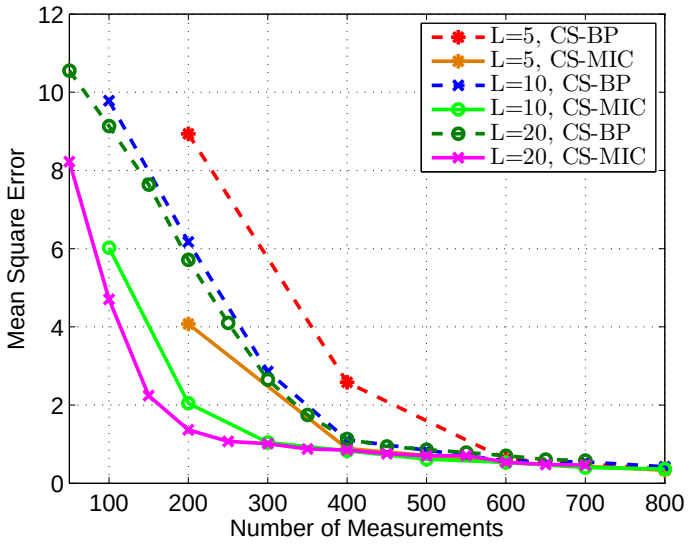


Figure 3.3: MSE as a function of the number of measurements M, for different row weights L. ($N = 1000$, $S = 0.1$, $\sigma_1 = 10$, $\sigma_0 = 1$ and noiseless measurements.)

In another experiment we apply the CS-BP, which is "slightly" modifie for non-sparse signals, and the proposed CS-MIC to the image sequences Rena and Akko. To compare the performance of the proposed method with the original CS-BP we use *PSNR* (Peak Signal to Noise Ratio).

PSNR is a term that basically compares the image qualities, and is usually used in image compression techniques. In more detail, PSNR determines the ratio between the maximum possible power of the data and the power of the compression noise and is define in logarithmic scale as bellow:

$$PSNR = 10\log_{10}\left(\frac{\text{Max}_s^2}{\text{MSE(e)}}\right) \tag{3.8}$$

where, e is the error between the original image and the noisy one, and Max_s is the maximum possible pixel value of the obtained image, which is equal to 255 in the case of 8-bit pixel representations per sample. Therefore, in this case which is the case of our experiments as well, PSNR is simplifie to $10\log_{10}\left(255^2/\text{MSE(e)}\right)$.

The PSNR comparison of the proposed method and the CS-BP for $L = 20$ is described in Fig. 3.4.

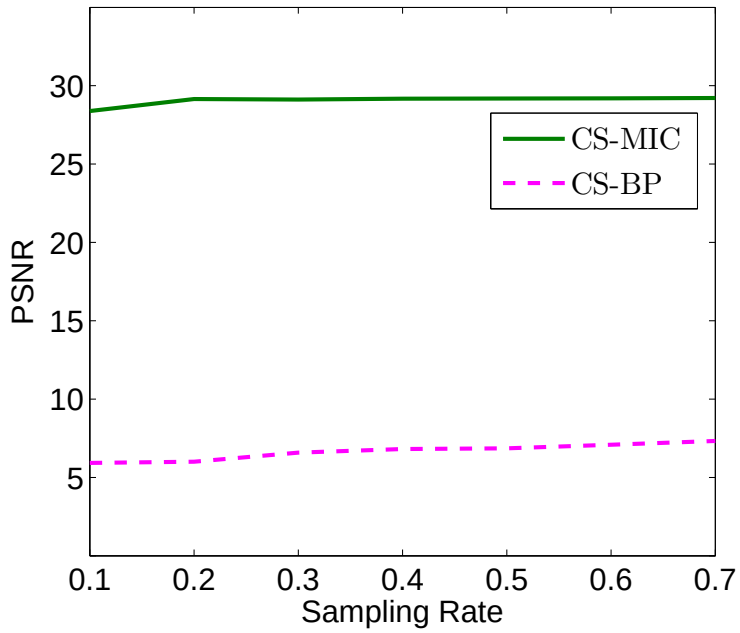
We test the impact of the SI quality on the performance of the algorithm firstl with the signal model produced as the mixture of two Gaussian distributions as before. In a similar experiment we

generate a different signal model, which can resemble the data in the next chapters of this thesis. This model stimulates the DCT transformed signal by generating the signal as the mixture of many Gaussian signals with different variances. Specificall , large coefficient are set to have decreasing variances and for simplicity, small coefficient have a f xed variance $\sigma_0^2 = 1$, for simplicity. This signal model is similar to the DCT transformed signal, which is the sparsifying basis used for non-sparse signals in this thesis. The impact of SI quality is depicted in Fig. 3.5, representing an increasing sequence of 4 SI SNR values by decreasing the variances.

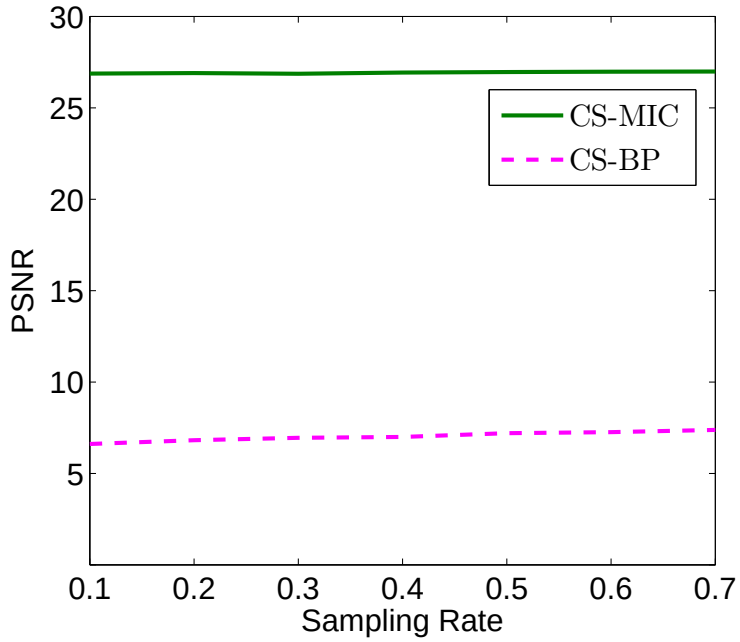
As mentioned earlier, using the prior information results in a faster convergence, and slower run time. For instance, for 32×32 blocks of the 256×256 Rena sequence, the average number of sufficien iterations and the CPU time usage for the original CS-BP and the proposed one, are $[5, 12.79$ sec] and $[2, 12.34$ sec], respectively.

3.4 Summary

In this chapter, we introduced an algorithm on generalization of the compressive sensing belief propagation method for multiview image coding scheme. In multiview image coding, compressive sensing techniques can efficientl improve the time and storage requirements. In addition, belief propagation decoder as well as sparse LDPC-based CS encoder will greatly increase the speed of the analysis. However, the original CS-BP method considers similar prior distributions for all the signal coefficients which is not proper for multiview applications. The proposed method in this chapter improves the performance of CS-BP for multiview case, by enabling CS-BP to take side information. The generalized CS-BP firstl make it feasible for the decoder to fully utilize side information and the method can efficientl handle variables with different distributions, as a result.

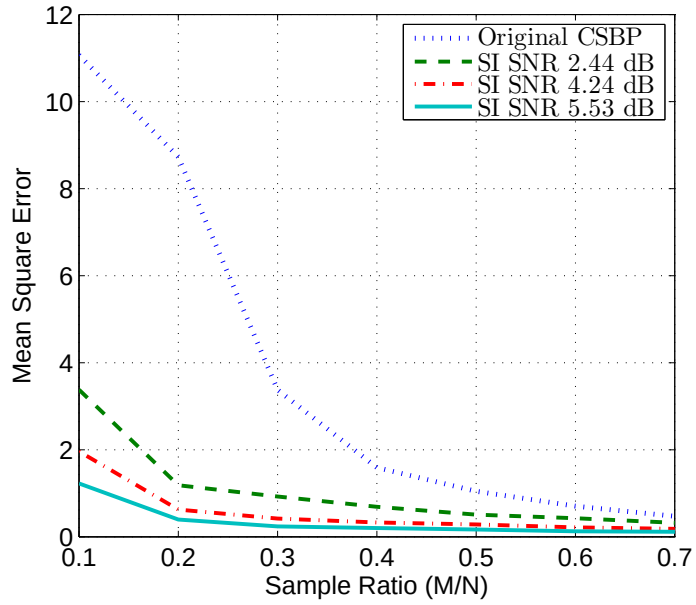


(a)

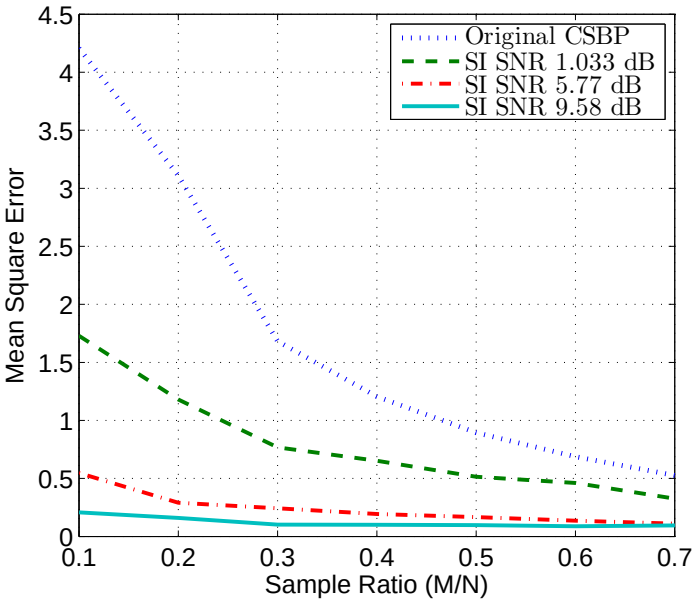


(b)

Figure 3.4: PSNR comparison of CS-MIC and CS-BP for (a) Rena (b) Akko



(a)



(b)

Figure 3.5: MSE as a function of the number of measurements M on the signal produced as: mixture of (a) two Gaussians and (b) many Gaussians. ($N = 1000$, $S = 0.1$, $\sigma_1 = 10$, $\sigma_0 = 1$ and noiseless measurements.)

Chapter 4

DC-Based Rectified CS-MIC

4.1 Introduction

In the CS-MIC, converging performance of loopy belief propagation decoder, is a significan issue, which can be maintained using damping methods or by adjusting the "step-size" of the pdf samples. In image coding, after the transform, different coefficient have dramatically different amplitudes and variances; however, having fi ed step-size and number of samples, imposed an over-bound region for the smaller coefficients This causes the fi ed number of samples to be taken over the large sampling region, which will result in an inaccurate recovery of these smaller coefficients Besides, if smaller sampling bound is used for all the coefficients large amount of data about the larger coefficient will be lost. Therefore, a proper method should be used which can gain perfect recovery for both the smaller and larger coefficients In this chapter, we have proposed an algorithm which will overcome this difficult by moderating the effect of the largest coefficien on the recovery of the smaller coefficients thus result in a greatly improved the performance. In more detail, generalizing the CS-BP method by implementing the proposed DC disjoint technique can greatly outperform the original CS-BP. Numerical results demonstrate that the proposed generalized algorithms on transformed images can achieve up to 29 dB of improvement compared to the original CS-BP method which is quite promising.

4.2 Message Passing in BP Decoder

As discussed in Section 2.5, messages in message passing scheme, are actually samples of pdfs. Using samples of pdfs, makes the algorithm fl xible to different prior distributions. For instance,

the algorithm can easily support the mixture Gaussian prior, and the message passing process will follow efficientl . However, there are some issues in using this method, such as high memory storage requirements and quantization error. To produce samples that result in more precise recovery, pdfs should be sampled with a relatively small step-size. In Section 2.5, the step-size is chosen to be less than σ_0, the stadard deviation of the narrow mixture Gaussian component.

As discussed earlier messages in this method are P dimensional vectors of pdf samples and as a result point-wise computations occur in message passing technique. In CS-BP, the fast Fourier transform (FFT) is used for computing the convolutions at constraint nodes. Since a high percentage of CPU time is spent on FFT computations, the CS-BP code is accelerated using an odd message length that factors well (an artifact of the implementation) [3]; therefore, the choice of $P = 243$ samples is used in the experiments.

Maximal sample values of the Gaussian pdfs will determine the bound on which the data is sampled. Based on the choice of samples, the produced messages may be a result of the measurements that consist of several large coefficients For example, in the simulations presented in previous chapter, for $L = 20$ and $S = 0.1$, an average of $L * S = 2$ larger coefficient occupied a message, so the choice of $5 * \sigma_1$ for the bound may seem reasonable at first However, it is possible that some messages contain more than 5 larger coefficients Consequently, some messages have amplitudes that surpass $5 * \sigma_1$, so limiting the absolute magnitude of messages to $5 * \sigma_1$ causes the loss of data and results in lower performance. Therefore in [3], the range $(-10\sigma_1, +10\sigma_1)$ is used which compared to smaller bounds offers better performance. In the analysis based on the choices of $\sigma_1 = 10$ and $\sigma_0 = 1$, and the assumption of step-size equal to σ_0, the approximate $20 * \sigma_1 = 200$ number of samples is an acceptable choice. Although, as mentioned earlier for better FFT computation the choice of 243 is used.

According to the Table 2.2, in Chapter 1, messages are firs initialized by signal priors, then for each measurement coefficient $1, \ldots, M$, the convolution of beliefs is computed which is followed by the multiplication at the variable nodes for $1, \ldots, N$ signal coefficients in the reverse direction. In a simple example we will study the impact of produced messages in signal recovery. For the factor graph shown in Fig. 4.1, with the assumption of only one iteration, we can derive the following equations which directly determine the reconstructed signal using the statistical characterizations of the marginal distributions. In this example 3 variable nodes are initialized by the prior distributions, and after a 3-stage computation, the marginal distribution for the coefficient are derived.

Note that these 3 coefficient have the same mixture Gaussian prior distributions; therefore the factor $f_m(x_i, x_j)$, is define as $\frac{K}{N}\,\mathcal{N}(y_m, \sigma_1^2) + \left(1 - \frac{K}{N}\right)\,\mathcal{N}(y_m, \sigma_0^2)$, where $m = \{1, 2\}$ is the

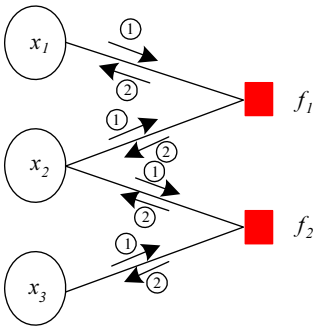


Figure 4.1: A factor graph which describes BP decoder, simply consisting of 2 measurements and 3 signal coefficient to be reconstructed.

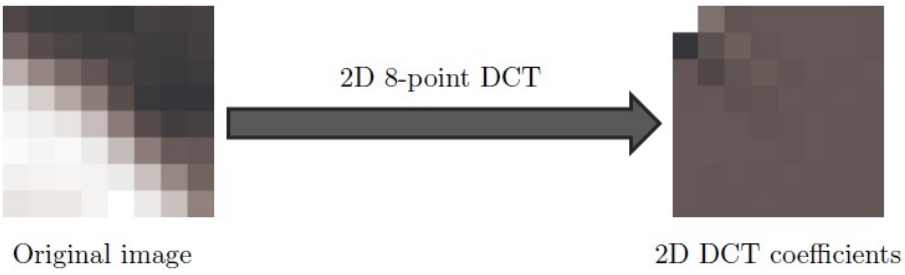


Figure 4.2: 8-point DCT transform on an 8x8 test image

index of measurements.

4.3 Reduction of the PDF sampling step-size

As mentioned before, an important issue of the original CS-BP algorithm is that all pdfs are sampled with the same step-size, as each unknown variable is assumed to have the same (mixture Gaussian) pdf. However, in image coding applications, after the DCT, different coefficient will have different amplitudes and variances. As a case in point, 2D DCT is the sparsifying basis which is widely used for images. Based on this transform, most of the signal energy is in the firs transformed coefficients especially DC. The effect of 2D DCT on an 8x8 test image is shown in Fig. 4.2.

As it is shown, there exists a large coefficien (DC) in the transformed signal due to DCT. This large coefficien as well as other relatively large coefficient require larger sampling region for the prior to be best fi in. Larger sampling bound and a f xed number of samples will cause the sampling step-size to increase which will result in more quantization error. As studied in Table

Table 4.1: BP decoding algorithm for the graph presented in Fig. 4.1.

- Stage 1: Initialization.
  $P_0 = \frac{K}{N}\,\mathcal{N}(0,\sigma_1^2) + \left(1-\frac{K}{N}\right)\,\mathcal{N}(0,\sigma_0^2)$

$$\begin{aligned}\mu_{x_1\to f_1(x_1)} &= P_0\\ \mu_{x_2\to f_1(x_2)} &= P_0\\ \mu_{x_2\to f_2(x_2)} &= P_0\\ \mu_{x_3\to f_3(x_3)} &= P_0\end{aligned}$$

- Step 2: Convolution.

$$\begin{aligned}\mu_{f_1\to x_1(x_1)} &= \sum_{\sim\{x_1\}} f_1(x_1,x_2)\mu_{x_2\to f_1(x_2)}\\ \mu_{f_1\to x_2(x_2)} &= \sum_{\sim\{x_2\}} f_1(x_1,x_2)\mu_{x_1\to f_1(x_1)}\\ \mu_{f_2\to x_2(x_2)} &= \sum_{\sim\{x_2\}} f_2(x_2,x_3)\mu_{x_3\to f_2(x_3)}\\ \mu_{f_2\to x_3(x_3)} &= \sum_{\sim\{x_3\}} f_2(x_2,x_3)\mu_{x_2\to f_2(x_2)}\end{aligned}$$

- Step 3: Marginalization.

$$\begin{aligned}f(x_1) &= \mu_{f_1\to x_1(x_1)}\\ f(x_2) &= \mu_{f_1\to x_2(x_2)}\mu_{f_2\to x_2(x_2)}\\ f(x_3) &= \mu_{f_2\to x_3(x_3)}\end{aligned}$$

4.1, the accuracy of the pdf samples have great influenc on the signal recovery. If the same pdf sampling step-size is used for all coefficients the step-size has to be very large, in order to cover the range of the large coefficient (specificall DC). This will introduce more quantization errors and will limit the resolution in the small coefficient recovery as a result. On the other hand, if small step-size is used instead, a lot portion of the larger coefficient data will be lost. Therefore, in order to get good reconstruction, the same step-size for all coefficient does not seem to be a reasonable assumption and a method should be introduced which can effectively choose the step-size among different coefficients In the following we will propose two methods on how to overcome the sampling step-size issue.

4.3.1 Direct DC Encoding

Generally, to decrease the quantization error, the sampling step-size should be reduced, and as mentioned earlier, step-size is determined by the largest signal coefficient thus we need to somehow remove the DC (and possibly other larger coefficients from the belief propagation iterations so that other coefficient will not be sacrifice because of DC. The easiest way to remove the DC from the belief propagation is to transmit the DC coefficien directly in the CS-MIC encoder. This situation can be viewed as an irregular CS-LDPC for the encoder, where one of the measurement nodes is only connected to the DC. Since the decoder will not estimate the DC, it can use smaller pdf sampling step-size to estimate other coefficients therefore leading to a better reconstruction quality. The block diagram of this scheme in the CS-MIC system can be simply viewed as Fig. 4.3.

As compared to original CS-BP, this approach of CS-MIC to DC component can greatly improve the performance of the generalized CS-BP, and make it suitable for different kinds of transformations. Basically the original CS-BP can be seen as a non-applicable approach for transformed data, and that's why its performance is very poor. Although our proposed approach on DC removing technique greatly outperforms original CS-BP, in most applications separating DC coefficien from the whole data is not desired. Therefore, a better approach which is able to reduce the sampling step-size without detached DC analysis should be introduced.

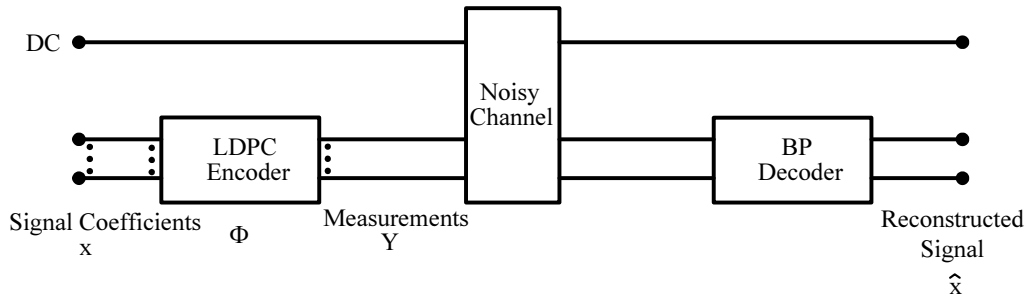


Figure 4.3: Block diagram of the direct DC encoding technique

4.3.2 Iterative DC Updating

In this section, we propose a method on dealing with DC coefficient which is more applicable and suitable if direct transmitting of DC is not desired. In our set-up we firs estimate the DC using the DC of the side information (interpolated data), then we remove the DC node and its contribution from the bipartite graph of the encoder. In more detail, as what is available at the decoder is only

the compressed sensed version of the data, with the assumption of knowledge about sensing matrix, we omit the DC effect from the measurement vectors, using the DC estimate. This is done by the assumption that DC of the target block equals to the DC of the SI. In addition, the factor graph edges that are connected to DC will be omitted thus BP iterations occur on updated $\mathbf{\Phi}$ matrix, as described in the Fig. 4.4.

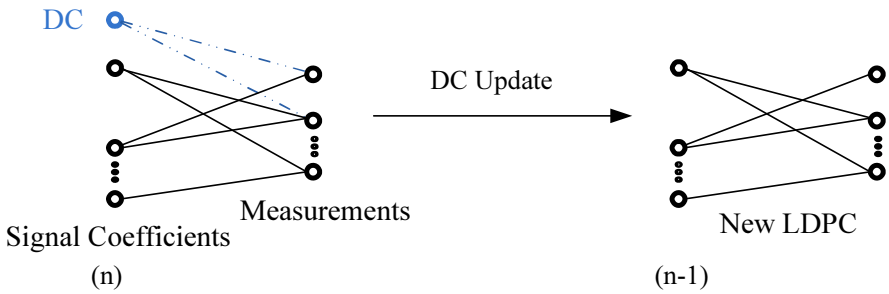


Figure 4.4: Block diagram of the DC updating technique

As shown in the figure in the proposed DC updating technique, the DC connected edges are removed from the BP iterations. In addition, the contribution of the estimated DC will be also subtracted from the check nodes that are connected to the DC. The measurements update is thus applied as follows:

$$y_i' = y_i - \mathbf{\Phi}_{i1}\hat{\theta}_{1,SI}, \tag{4.1}$$

where y_i is one of the measurements that are affected by the DC, $\mathbf{\Phi}_{i1}$ (1 or -1) is the $(i,1)$-th entry of the CS-LDPC matrix which connects the DC to the measurement y_i, $\hat{\theta}_{1,SI}$ is the DC of the side information obtained from view interpolation, and y_i' is the updated measurement after removing the DC contribution.

After taking off the DC and the corresponding edges from the factor graph, the belief propagation can be applied on the remaining nodes and connections of the graph. After sufficien iterations, the estimations of the other $N-1$ variable nodes can be calculated, which can then be used to refin the DC as follows:

$$\hat{\theta}_{1i} = \mathbf{\Phi}_{i1}(y_i - \sum_{j=2}^{N}(\mathbf{\Phi}_{ij}\hat{\theta}_j)), \quad i = 1,\ldots,T, \tag{4.2}$$

where $\hat{\theta}_{1i}$ is the i-th refinemen of the DC, $\hat{\theta}_j$ is the reconstructed j-th coefficien and $\mathbf{\Phi}_{ij}$ is the ij-th entry of the CS-LDPC matrix which connects the j-th signal coefficien to the i-th measurement. The average of the T refinement of the DC is then used as the fina refinement

Table 4.2: Generalized CS-BP decoding Algorithm.

- Initialization:
  1. Remove the edges affected by DC from the LDPC factor graph.
  2. Initialize the iteration counter $i = 1$.
  3. Initialize the DC value with the DC of the side information.
  4. Remove the DC contribution from the measurements. $y_i' = y_i - \mathbf{\Phi}_{i1}\hat{\theta}_{1,SI}$
- BP Decoder:
  1. Apply the updated measurements and new factor graph to the BP decoder.
- DC update:
  1. Estimate the DC from each connected check node. $\hat{\theta}_{1i} = \mathbf{\Phi}_{i1}(y_i - \sum_{j=2}^{N}(\mathbf{\Phi}_{ij}\hat{\theta}_j))$
  2. Average the estimates above to refin the DC. $\tilde{DC} = \frac{1}{T}\sum_{i=1}^{T}\hat{\theta}_{1i}$
  3. If the iteration counter has yet to reach its maximal value, go to step II. (i = i+1)

$$\tilde{DC} = \frac{1}{T}\sum_{i=1}^{T}\hat{\theta}_{1i} \tag{4.3}$$

The updated DC is closer to the DC of the original signal, compared to the DC of the side information, thus it is better to be used for DC extraction. The generalized CS-BP decoding algorithm is shown in Table 4.2.

After omitting DC, the pixel values of the $N-1$ remaining coefficient are smaller, therefore, the pdf sampling step-size can be reduced compared to the existence of all N samples; therefore, better reconstruction will be accomplished. The accuracy of the estimated DC during 3 consecutive iterations for sequence Rena is listed in Table 4.3. As it is clear in the results, among the 4 tested blocks, the DC of the blocks $1, 4$ and the DC of the blocks $2, 3$ are updated the best with 3 and 2 iterations, respectively. If an adaptive technique can be used to determine the number of iterations that make the prediction with the less error, then the method would be more efficien in the case of

costs and reconstruction quality. This adaptive technique will be discussed in Chapter 5.

Table 4.3: DC refreshment for 4 random blocks

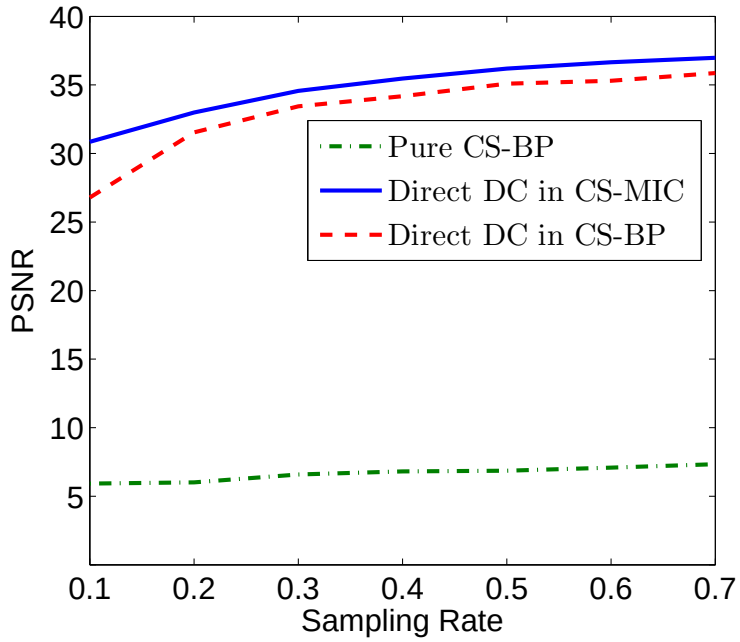
DC value	Block#1	Block#2	Block#3	Block#4
Side information	3.8894k	3.9505k	3.5050k	3.6419k
1^{st} iteration	3.7695k	3.8824k	3.4643k	3.5549k
2^{nd} iteration	3.7012k	**3.8663k**	**3.4359k**	3.5173k
3^{rd} iteration	**3.6630k**	3.8690k	3.4077k	**3.4966k**
Original DC	3.6556k	3.8650k	3.4272k	3.4965k

4.4 Performance Comparison

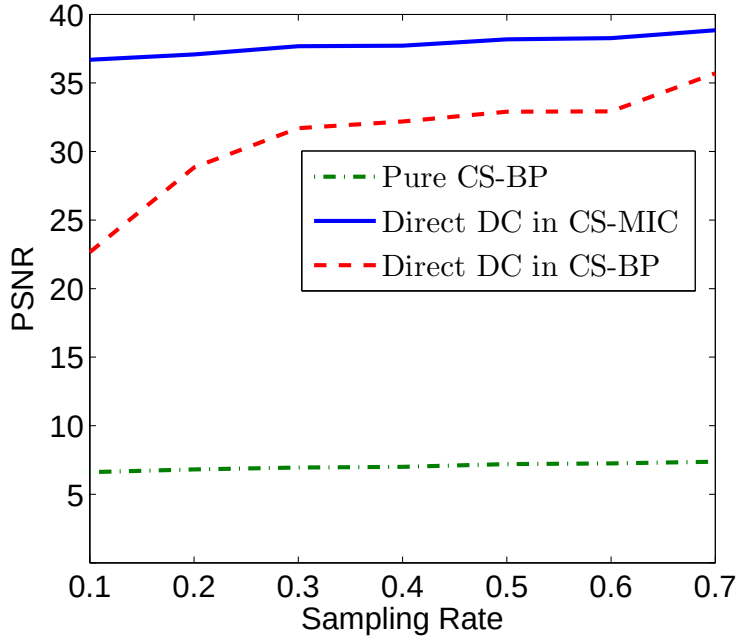
In this section, we test the result of the two proposed techniques on one frame of two different sequences Rena and Akkon, and obtain the PSNR comparisons. In the experiments, the size of each frame is 256×256 and the block size of block-based CS-MIC is 32×32. The results of the direct DC encoding technique is summarized in Fig. 4.5. It is worth noting that the firs method (direct DC encoding) can be seen as an upper bound for the second method (iterative DC updating). The comparison between the two methods is described in Fig. 4.6.

4.5 Summary

In this chapter, sampling step-size reduction, as a key point on improving the performance of CS-BP algorithms for transformed signals, especially images was introduced. Knowing that the sampling step-size value is enforced by the largest signal coefficient we have proposed two methods which deal with the DC of the signal separately. In both of the proposed techniques, DC is somehow removed from the BP iterations, and as a result, the second largest coefficien will determine the sampling step-size, thus signal recovery for smaller coefficient will be improved greatly.

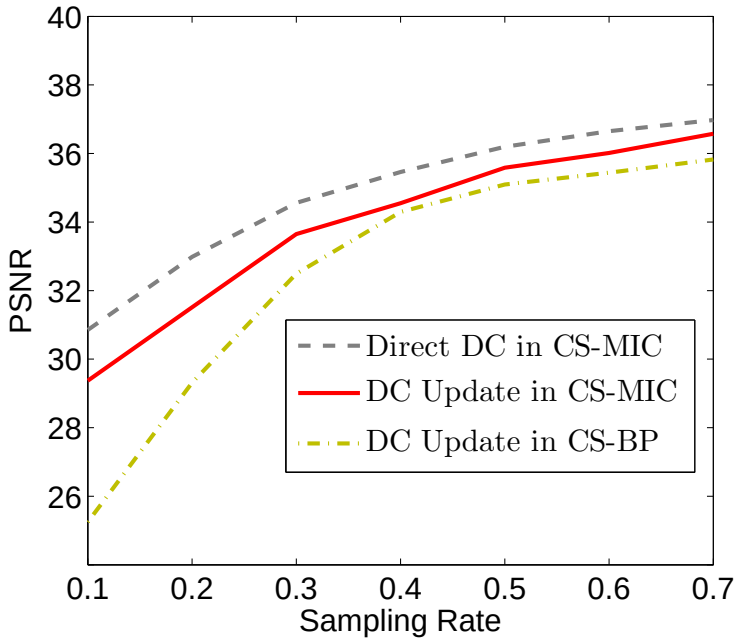


(a)

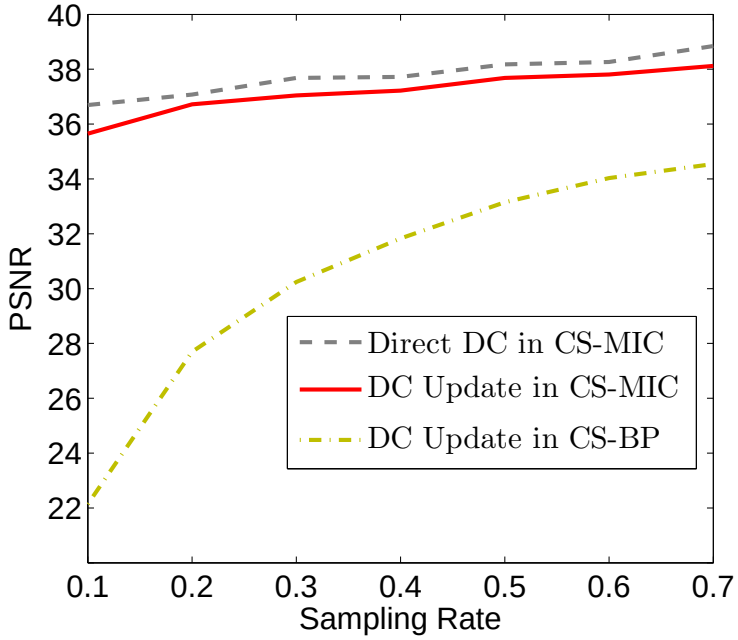


(b)

Figure 4.5: PSNR results of direct DC encoding technique for (a) Rena (b) Akko



(a)



(b)

Figure 4.6: PSNR comparison of the DC updating technique on (a) Rena (b) Akko

Chapter 5

CS-MIC Rectification Using Largest Coefficients

5.1 Introduction

As mentioned before, converging performance of loopy belief propagation is a significan issue which can be maintained using damping methods and by adjusting the step-size of the pdf samples. If the data contains large-valued coefficients the original CS-BP technique requires a very large sampling region, and thus large sampling step-size. Therefore, this scheme poorly handles the smaller coefficient and leads to inaccurate recovery. As CS-BP is not suitable for transformed data, especially images, in the previous chapter, we have introduced two possible methods to overcome this issue. The proposed methods dealt with DC component of the transformed data, and lead to improved accuracy of the reconstruction by reducing the sampling step-size in the BP decoder. One of the methods, the direct DC encoding acts as a benchmark for the other method, the iterative DC update, which is more applicable. In this chapter, we will study the previous methods not only on DC but also on other larger coefficient using an iterative manner. Simulation results show that depending on the data set, updating more coefficient rather than DC can further improve the reconstruction. In addition, in another experiment we propose an adaptive method which determines the number of coefficient that are to be updated, based on how much the SI coefficient are far off. Simulation results shows how efficien the latter method can be regarding the cost and storage requirements.

In the following we will describe some approaches on reducing the effect of DC as well as

other larger coefficients In the firs two schemes, the method deals with a f xed number of larger coefficients In these methods especially the second one, updating more large coefficient requires more computation costs and storages. Therefore, at the end of this chapter, we have proposed an adaptive technique which simply determines the number of required larger coefficient based on the trade-off between the costs and the reconstruction accuracy. At the end, simulation results state the advantages of using the adaptive technique.

5.2 Fixed Large Coefficients Refinement

5.2.1 Direct Large Coefficients Encoding

As what we have done in Chapter 4.3.1, we are now going to follow the same approach not only for the DC but for other larger coefficient as well. The block diagram of the introduced method in the CSMIC system is shown in Fig. 5.1. As shown in the figure larger coefficient of the data are sent directly to the encoder. To recover these coefficient one can also make use of the SI larger coefficients which are good estimates in most cases. The other coefficient will be applied to the LDPC-based CS encoder and BP decoder. The performance of this approach for different sample rates is tested on two sequences Rena and Akko and is shown in Fig. 5.2.

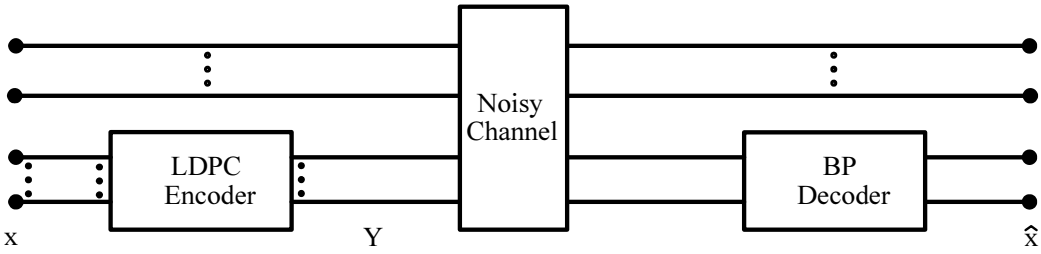
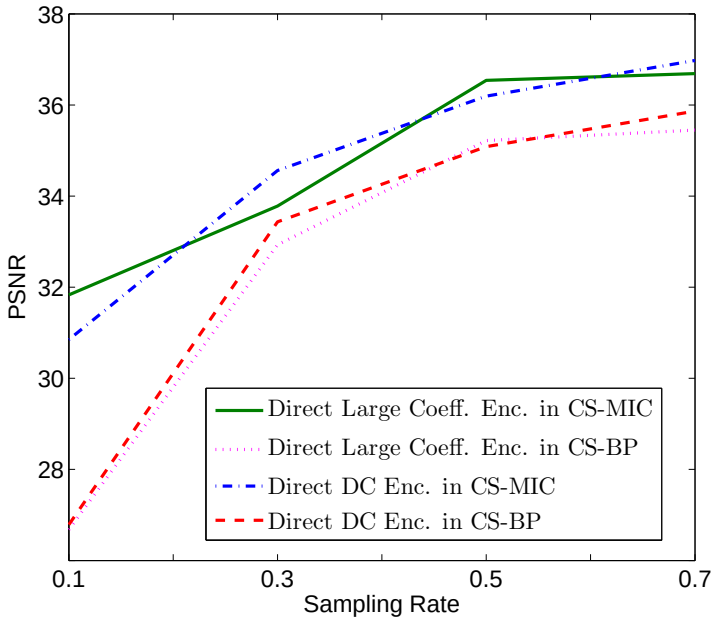


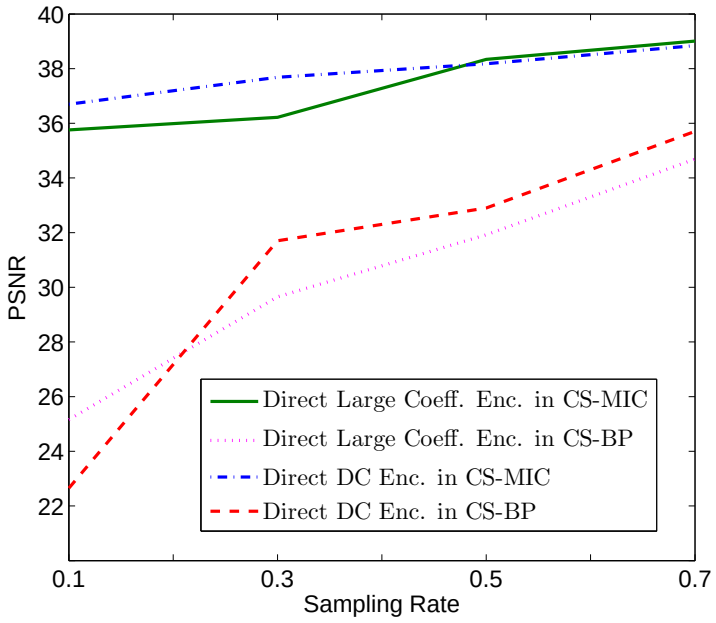
Figure 5.1: Block diagram for our algorithm

5.2.2 Iterative Large Coefficients Updating

The method in this section is a generalization of the method presented in Section 4.3.2. In the proposed algorithm, l larger coefficient in the data will sequentially get updated one after another. The idea is basically the same as iterative DC updating, however it is done on large coefficient rather than DC in a sequential manner. The block diagram of the updated encoding matrix is shown in Fig. 5.3.



(a)



(b)

Figure 5.2: PSNR results of direct large coefficient technique for (a) Rena (b) Akko

As shown in the figure such as the DC updating method, the edges that are connected to the l large coefficient will be ommited from the LDPC matrix and other coefficient will be decoded using the new LDPC matrix. Considering the amount of improvement based on the number of larger coefficient that get updated, as well as the introduced complexity, we fin out that up to 4 coefficient involvement is good enough. Note that according to the choice of L, the constant row weight of LDPC matrix, and the choice of the number of updated larger coefficients the number of involved measurements remain the same. In other words, there are at least 5 edges connected to each measurement node, then even after 4 coefficient update, there is still one edge connected to the measurement nodes. In addition, the contribution of the l larger coefficient should also get omitted from the affected measurement nodes. The summary of the large coefficient updating technique is summarized in Table 5.1.

Table 5.1: Large Coefficient Updating Algorithm.

- Initialization:
  1. Initialize the iteration counter $i = 1$.
  2. Remove the edges affected by the ith largest coeffi cient (L.C.) from the LDPC factor graph.
  3. Initialize the L.C. with the ith largest coefficien of the side information.
  4. Subtract the L.C. contribution from the measure- ments.
     $y_i' = y_i - \mathbf{\Phi}_{i1}\hat{\theta}_{1,SI}$
- BP Decoder:
  1. Apply the updated measurements and new factor graph to the BP decoder.
- L.C. update:
  1. Estimate the L.C. from each connected check node.
     $\hat{\theta}_{1i} = \mathbf{\Phi}_{i1}(y_i - \sum_{j=2}^{N}(\mathbf{\Phi}_{ij}\hat{\theta}_j))$
  2. Average the estimates above to refin the L.C.
     $\tilde{L.C.} = \frac{1}{T}\sum_{i=1}^{T}\hat{\theta}_{1i}$
  3. If the iteration counter has yet to reach its maximal value, go to step I.
  4. i = i+1. Go to step II.

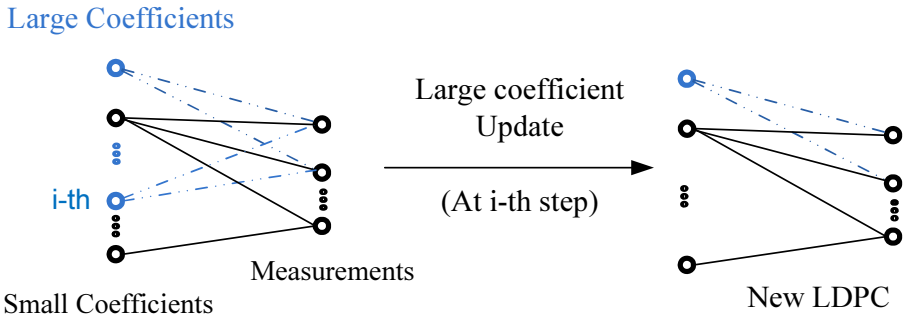


Figure 5.3: Block diagram for our algorithm

As the pixel values of the $N - l$ remaining coefficient are smaller, better reconstruction might be accomplished. However, the updated large coefficient introduce some error in addition. Besides, the computational complexity of the algorithm will increase as l increases. The result of the method is described in Fig. 5.4 and as it is shown direct large coefficien technique is considered as an upper bound.

We have tested the algorithm on the two sequences Rena and Akko, with the frame size of 256 and the block size of 32×32, and obtained the PSNR results. It is well noticed that as l increases the advantage of the method will decrease.

As mentioned earlier, more larger coefficient update, introduces more storage and computational complexities. In addition, determining the larger coefficient from the side information (in the second method) will introduce error in the recovery of large coefficient themselves. Also, lth largest coefficien gets closer to $(l-1)$th largest coefficien as l increases, therefore, removing the lth largest coefficien will not decrease the step-size profoundly, thus will not have a great effect on the performance. According to the existing trade-offs, an adaptive technique which decides on the value of l based on the proximity of the large coefficients can make a great adaptation.

5.3 Adaptive Large Coefficients Refinement

In this section, we introduce the adaptive technique, which will decide on the number of coefficient that should be refined This technique is applied to both of the previous methods and the results are discussed:

- **Adaptive Large Coefficients Removal**

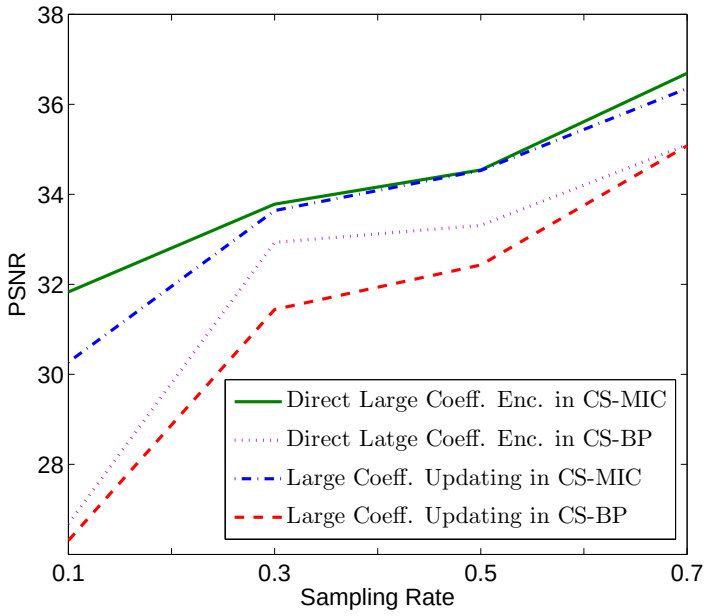
  In this technique, l largest coefficient will be coded directly, and the smaller LDPC will be

assigned and utilized for decoding the remaining $n - l$ coefficients It is basically acting similar to the scheme in section 5.2.1, however, l is not fi ed for all images here. In the new scheme, the best choice for l is assigned according to the image. We use the term *relative distance* to discuss how far the original signal coefficient are. In more detail we defin a threshold on the relative distance between the signal large coefficients For coefficient s_i and s_j, the relative distance which is simply calculated as $|(s_i - s_j)|/|s_i|$ will determine how many coefficient are sufficien to get encoded separately.

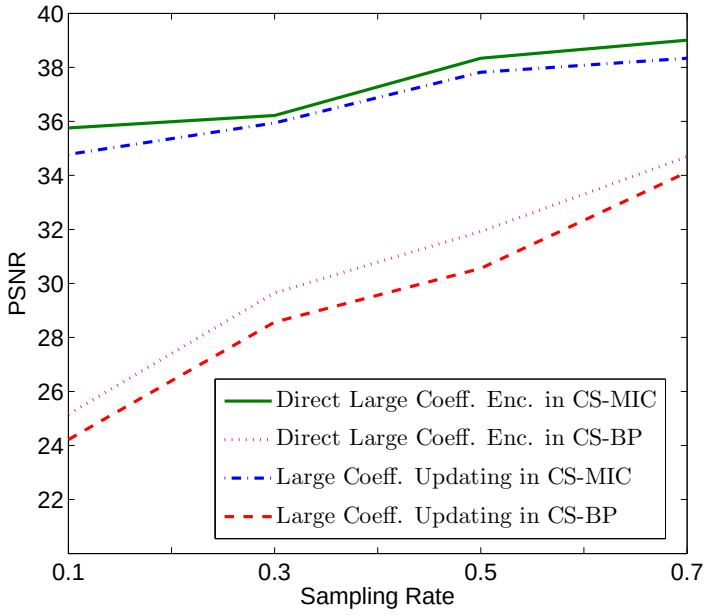
- **Adaptive Large Coefficients Update**

  In this scheme, similar to the idea presented in section 5.2.2, l largest coefficient will get updated sequentially, and the LDPC matrix get revised at the end of each iteration, which will then be employed in BP iterations for decoding the remaining $n - l$ coefficients The difference however, is in the choice of l which is adaptively selected in the new scheme. The relative error will determine how many coefficient are sufficien to get updated separately and how many are good to be processed using the CS-MIC path.

The adaptive large coefficient refinemen technique is summarized in Table 5.2. The performance of adding this technique to both of the previous methods is applied on Rena sequence and is discussed in Fig. 5.5.



(a)



(b)

Figure 5.4: PSNR results of large coefficient updating technique for (a) Rena (b) Akko, with $l = 3$

Table 5.2: Obtaining variance for prior distribution

- Adaptive DC Removal:
  Initialize the iteration counter $i = 1$, and the threshold value TH.
  1. Remove the ith largest coefficien to be processed individually.
  2. Apply the remaining coefficient to the CS-MIC system.
  3. Calculate the relative distance (RD) between ith and $i + 1$th larger coefficients
  4. If RD $\geq$ TH then $i = i + 1$ and proceed to step 1.
  5. END
- Adaptive DC Updating Initialize the iteration counter $i = 1$, and the threshold value TH.
  1. Remove the edges affected by the ith largest coeffi cient from the LDPC factor graph.
  2. Initialize the ith largest coefficien value with the ith of the side information.
  3. Subtract the ith largest coefficien contribution from the measurements.
  4. Calculate the relative distance (RD) between ith and $i + 1$th larger coefficients
  5. If RD $\geq$ TH then $i = i + 1$ and proceed to step 1.
  6. END

In the following section, we will discuss the simulation results for all the approaches that proposed in this thesis, and we compare them with the original CS-BP method.

5.4 Performance Evaluation

In this section we will compare the methods proposed in the previous sections. The proposed techniques are tested in a block-based framework and the results are depicted for a 256×256 Rena and Akko frames. Figure (b) represents the result of view interpolation on two neighboring views, where the interpolated result has shift at border pixels [46]. Due to the fi sampling bound in the original CS-BP method, this technique has a very poor performance and a very low PSNR as listed in Chapter 3, thus we have not presented it here. Methods shown in (c) and (d) have improved the performance of the original CS-BP to a great extent. Finally the results of e and especially f have

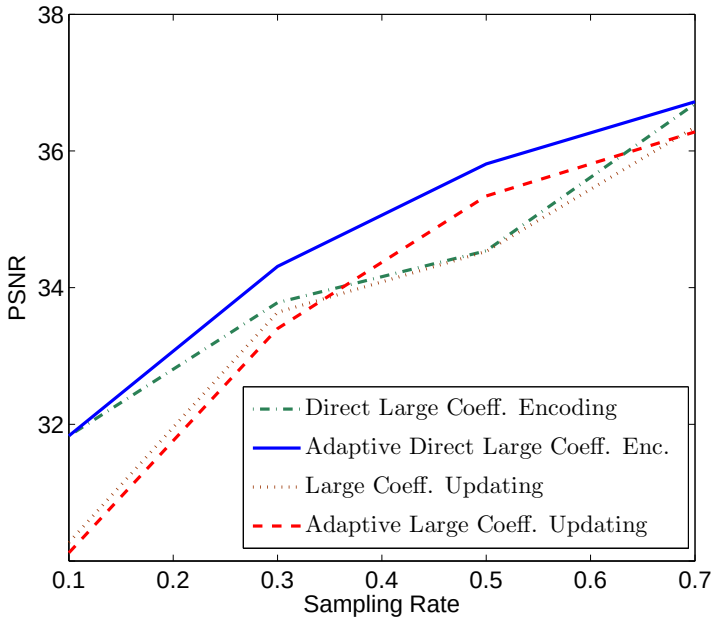


Figure 5.5: PSNR results of the proposed adaptive technique on Rena sequence

greatly outperformed the previous methods. The PSNR results for figure $b-f$ are listed in Table 5.3. As presented in previous chapters, proposed techniques can achieve up to 8 dB improvement in border pixels recovery regarding the side information, and up to 29 dB improvement compared to the original CS-BP method. The performance depends on the image, the sampling rate and the employed method. Figures (e) and (f) have greater PSNR regarding (b), which shows better reconstruction of border pixels.

Table 5.3: PSNR comparison of the proposed schemes

Figure	(b)	Original CS-BP	(c)	(d)	(e)	(f)
Rena PSNR	28.10 dB	6.43 dB	30.13 dB	32.49 dB	35.56 dB	36.35 dB
Akko PSNR	35.29 dB	7.34 dB	31.78 dB	30.24 dB	37.13 dB	38.02 dB

5.5 Summary

In this chapter, we have extended the idea of sampling step-size reduction from mere DC component to more large coefficients Two methods are proposed which deal with largest coefficient differently. In an iterative process, the effect of largest coefficient are cleared away from the BP iterations, thus the signal recovery for other coefficient will be preciser. In addition, because of the trade-off between the accuracy of smaller signal coefficient and computational costs, we proposed an adaptive technique which will determine how many larger coefficient are to be updated. This method can efficientl bypass the unnecessary computations, which is required in image processing especially in multiview technology.

(a) Original middle view

(b) Interpolated middle view

(c) SI-assisted CS-BP result

(d) DC updating CS-BP result

(e) DC updating CS-MIC result

(f) $\ell = 3$ Large coeff. updating CS-MIC result

Figure 5.6: Comparison of the proposed recovery techniques, for 256×256 Rena frame and 32×32 block size and sampling rate 0.3.

(a) Original middle view

(b) Interpolated middle view

(c) SI-assisted CS-BP result

(d) DC updating CS-BP result

(e) DC updating CS-MIC result

(f) $\ell = 3$ Large coeff. updating CS-MIC result

Figure 5.7: Comparison of the proposed recovery techniques, for 256×256 Akko frame and 32×32 block size and sampling rate 0.3.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

In this thesis, we have proposed a multiview image coding scheme which is well suited for transformed signals. The method of reconstruction used for this system is LDPC encoding and BP decoding (CSBP). In more detail, we firstl proposed a multiview image coding setup, which is perfectly adapted to utilizing side information in image reconstruction. The side information is the result of view interpolation technique on adjacent views of an image, which can greatly outperform the compressive sensing image recovery. The firs proposed method can perfectly match itself to multiview scenario, and it can be employed to general non-sparse signals as well.

Secondly, we have generalized the original CSBP algorithm which adjusts itself to transformed signals that dramatically have different coefficien values. In original CSBP the prior pdf which is used for sparse signal reconstruction, is sampled over a fi ed sampling region which is assigned based on largest coefficien values. This approach in assigning the sampling area will increase the sampling step-size and thus is not favorable in recovery of smaller coefficients In the firs step in our generalized method we have proposed techniques that peel off DC in the CSBP decoding, and improved the accuracy of signal recovery especially for smaller coefficients

We have further improved the signal recovery by removing more larger coefficien rather than DC. For a f xed number of coefficients we sequentially remove other large coefficient to further reduce the pdf sampling step-size and improve the reconstruction quality, which can be considered as scalable quantization.

Finally we introduced an adaptive method which will choose the number of larger coefficient

that are to be removed. According to the image, and the relations among signal coefficient values, the proper number of large coefficient removal will be obtained. This approach have great advantages in saving costs and storage requirements, as updating coefficient will increase both complexity and costs.

6.2 Future Work

6.2.1 Various Compressive Sensing techniques

Compressive sensing applications in multiview image scenario is relatively a new topic. Due to the high volume of the analysis and experiments needed for the current setup, we have just considered the belief propagation based compressive sensing technique. There is still a lot of work needed to test other compressive sensing methods as well, which will be employed as the future work.

6.2.2 Multiview Video Coding

The current program is explicitly designed for multiview image network. Studying the compressive sensing techniques on sequences of videos is also of a great interest and can introduce low complex applications for video compression. Designing compressive sensing multiview video coding setup will result in even a more efficien scheme, as analysis of high volume of frames of videos is costly computationally complex and time consuming. As a case in point, one may encode the reference frames using compressive sensing methods, while other frames are encoded based on conventional techniques.

6.2.3 Investigation of other prior pdfs

In this thesis, Gaussian pdf is used as the prior distribution; however, other prior pdfs may also be used. Thus, an interesting part of the future work is to assess the use of other prior pdfs such as Laplace, and determine their capability compared to Gaussian pdf.

Appendix A

Proof of Theorem 1

Our goal is to show that the proposed algorithm obviously achieves a better performance than the original CS-BP. We follow the same approach as [3] in the proof of the theorem, and describe the improved performance in terms of the number of measurements and sparsity of the measurement matrix $\mathbf{\Phi}$. We firs obtain some probabilistic bounds on $\|x\|_2^2$ and $\|x\|_\infty^2$, then, using the results obtained by Wang *et al.* in [44], the proof is completed.

Upper bound on $\|x\|_2^2$: Consider $\|x\|_2^2 = \sum_{i=1}^{N} x_i^2$, where X_i has the Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}(i), \sigma_i^2)$. Therefore, X_i^2 is distributed based on the non-central Chi-square distribution with parameter $\lambda = (\frac{\mu_i}{\sigma_i})^2$ as follow:

$$X_i^2 \sim \chi^2 \sigma_i^2 \tag{A.1}$$

Next step is to calculate the *moment generating function* (MGF) of X_i^2. For the non-central chi-squared random variable with 1 degree of freedom, the MGF is define as,

$$M_{\chi^2}(t) = \frac{exp\left(\frac{\lambda t}{1-2t}\right)}{\sqrt{1-2t}} \tag{A.2}$$

Since X_i are iid, MGF of $\|x\|_2^2$ is obtained as the multiplication of MGF of X_i's. For simplicity and to reduce the computations, we have considered the same mean and variance for all X_i, and MGF is thus calculated as follow:

$$M_{\|x\|_2^2}(t) = \left[M_{X_i^2(t)}\right]^N = \frac{exp\left(N(\frac{\mu}{\sigma})^2 \frac{t\sigma^2}{1-2t\sigma^2}\right)}{(1-2t\sigma^2)^{N/2}} \tag{A.3}$$

As $E[X^2] = \sigma^2 + \mu^2$, $Pr(\|x\|_2^2 > N\sigma^2)$ is greater than $Pr(\|x\|_2^2 < N\sigma^2)$. Then after applying the Chernoff bound, for $t < 0$ we have,

$$Pr(\|x\|_2^2 < N\sigma^2) < exp(-tN\sigma^2)\frac{exp\left(N\frac{\mu^2}{1-2t\sigma^2}\right)}{\sqrt{(1-2t\sigma^2)^N}} \tag{A.4}$$

Let $\alpha = -t\sigma^2 > 0$, we defin two functions $f_1(\alpha)$ and $f_2(\alpha)$ as follows:

$$Pr(\|x\|_2^2 < N\sigma^2) < \frac{exp(\alpha N^2 \frac{\mu^2\sigma^2}{1+2\alpha})}{\sqrt{(1+2\alpha)^N}} = \frac{f_1(\alpha)}{f_2(\alpha)} \tag{A.5}$$

Based on the Taylor series, $f_1(\alpha)$ and $f_2(\alpha)$ are estimated by $f_i(0) + f_i'(0)\alpha + O(\alpha^2)$, and the above result is obtained as bellow,

$$\frac{f_1(\alpha)}{f_2(\alpha)} = \left(1 + N^2\mu^2\sigma^2\alpha + O(\alpha^2)\right)\left(1 - N(1+2\alpha)^{-(1+N/2)}\alpha + O(\alpha^2)\right) \tag{A.6}$$

As it is obvious, the term $-N(1+2\alpha)^{-(1+N/2)}\alpha$ is negative and dominant over $O(\alpha^2)$ for small α. Therefore, there exists α which results in $f_1(\alpha)/f_2(\alpha) < 1$, and we can show [3],

$$Pr(\|x\|_2^2 < N\sigma^2) = O(N^{-\gamma}) \tag{A.7}$$

Lower bound on $\|x\|_2^2$: Similar to the approach in [3], to obtain the upper bound, we utilize the expected squared ℓ^2-normand obtain,

$$Pr(\|x\|_2^2 > N(2\sigma^2 + \mu^2)) \tag{A.8}$$

Based on the choice of σ in our algorithm which is obtained from some training data, and the value of μ, we conclude that $\sigma^2 + \mu^2 > 2\sigma^2 > (1+S)\sigma^2$, where $0 < S < 1$ is the sparsity rate. Therefore, $Pr(\|x\|_2^2 > N(2\sigma^2+\mu^2))$ is less than $Pr(\|x\|_2^2 > N(1+S)\sigma^2)$; this amount is a proper value for later analysis. Following the details in [3], we conclude that,

$$Pr(\|x\|_2^2 > N(1+S)\sigma^2) = O(N^{-\gamma}) \tag{A.9}$$

Bound on $\|x\|_\infty$: As known $\|x\|_\infty$ is define as $\max(|x_1|, \ldots, |x_n|)$. In order to provide clearer comparison between our method and the original CS-BP, to calculate the upper bound on $\|x\|_\infty$, we compare our proposed signal model ($\mathcal{N}(\hat{x}, \sigma^2)$) with a non-zero mean mixture Gaussian model ($S\mathcal{N}(\hat{x}, \sigma_1^2) + (1-S)\mathcal{N}(\hat{x}, \sigma_0^2)$), where σ_0 and σ_1 are define as before. Based on the signal models we can conclude that for the same x, $\|x\|_{\infty,\, Gauss.} < \|x\|_{\infty,\, Mix.\, Gauss.}$. Therefore, for a

clearer comparison we employ the $\|x\|_\infty$ upper bound for our method, which, according to analysis in [3], is $\sqrt{(2\ln(SN^{1+\gamma}))}\sigma_1$.

After obtaining the required statistical bounds, we then employ the theorem by [44], to calculate the required number of measurements to estimate the signal.

According to *theorem* 1 in [44], if $x \in \mathbb{R}^N$ satisfie $\frac{\|x\|_\infty}{\|x\|_2} \leq T$, and W be the matrix of N vectors that satisfie $w_1, \ldots, w_N \subset \mathbb{R}^N$. Consider the encoding matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ satisfying $E[\phi_{mn}] = 0, E[\phi_{mn}^2] = 1, E[\phi_{mn}^4] = s$, where $\frac{1}{s} = \frac{L}{N}$ is the ratio of the non-zero elements in $\mathbf{\Phi}$. In this case, the random projections $\frac{1}{M}\mathbf{\Phi}x$ and $\frac{1}{M}\mathbf{\Phi}w_i$ can reconstruct $x^T w_i$ with the probability $1 - N^{-\gamma}$ and with the error less than $\epsilon \|x\|_2 \|w_i\|_2$, if the number of measurements M satisfies

$$M = \begin{cases} O(\frac{1+\gamma}{\epsilon^2} sT^2 \log(N)) & \text{if } sT^2 \geq \Omega(1) \\ O(\frac{1+\gamma}{\epsilon^2}\log(N)) & \text{if } sT^2 \geq O(1) \end{cases} \tag{A.10}$$

According to our analysis, $\|x\|_2 \in (\sqrt{N}\sigma, \sqrt{N(1+S)}\sigma)$ and $\|x\|_\infty < \sqrt{2\ln(SN^{1+\gamma})}\sigma_1$ with probability $1 - N^{-\gamma}$ [3]; therefore, $T = \frac{\sigma_1}{\sigma}\sqrt{\frac{2\ln(SN^{1+\gamma})}{N}}$. Similar to the approach in [3], w_i's are chosen as the canonical vectors of the identity matrix I_N, providing $x^T w_i = x_i$. Moreover, based on the define L, $s = \frac{N}{L} = \frac{N\sigma^2}{2\eta\sigma_1^2\ln(SN^{1+\gamma})}$. Finally we set ϵ as,

$$\epsilon = \frac{\mu\sigma_1}{\sqrt{N(1+S)}\sigma} \tag{A.11}$$

Now based on the result of *theorem 1* in [44], $\tilde{x}_1, \ldots, \tilde{x}_N$ satisfy $|\tilde{x}_i - x_i| = \left|\tilde{x}_i - x^T w_i\right| \leq \epsilon \|x\|_2 \|w_i\|_2 < \mu\sigma_1$, with probability lower bounded by $1 - N^{-\gamma}$. Which will then lead to $\|x - \tilde{x}\|_\infty < \mu\sigma_1$ with probability lower bounded by $1 - 2N^{-\gamma}$ [3].

The proof is then finalize by obtaining the number of measurements based on (A.10),

$$M = O\left((1+\eta^{-1})\frac{1+\gamma}{\epsilon^2}\log(N)\right) = O\left(N(1+\eta^{-1})\frac{(1+\gamma)}{\mu^2}(1+S)(\frac{\sigma}{\sigma_1})^2\log N\right) \tag{A.12}$$

which confirm both the fewer number of L and the number of measurements.

Bibliography

[1] H. Wang A. Aldroubi and K. Zaringhalam. Sequential compressed sampling via huffman codes. *Preprint*, 2008.

[2] R. Baraniuk. Compressive sensing.

[3] R. Baron, S. Sarvoham, and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. 58(1):269–280, January 2010.

[4] C. M. Bishop. In Elisabet Engdahl, editor, *Pattern Recognition and Machine Learning*, Information Science and Statistics, pages 29–37. Springer-Verlag New York, LLC, 2006.

[5] E. Candes and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Prob.*, 23(3):969–985, June 2007.

[6] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006.

[7] E. Candes and M. Wakin. An introduction to compressive sampling. *IEEE Signal Proc. Magazine*, 25(2):21–30, 2008.

[8] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[9] Xiangyang Ji Changjun Fu and Qionghai Dai. Compressed multi-view imaging with joint reconstruction. *2010*.

[10] S. S. Chen, D. L. Donoho, and M .A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci Comput.*, 20(1):33–61, 1998.

[11] B. D. Rao D. P. Wipf. Sparse bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8), 2004.

[12] D. Donoho. Compressed sensing. 52(4):1289–1306, April 2006.

[13] P. J. S. G. Ferreira. Noniterative and fast iterative methods for interpolation and extrapolation. In *Signal Processing for Multimedia*, pages 35–54. J. S. Byrnes, Ed. Amsterdam, The Netherlands: IOS, 1999.

[14] S. Mallat G. Davis and M. Avellaneda. Adaptive greedy approximations. *Journal of Constructive Approximation*, 13:57–98, 1997.

[15] R. G. Gallager. Low-density parity-check codes. 8(1):21–28, January 1962.

[16] R.G. Gallager. In *Low-Density Parity-Check Codes*. M.I.T. Press, Cambridge, MA, 1963.

[17] G. Guo and C.-C. Wang. Multiuser detection of sparsely spread cdma. *IEEE J. Sel. Areas Commun.*, 26(3):421–431, 2008.

[18] W. Dai H. V. Pham and O. Milenkovic. Sublinear compressive sensing reconstruction via belief propagation decoding. *IEEE International Symposium on Information Theory (ISIT)*, 56, 2009.

[19] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.

[20] K. Kreutz-Delgado et al. Dictionary learning algorithms for sparse representation. *Neural Comput.*, 15(2):349–396, February 2003.

[21] J. Park M. Akcakaya and V. Tarokh. Low density frames for compressive sensing. *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 3642–3645, March 2010.

[22] E. W. Tramel J. E. Fowler M. Trocan, T. Maugey and B. Pesquet-Popescu. Multistage compressed-sensing reconstruction of multiview images. *2010*.

[23] S. Mallat. *A wavelet tour of signal processing*. Academic press, New York, 1999.

[24] S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[25] F. Marvasti. Random topics in nonuniform sampling. In *Nonuniform Sampling: Theory and Practice*, pages 169–234. F. Marvasti, Ed. Springer, formerly Kluwer Academic/Plenum Publishers, 2001.

[26] E. I. George; R. E. McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 1993.

[27] J. Mooij and H. Kappen. Sufficien conditions for convergence of the sumproduct algorithm. 53(12):4422–4437, 2007.

[28] D. Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal.*, 26(3):301–321, 2008.

[29] L. C. Potter P. Schniter and J. Ziniel. Fast bayesian matching pursuit: Model uncertainty and parameter estimation for sparse linear models. *IEEE Transactions on Signal Processing*, 2009.

[30] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, 1988.

[31] M. Pretti. A message-passing algorithm with damping. *Journal of Statistical Mechanics: Theory and Experiment*, 2005.

[32] M. Pretti. A message-passing algorithm with damping. *J. stat. Mech.*, November 2005.

[33] R. DeVore R. G. Baraniuk, M. Davenport and M. B. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28:253–263, 2008.

[34] A. K. Katsaggelos S. D. Babacan, R. Molina. Bayesian compressive sensing using laplace priors. *IEEE Transactions on Image Processing*, 19:53–63, 2010.

[35] Y. Xue S. Ji and L. Carin. Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356, 2008.

[36] M. A. Saunders S. S. Chen, D. L. Donoho. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

[37] P. Schniter, L. C. Potter, and J. Ziniel. Fast Bayesian matching pursuit. *Information Theory and Applications Workshop*, pages 326–333, January 2008.

[38] A. Smolic and P. Kauff. Interactive 3-d video representation and coding technologies. 93(1):98–110, January 2005.

[39] M. Tanimoto. Ftv (free viewpoint television) creating ray-based image engineering. In *Proc. ICIP*, volume 2, pages 25–28, September 2005.

[40] R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5), 1981.

[41] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50(10):22312242, 2004.

[42] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory*, pages 4655–4666, December 2007.

[43] M. Unser. Sampling-50 years after shannon. *Proceedings of the IEEE*, 88(4), 2000.

[44] M. Garofalakis W. Wang and K. Ramchandran. Distributed sparse random projections for refinabl approximation. *in Proc. Inf. Process. Sensor Networks (IPSN)*, pages 331–339, 2007.

[45] P. Beigi X. Xiu and J. Liang. Compressive sensing based multiview image coding with belief propagation. *Proc. 2010 IEEE Asilomar Conference on Signals, Systems, and Computers*, pages 430–433, 2010.

[46] X. Xiu, D. Pang, and J. Liang. Rectification-base view interpolation and extrapolation for multiview video coding. *IEEE Trans. on Circuits and Systems for Video Technology*, 21(6):693–707, June 2011.

[47] D. Farin Y. Morvan and P. H. N. de With. View interpolation along a chain of weakly calibrated cameras. *IEEE Workshop Content Generation and Coding for 3D TV*, June 2006.

[48] D. Farin Y. Morvan and P. H. N. de With. Multiview depth-image compression using an extended h.264 encoder. *Advanced Concepts for Intelligent Vision Systems*, pages 675–688, August 2007.