Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Multimediakommunikation und Signalverarbeitung

Prof. Dr.-Ing. André Kaup

Master Thesis

Examination Of Approximation Algorithms For Temporal-Spatial Prediction In Video Coding

by Haricharan Lakshman

April 2008

Supervisor: Dipl.-Ing. Jürgen Seiler

Abstract

In many modern video coding standards, the prediction of video signal contributes significantly to coding efficiency. Most existing prediction schemes exploit either only temporal or only spatial redundancy with Rate-Distortion optimization. This limitation is disadvantageous because a single video block could contain both temporal and spatial dependencies. In order to exploit these dependencies collectively, this thesis examines a prediction scheme which uses already transmitted frames as well as already transmitted neighboring blocks of the current frame. Since the decoder would also contain the same prediction scheme, only the prediction error needs to be transmitted, for the decoder to reconstruct the signal. One possibility of using spatial as well as temporal redundancies in prediction is when a motion compensated prediction is performed as a first step, followed by a modelling of the motion predicted signal along with its adjacent reconstructed blocks. It is shown previously that, using Frequency Selective Approximation (FSA) [1], a considerable reduction in spatial redundancy can be achieved. The focus of this thesis is to examine alternate modelling techniques for temporal-spatial prediction with reduced computational complexity. The newly designed prediction technique is implemented in H.264/AVC to evaluate its performance.

Acknowledgements

It is a pleasure to thank the many people who made this thesis possible.

I express my gratitude to my guide Prof. Dr.-Ing André Kaup and supervisor Dipl.-Ing. Jürgen Seiler for their continuous encouragement, support and advice throughout the time of this thesis.

I am deeply indebted to all my teachers, especially Prof. Walter Kellermann, Prof. Johannes Huber, Prof. Wolfgang Koch, Prof. Josef A. Nossek and my guide during undergraduate studies Prof. Subbanna Bhat. They provided me a lot of inspiration to pursue research in Signal Processing.

I would like to thank Siemens AG for providing financial support for foreign students like me for studying in Germany.

Lastly, and most importantly, I wish to thank my parents, Jayanthi Lakshman and K. H. Lakshman. I dedicate this thesis to them.

Contents

Contents **7**

1 Introduction **9**

2 Prediction In Video Coding **13**

2.1 Overview of Video Coding Layer 13

2.1.1 Prediction modes 14

2.2 Spatial refinement of Motion Compensated prediction block 16

2.2.1 Modelling of data 20

2.3 Scope and contribution of this thesis 22

3 Approximation Algorithms **23**

3.1 Approximation in Hilbert space 23

3.2 Linear vs Nonlinear approximation 25

3.3 Sparse approximation 26

3.3.1 Greedy schemes 30

3.3.2 Convex relaxation 31

4 Prediction Using Frequency Selective Approximation **33**

4.1 Introduction to Frequency Selective Approximation 34

4.1.1 DFT basis functions 38

4.1.2 DCT basis functions 38

4.2 Performance evaluation of FSA 40

4.2.1 Encoder Settings 42

4.3 Termination criteria 43
4.3.1 Oracle Assisted Stopping 45

5 Prediction Using Best Approximation **49**
5.1 Introduction to Best Approximation 50
5.1.1 Best Approximation Using DFT Basis Functions 53
5.2 Best Approximation with Relaxation 55
5.3 Speed Improvements 58
5.4 Design of weighting function 60
5.5 Termination criteria 62

6 Prediction Using Constrained Weighted Least Squares **67**
6.1 Formulation of Weighted Least Squares Approximation 68
6.2 Selection Of Constraints 69
6.2.1 Optimization Using Gradient Descent Method 70
6.3 Performance Evaluation Of CWLS 72
6.4 Analysis Of Results 74

7 Prediction Using Projections Onto Convex Sets **75**
7.1 Description of POCS method 76
7.2 Formulation of constraints 77
7.3 Performance evaluation of POCS 80

8 Summary and Future Work **83**

A Abbreviations **87**

B Additional PSNR plots **88**

List of Figures **93**

Bibliography **95**

Chapter 1

Introduction

Video coding deals with representation of video data for storage and/or transmission purpose. The search for efficient coding strategies has been on-going since the last few decades. The key issues in video compression include compression efficiency, computational complexity, frame rate, error robustness etc. The general purpose compression techniques that remove statistical redundancies by entropy coding are more suitable for text files but perform poorly for image and video data. The solution is to add a model to represent image/video data in a form that can be easily compressed by the entropy encoder. The model can be designed based on subjective redundancies for achieving higher compression. In such cases, the decoded image may not be identical to original image.

An important step in video coding is the prediction of video data. This step increases compression efficiency significantly becuase only the prediction error needs to be stored. For instance, the amount of data to be coded can be reduced significantly if the previous frame is subtracted from the current frame, instead of coding the individual frames of a video sequence separately. In this case, the previous frame data is used as a prediction for the current frame. Furthermore, when motion in video is accounted, better prediction and hence higher compression can be achieved.

In this thesis, various algorithms are evaluated for realizing a new method of prediction for improved compression. It is based on a two step process in which the initial estimate of the block to be coded is obtained through motion compensation. It is followed by a modelling based on signal approximation of the motion predicted signal together with already reconstructed adjacent blocks. This yields a spatial refinement of the temporally estimated data leading to a spatio-temporal prediction scheme. Finally, the model parameters are used to generate samples in the region to be predicted.

The basic system of a Hybrid Video Coder is explained in Chapter 2. The importance of prediction in video coding is elucidated and an overview of prediction methods in current video coders is presented. The idea of spatio-temporal prediction is then introduced. A novel design for realizing spatio-temporal prediction using motion compensated data is briefly described.

In order to build a foundation for discussing the methods adopted in this thesis, Chapter 3 provides a literature survey of Approximation Theory. It also analyzes the special properties of sparse approximation that are suitable for the current application. Later sections focus on various approaches, especially greedy schemes, for producing sparse representations of image data.

The reference implementation of the new prediction scheme, at the start of this thesis, is based on an algorithm called Frequency Selective Approximation (FSA) [2]. In Chapter 4, this algorithm is introduced and its performance is analyzed. The importance of the termination criteria in this iterative algorithm is examined and experimentally shown for various video sequences.

According to approximation theory, FSA can be classified as a greedy algorithm. An improvement over such schemes is the Orthogonal Greedy approach [3]. In Chapter 5, *Best Approximation*, which is Orthogonal Greedy, is evaluated. A direct implementation of Best Approximation results in a performance decrement compared to FSA. Nonetheless, a simple modification to this algorithm is shown to provide similar

performance compared to FSA but with much lesser computationally complexity.

Prediction using spatial and temporal information is converted into an optimization problem in Chapter 6. Modelling a signal in terms of only a few parameters is framed as a constrained optimization problem. The cost function is designed so as to minimize the reconstruction error but is penalized for using more parameters.

In Chapter 7, another viewpoint of spatio-temporal prediction is presented in terms of projections. Initially, the theory of projections onto convex sets is elaborated. Two classes of signals, covering spatial domain properties and frequency domain properties, are defined. Motion compensated prediction data is modified iteratively in order to satisfy these properties.

Finally, in Chapter 8, results of all the evaluated algorithms are summarized. Ideas for improving the performance of these algorithms are also proposed.

Chapter 2

Prediction In Video Coding

Video coding for telecommunication applications has evolved through the development of ITU-T H.261, H.262 (MPEG-2), H.263, MPEG-4 Part-2 and the recent H.264/AVC [4] video coding standards. From the initial ISDN & T1, the coding has diversified to include PSTN, mobile wireless networks and Internet network delivery. There is considerable research in maximizing coding efficiency while diversifying network types for content delivery. The applications of video coding are videoconferencing, digital storage media, television broadcasting, Internet streaming, and communication.

2.1 Overview of Video Coding Layer

The video coding layer of popular standards follow the block-based hybrid video coding approach, in which each coded picture is represented as blocks of luma and chroma samples called MacroBlocks (MB). All luma and chroma samples are predicted and the resulting prediction residual is computed. The residual is transformed into frequency domain and then quantized in accordance to the available bitrate. These quantized residuals are entropy coded to reduce the required bits in a lossless manner. At the decoder, the quantized transform coefficients are then de-mapped and inverse trans-

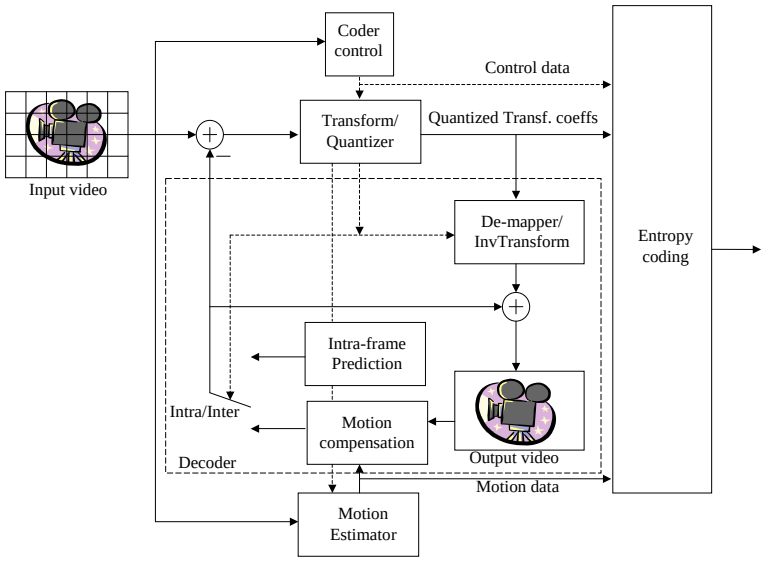

Figure 2.1: *Hybrid video coder. The rectangle in dashes depicts the embedded decoder.*

formed. The predicted signal is combined with the signal after inverse transform to produce the output video. At the encoder, the same combination process is performed in order to generate a closed-loop reference picture for coding of future frames. The block diagram of a Hybrid Video Coder is depicted in Fig. 2.1.

2.1.1 Prediction modes

Prediction plays a very important role in building practical data compression systems. According to Information Theory, in order to best exploit redundancy in data, joint entropy of entire data has to be considered. But such a scheme would be computationally complex and impractical for implementing video compression systems. However, if the signal to be coded can be predicted, the residual generally has greatly reduced statistical dependencies between adjacent samples. This facilitates the design a simpler compression system, based on the entropy of prediction residual, instead of joint entropy of the original signal.

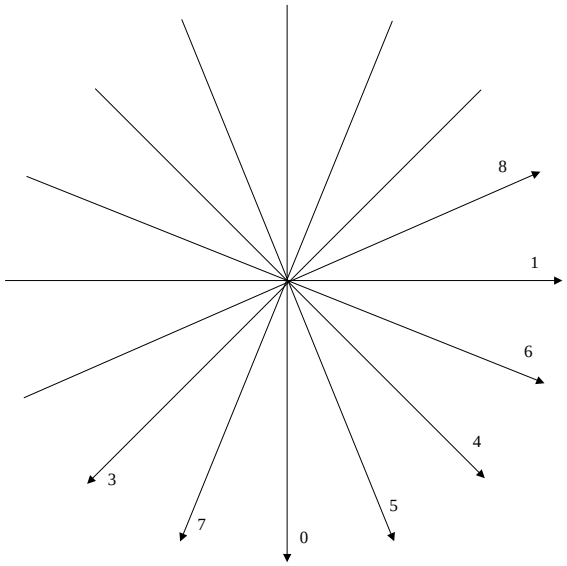

Figure 2.2: *Example Intra-Prediction. Eight prediction directions for Intra 4 × 4 mode in H.264/AVC.*

There are two kinds of prediction in contemporary video coders:

- Intra Prediction: Samples to be coded are predicted based on spatially neighboring samples. There are several prediction modes that use prior decoded samples of adjacent blocks. In addition to DC prediction, several directional prediction modes are employed. These modes are suitable to capture directional structures at different angles.

- Inter Prediction: Prediction signal for a macroblock is obtained by searching a region in reference picture that best matches the block to be coded according to a specified criterion. A translational motion vector is used to specify the inter-predicted signal. Techniques such as fractional sample accuracy, bi-directional prediction and multiple reference pictures provide additional gain.

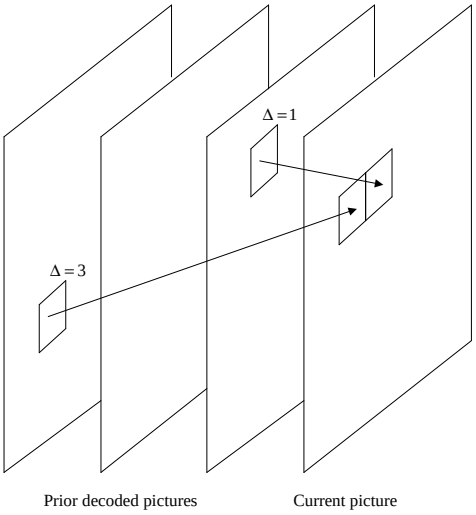

Figure 2.3: *Motion Compensated Prediction. When using multiple reference pictures, a translatioinal motion vector and reference picture index are transmitted in bitstream.*

2.2 Spatial refinement of Motion Compensated prediction block

When a macroblock is being coded, the data available at the encoder are the reconstructed samples of previously coded pictures and the samples of current picture that are already coded.

- I mode: Spatial information used, Temporal information ignored
- P, B mode: Temporal information used, Spatial information ignored

In case of smooth motion, motion compensation gives useful data that is similar to original block. Typically, motion compensation performs well at the block centers and poorly at the edges [5]. Many times, there are observable discontinuities at the edges of the predicted block.

The existing prediction algorithms in modern standards like H.264/AVC use either only temporal or only spatial data with Rate-Distortion (RD) optimization. This thesis attempts to combine these information and design a spatio-temporal scheme that can better predict the block to be coded. The techniques for directly using both spatial and temporal information in one-step are computationally complex because of the high volume of the 3D data to be handled. Also, they do not provide good gain over RD optimized I/P decisions, for the computational cost that is spent. In this thesis, methods are investigated to use motion compensated prediction block as a preliminary estimate for the block to be coded which is then refined by incorporating spatial information from neighboring blocks. Such a scheme would also keep the computational complexity under control as it operates on 2D data after motion compensation, instead of direct 3D data.

The technique examined in this thesis is to approximate the set of blocks (current block along with the neighboring reconstructed blocks) using parametric models as shown in Fig. 2.4. In computation of parameters of the model, the temporal data (motion predicted block) together with spatial data (reconstructed blocks) are used. Once the model is built, sample values at the desired locations can be generated using the model parameters. This is then used as the new predicted block for the computation of error signal to be entropy coded.

At times, the motion predicted block could also be an accurate estimate of the block to be coded. In such cases, any further processing/refinement of the estimated block may only hamper the quality of prediction. In such situations, it is best to retain the existing predicted data. A new prediction mode should therefore be able to work in conjunction with existing prediction schemes. Fig. 2.5 shows the seemless integration of the new prediction into RD optimized decision scheme which compares the different predicted blocks and chooses a prediction mode in an optimal way.

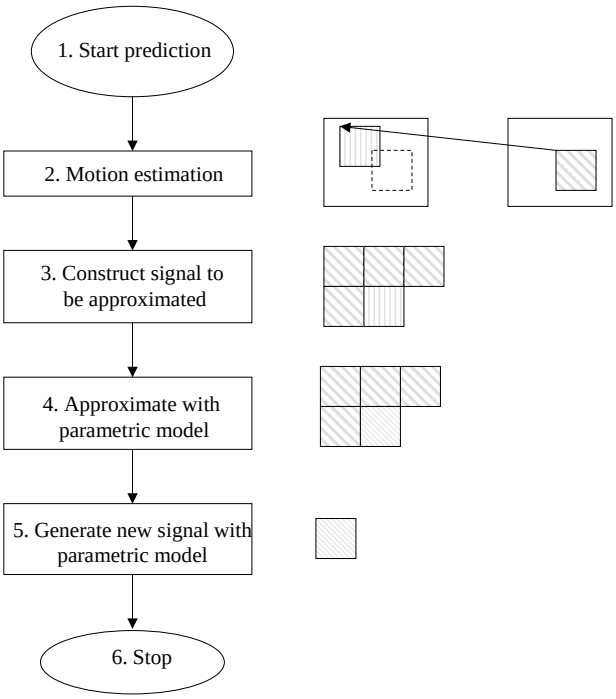

Figure 2.4: *Flow chart for spatial refinement of motion compensated prediction block. The main focus of this thesis is to examine different approximation algorithms for spatial refinement (Step 4 in flowchart)*

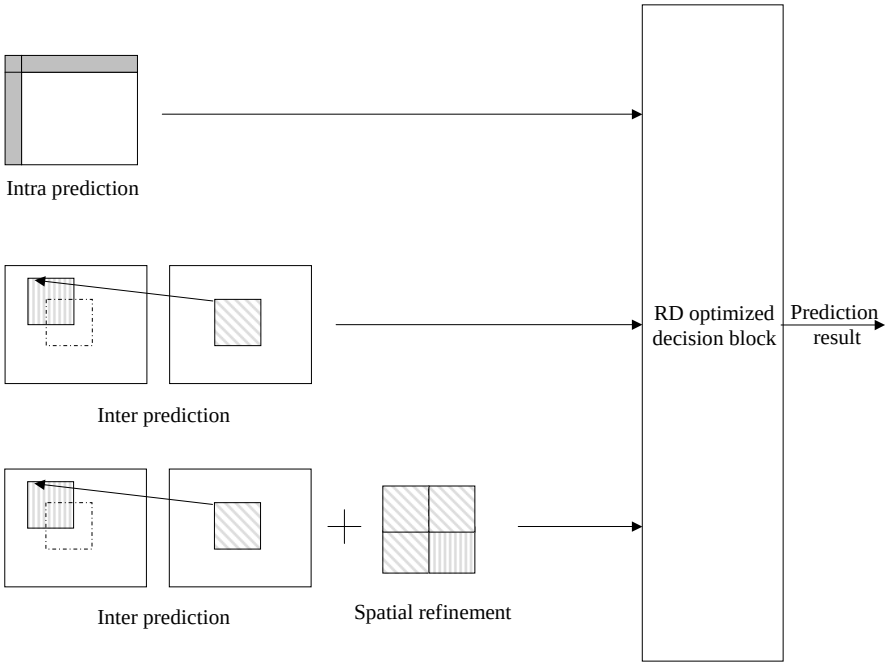

Figure 2.5: *RD Optimizated Decision. Integration of new prediction mode into decision scheme.*

2.2.1 Modelling of data

As can be seen in Fig. 2.4, the building of a parametric model is the crucial step in obtaining refinement over the MC predicted block. This section introduces the basics of procedure used in this thesis.

When examining data of high dimensionality, we generally look for subsets of data that have 'interesting' structure. This structure can take the form of trends, clusters, hypersurfaces, edges or anomalies. However, with high-dimensional data this is made difficult by the fact that higher dimensional space is quite often very sparse. In addition, the structure may span any arbitrary subspace of the data, thus increasing the computational cost required to locate the structure algorithmically.

The traditional approach to examining these high dimensional data is to reduce their dimensionality. Humans are very good at visual pattern recognition, and projecting the data set to low dimensions allows this ability to be utilized.

To describe the parametric modelling mathematically, consider the 1D-vectorized form of image data to be $\mathbf{f}$. Let $\mathbf{c}$ be the parameters that are used to model the signal when operated through a matrix $\mathbf{\Phi}$. The signal $\mathbf{f}$ can be considered as the original signal $\mathbf{f_0}$ corrupted by motion compensation noise η.

$$\mathbf{f} = \mathbf{f_0} + \eta.$$

The modelling can be expressed as

$$\mathbf{f} = \mathbf{\Phi c}.$$

The matrix $\mathbf{\Phi}$ needs to be chosen so that it provides a good bases for image data (e.g. DFT, DCT, Wavelet, Redundant bases etc). When there are no constraints on the parameters $\mathbf{c}$, the modelling is a simple linear system of equations that can be solved to obtain $\mathbf{c}$. In such a scenario, the modelling is perfect and original data is synthesized exactly. Nonetheless, we regenerate the same signal that we started with.

The main idea of spatial refinement is to impose constraints on $\mathbf{c}$ in order to discover the structures in $\mathbf{f}$ and to reduce the motion compensation noise (prediction error) η.

In such a framework, the refinement problem can be treated as a de-noising problem. The major difference to other de-noising problems is that the motion compensation noise is highly non-stationary and an assumption about the noise statistics (e.g. I.I.D. gaussian distribution) undermines the quality of refinement.

The modelling can be treated as an optimization problem which tries to minimize the error function

$$E = ||\mathbf{f} - \mathbf{\Phi c}||^2 = (\mathbf{f} - \mathbf{\Phi c})^T(\mathbf{f} - \mathbf{\Phi c}).$$

When a constraint on the parameters is included in the optimization, it can be written as

$$E = (\mathbf{f} - \mathbf{\Phi c})^T(\mathbf{f} - \mathbf{\Phi c}) + \lambda \{\text{constraint}(\mathbf{c})\}.$$

The samples in the reconstructed block that are close to the motion compensated predicted block are more useful for incorporating the spatial information than the samples that are far from it. Hence, the modelling error needs to be emphasized near the vicinity of the current block. This can be done by adding a weighting factor in the error calculation. Let $w_1, w_2, w_3, ...w_n$ be weights for errors at different locations, a matrix $\mathbf{W}$ is formed consisting of these weights at the diagonal positions.

$$\mathbf{W} = \text{diag}\{w_1, w_2, w_3, ...w_n\}.$$

The optimization can now be modified as

$$E = (\mathbf{f} - \mathbf{\Phi c})^T\mathbf{W}(\mathbf{f} - \mathbf{\Phi c}) + \lambda \{\text{constraint}(\mathbf{c})\}.$$

Further details regarding the solution of the optimization problem and the design of constraints on the parameters are elaborated in Chapter 3.

2.3 Scope and contribution of this thesis

At the starting of this thesis work, the idea of spatial refinement of motion compensated prediction block was experimented using Frequency Selective Approximation (FSA). This implementation will be referred to as FSA_{ref} in this thesis. The PSNR improvement over H.264/AVC JM10.2 is recorded in chapter 4. However, the main diasdvantage of FSA_{ref} is its computational complexity. The FSA_{ref} is an iterative algorithm which needs the result of one iteration to perform the next iteration, thereby introducing dependencies and precluding parallel execution of the algorithm. The aim of this thesis is the design of approximation algorithms that are suitable for providing good PSNR gain over standard H.264/AVC, but keeping the computational complexity under control. Some of the main contributions of this work are - examination of Best Approximation for prediction, formulation of spatial refinement task as a constrained weighted least squares optimization problem, evaluation of alternating projections method and design of stopping criteria for approximation.

Chapter 3

Approximation Algorithms

Approximation theory is concerned with approximating functions of a given class using functions from another class, usually more elementary. A simple example is the problem of approximating a speech signal by means of sinusoidal functions. In this chapter, various methods of approximation techniques are discussed and their applicability to spatial refinement is analyzed.

The concept of approximation in Hilbert space is first introduced in Sec. 3.1. Then, methods of linear and non-linear approximation are compared in Sec. 3.2. For capturing important spatial properties, the idea of sparse approximation is discussed in Sec. 3.3. Finally, an overview of different categories of sparse approximation algoithms is presented.

3.1 Approximation in Hilbert space

Assume we have a N-dimensional Hilbert space S and we wish to approximate an element $\mathbf{f} \in S$. The approximation is performed by signal expansion in terms of linear combination of some basic synthesis signals. If we design a set of functions $\{\boldsymbol{\varphi}_i\}_{i=1}^{N}$ such that all functions in S can be represented, then we say that it is *complete* in S. If

furthermore the elements are linearly independent, then we say that $\{\boldsymbol{\varphi}_i\}_{i=1}^{N}$ is a *basis* for S.

Basis functions and Basis vectors

The 2D basis functions $\boldsymbol{\varphi}_i$ can be parsed columnwise and a long vector containing concatenated columns of $\boldsymbol{\varphi}_i$ can be formed. In that case, these basis functions in $N \times N$ dimensional space become vectors in N^2 dimensional space. Many a times, it is convenient to interpret basis functions as such vectors and hence are called basis vectors. In this thesis, the terms basis functions and basis vectors are used interchangeably, but refer to the same elementary functions used to model a signal.

Linear combination of basis functions

Any element $\mathbf{f} \in S$ can be written as:

$$\mathbf{f} = \sum_{i=1}^{N} c_i \boldsymbol{\varphi}_i.$$

In matrix notations, the signal may be represented using some transform coefficients, denoted by $\mathbf{c}$. The forward transform, which we denote by $\boldsymbol{\Phi}^{-1}$, is used to compute the transform coefficients. Hence, the analysis equation can be written as, $\mathbf{c} = \boldsymbol{\Phi}^{-1}\mathbf{f}$. The reconstructed signal is given by the corresponding synthesis equation:

$$\mathbf{f} = \boldsymbol{\Phi}\mathbf{c}.$$

The synthesis vectors $\boldsymbol{\varphi}_i$ are the columns of the matrix $\boldsymbol{\Phi}$. They should form a basis for S since this is necessary for the inverse $\boldsymbol{\Phi}^{-1}$ to exist. The reconstructed signal is built up as a linear combination of these synthesis vectors.

If the possibility of having more than N terms in the linear combination is considered, say $K > N$ terms, the collection of K vectors will be denoted as $\{\boldsymbol{\varphi}_i\}_{i=1}^{K}$. These vectors

are collectively referred to as a *frame* or a *dictionary*, as columns of an $N \times K$ matrix $\mathbf{\Phi}$. The synthesis equation for the frame case has the same form as in the transform case:

$$\mathbf{f} = \mathbf{\Phi w} = \sum_{i=1}^{K} w_i \boldsymbol{\varphi}_i.$$

Here, the transform coefficients are replaced by the weights used for each synthesis vector. Depending on the weights we set for different basis vectors, we get different approximations of the original function $\mathbf{f}$. A perfect reconstruction of $\mathbf{f}$ is also possible.

Orthogonal bases are particularly desirable among all possible bases. The basis functions are all mutually orthogonal in that case. Additionally when these vectors are normalized, they satisfy:

$$\langle \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j \rangle = \delta_{ij},$$

where the $\langle ., . \rangle$ denotes inner product.

When the bases are orthonormal, the expansion formula can be written as:

$$\mathbf{f} = \sum_{i=1}^{N} \langle \mathbf{f}, \boldsymbol{\varphi}_i \rangle \boldsymbol{\varphi}_i$$

3.2 Linear vs Nonlinear approximation

Assume a space V and an orthonormal basis $\{\boldsymbol{\varphi}_i\}_{i \in N}$ for V. A vector $\mathbf{f} \in V$ can be expressed as a linear combination

$$\mathbf{f} = \sum_{i \in N} \langle \mathbf{f}, \boldsymbol{\varphi}_i \rangle \boldsymbol{\varphi}_i$$

Consider a subspace W of M dimensions and spanned by M vectors of the basis.

$$W = \text{span}(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \boldsymbol{\varphi}_3, ..., \boldsymbol{\varphi}_M).$$

Once the basis vectors are selected, the subspace spanned by these vectors gets fixed. It is a fixed subspace because the same basis vectors are employed independent of the

signal to be modelled. Then, we can construct the approximant $\hat{\mathbf{f}}$ by projecting $\mathbf{f}$ onto the fixed M-dimensional subspace.

$$\hat{\mathbf{f}} = \sum_{i=1}^{M} \langle \mathbf{f}, \boldsymbol{\varphi}_i \rangle \boldsymbol{\varphi}_i$$

The squared approximation error becomes

$$\hat{\epsilon}_M = ||\mathbf{f} - \hat{\mathbf{f}}||_2^2 = \sum_{i=M+1}^{N} |\langle \mathbf{f}, \boldsymbol{\varphi}_i \rangle|^2.$$

Since each signal is projected onto the same subspace W, independent of $\mathbf{f}$, it makes the approximation linear. This follows from the fact that $\alpha\mathbf{f}_1 + \beta\mathbf{f}_2$ is equal to the weighted sum of approximations, $\alpha\hat{\mathbf{f}}_1 + \beta\hat{\mathbf{f}}_2$.

Suppose we choose the subspace W depending on the signal $\mathbf{f}$, we can select the basis vectors that best model $\mathbf{f}$ and it results in a Nonlinear approximation. Consider the case when we select the largest M coefficients of the signal expansion. Thus, we define an index set I_M of the largest inner products $|\langle \mathbf{f}, \boldsymbol{\varphi}_m \rangle| \geq |\langle \mathbf{f}, \boldsymbol{\varphi}_n \rangle|$ for every $m \in I_M$ and $n \notin I_M$. Then we define the nonlinear approximation as:

$$\tilde{\mathbf{f}} = \sum_{i \in I_M} \langle \mathbf{f}, \boldsymbol{\varphi}_i \rangle \boldsymbol{\varphi}_i$$

The approximation error becomes:

$$\tilde{\epsilon}_M = ||\mathbf{f} - \tilde{\mathbf{f}}||_2^2 = \sum_{i \notin I_M} |\langle \mathbf{f}, \boldsymbol{\varphi}_i \rangle|^2.$$

3.3 Sparse approximation

The spatial refinement step using approximation algorithms aims at combing spatial properties from the nearby blocks with the motion compensated block. The idea of approximation is to come close to the signal being approximated such that it captures most important characteristics of the signal. But, if we approximate too well, we get

the input signal again. Since the approximation algorithms are capable of producing arbitrarily close representations of the signal $\mathbf{f}$, we need to impose some constraints on the algorithm, in order not to generate the motion predicted image again, which would defeat the purpose of approximation. In the de-noising model, we need to generate a signal that is close to $\mathbf{f}_0$, thereby reducing motion conpensation error.

One key constraint on $\mathbf{c}$, based on research on natural images, is the sparsity of its components $\{c_i\}_{i=1}^N$. There are three common justifications:

- The main reason for believing that sparsity of $\mathbf{c}$ is an appropriate prior is based on the intution that natural images are generally composed of a small number of structural primitives, for example - edges, lines or other elementary features [6]. We can also see this by filtering with log-gabor functions and collecting histograms of the resulting output distributions. These distributions show high kurtosis which is indicative of sparse structure [6]. According to results in human vision research, there are many other biological reasons for desiring sparsity - increasing capacity in associative memory, minimizing wiring length and ease of forming associations, or metabolic efficiency.

- The approximation generally has an associated cost that must be controlled. For example, the computational complexity of approximation depends on the number of elementary functions that are used for the representation. So, in order not to have high complexity, we need to impose sparsity.

- The third important reason is due to the principle of Occam's razor [7]. It states that 'Causes must not be multiplied beyond necessity'. This is usually employed by statisticians in selecting models for inference. Among different methods for modelling the same data, statisticians usually select the *simpler* method or the model that has *less* number of parameters. So, for representation of signal $\mathbf{f}$ using $\{c_i\}_{i=1}^N$, an appropriate choice based on Occam's razor would be to model it with less number of non-zero c_i.

In this thesis, both sparse and non-sparse approximation techniques are examined. Sparse approximation algorithms were observed to perform well for the spatial refinement. This is intutive as sparse representations are known to capture important structure within the signal.

Sparse representation vs. Coarse representation

The notion of sparse representation, where relatively small number of units are used to represent the signal, would be opposite to that of coarse representation or population codes, in which large number of units participate in representing a signal. In sparse representation, when the learning of basis vectors is allowed, they are broadly tuned to some stimulus dimensions (e.g. spatial frequency) while narrowly tuned to others (position) [6]. Dense population codes are appropriate when single or a few attributes are needed to be encoded. When many different attributes are to be simultaneously represented, introducing population codes would cause blurring. Thus sparse representation and coarse representation are appropriate for different circumstances.

Measures of Sparsity

In order to utilize sparsity as a constraint in an approximation problem, we need to first define sparsity in mathematical terms. It would best be to define sparsity as the number of places where the coefficients of approximation equals zero. Consider the L_p norm:

$$||\mathbf{c}||_p = \left[\sum_{i=1}^{N} |c_i|^p\right]^{1/p}.$$

In order to count the number of zero elements in the vector $\mathbf{c}$, we can set $p = 0$. However, it is well known that L_p function is convex, if and only if, $1 \leq p \leq \infty$. So, although L_0 gives a measure of sparsity, it cannot be directly used in an optimization problem as it yields a non-convex function.

Problem statements for Approximation

If we are interested in the exact representation of a signal $\mathbf{f}$, we can state the optimization problem as:

$$\min_{\mathbf{c}\in\mathcal{C}} ||\mathbf{c}||_0 \quad \text{subject to} \quad \mathbf{f} = \Phi\mathbf{c}.$$

But for the spatial refinement, we need to seek the sparsest representation with a prescribed approximation error. The approximation must not introduce too great an error. But at the same time, the objective function should be penalized for every additional term used in the linear combination. For a fixed error tolerance ϵ, we wish to solve the optimization problem:

$$\min_{\mathbf{c}\in\mathcal{C}} ||\mathbf{c}||_0 \quad \text{subject to} \quad ||\mathbf{f} - \Phi\mathbf{c}||_2 < \epsilon.$$

In approximation theory, another version of the sparse approximation problem is preferred, called sparsity constrained optimization. Many times, the actual value of the approximation error cannot be set a priori. In such cases, the number of dictionary atoms participating in the approximation is limited. The problem can be restated as to provide the best approximation using m or fewer atoms from the dictionary:

$$\min_{\mathbf{c}\in\mathcal{C}} ||\mathbf{f} - \Phi\mathbf{c}||_2 \quad \text{subject to} \quad ||\mathbf{c}||_0 \leq m.$$

The optimization problem using L_0 quasi-norm is proved to be NP-Hard [8]. A brute force search over all possible $\mathbf{c}$ is required to find the best solution. Over the last few decades, many different methods have been proposed to solve sparse approximation problems. They fall under two broad categories:

- Greedy schemes (Boosting, Projection pursuit, Matching pursuit, Frequency selective)
- Convex relaxation (L_1 regularization, L_0-SVM)

3.3.1 Greedy schemes

If the dictionary is orthonormal, the signal expansion can be written as $\sum_{i \in I} \langle \mathbf{f}, \boldsymbol{\varphi}_i \rangle \boldsymbol{\varphi}_i$. We can sort the terms such that the inner products are non-increasing. We may then truncate the series after m terms to obtain the optimal m-term approximation. The coefficients c_i are just the inner products. Hence, the idea is that, we choose the atoms that have the largest absolute inner products with the target signal. In an iterative implementation, this can be accomplished by choosing the atom which has the most strong correlation with the signal, subtracting its contribution from the signal and repeating these steps again. Greedy schemes operate on the same principle and refine this to be applied to more general dictionaries.

Similar techniques have been proposed by different research communities with different names:

- Statistics - Forward stagewise regression
- Approximation Theory - Greedy algorithms
- Machine Learning - Boosting methods
- Signal Processing - Projection pursuit

The algorithms are based on the iteration of the following steps after some initialization:

- Selection of an atom from the dictionary
- Update of the solution

Projection Pursuit Regression was invented in 1981 [9]. In 1992, an algorithm based on successive approximation was proposed by Kaup and Aach for processing arbitrarily shaped image segments [10]. Projection pursuit algorithm was reintroduced into Signal Processing, under the name Matching Pursuit, by Mallat and Zhang for overcomplete dictionaries [11].

The reference implementation at the start of this thesis, FSA_{ref}, uses an improved

version of algorithm designed by Kaup and Aach and is an example of Greedy scheme. The main difference between FSA and other greedy schemes is that FSA allows seemless addition of weighting function into the optimization problem and operates completely in frequency domain. The weighting function is very critical in the spatial refinement process as it allows us to emphasize the samples near the block of interest during the calculation of approximation error.

3.3.2 Convex relaxation

The non-convex L_0 function makes the sparse approximation problem combinatorial in nature. Combinatorial problems are generally solved by replacing them with a relaxed version that can be solved more efficiently. Under the name Basis Pursuit [12], it was showed that L_1 norm provides a natural relaxation of the L_0 quasi-norm and suggests that L_1 can be used in place of L_0 for solving sparse approximation problems.

The convex relaxation can be mathematically stated as:

$$\min_{\mathbf{c}\in\mathcal{C}} ||\mathbf{c}||_1 \quad \text{subject to} \quad \mathbf{f} = \Phi\mathbf{c}.$$

This requires the solution of a convex, non-quadratic optimization problem. It involves considerable effort and sophistication and hence the resulting algorithms are computationally complex. Although convex relaxation produces sparse representations, they are not evaluated in this thesis because of their high computational complexity compared to greedy schemes.

Chapter 4

Prediction Using Frequency Selective Approximation

The Frequency Selective Approximation (FSA) algorithm is based on the principle of successive approximation. It is an example of 'Greedy scheme' as discussed in Chapter 3 where the signal is approximated by a weighted linear combination of basis functions. There has been considerable research in developing FSA for *extrapolation* into unknown areas [1, 13] and for texture *representation* [10]. This thesis is concerned with designing approximation schemes such that it produces a *refinement* of preliminary temporally estimated data.

In this chapter, the method of FSA is introduced for generic basis functions. It is then applied to spatial refinement using Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) basis functions. A method is developed for implementing the algorithm entirely in frequency domain. The performance of the algorithm for these basis functions is compared.

Since FSA is capable of producing arbitrarily close approximation of the signal being modelled, a termination criteria needs to be established. The last section of this chapter investigates the rules for stopping the algorithm for best results.

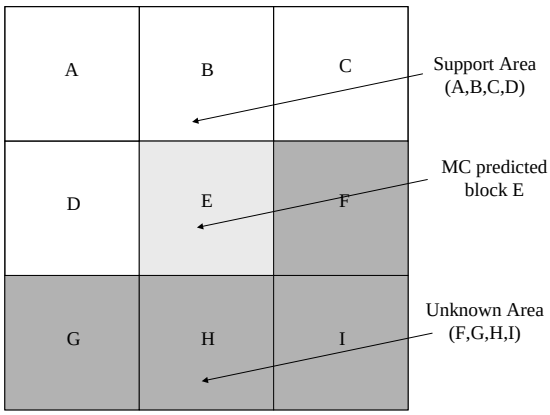

Figure 4.1: *Adjacent blocks in a frame. The blocks A,B,C,D contain reconstructed samples, block E holds samples from motion compensated prediction, blocks F,G,H,I have future samples which are not yet known at the decoder.*

4.1 Introduction to Frequency Selective Approximation

The derivation of FSA algorithm is discussed briefly in this thesis, as it forms the background for explaining modifications necessary for refinement application. For exact derivation, please refer to [1].

Fig. 4.1 shows a part of a video frame being coded. Consider the blocks A, B, C and D to be the reconstructed data of already coded blocks. Let block E be the motion compensated prediction block. Therefore, the areas shaded in dark gray constitute the future data and are not yet known at the decoder.

Let $f[m, n]$ denote the intensities of the samples of the entire area $\mathcal{L}$ shown as blocks A to I in Fig. 4.1. Let the number of samples in $\mathcal{L}$ be $M \times N$. The available signal can be interpreted as the original signal multiplied with a binary window function $b[m, n]$

which equals zero in the unknown area and unity elsewhere.

$$b[m,n] = \begin{cases} 1 & (m,n) \in A,B,C,D,E \\ 0 & (m,n) \in F,G,H,I \end{cases} \tag{4.1}$$

Let the approximated signal be denoted by $g[m,n]$ which is defined in the entire area. The signal in blocks A to E is modelled using a parametric model. Finally, the model parameters are used to generate new samples $g[m,n]$ for the block E which replaces the motion predicted data in that area, thereby obtaining spatial refinement.

The parametric model used is a linear combination of a few basis functions $\varphi_{k,l}$ with appropriate weights $c_{k,l}$.

$$g[m,n] = \sum_{(k,l)\in\mathcal{K}} c_{k,l}\varphi_{k,l}[m,n] \tag{4.2}$$

where $\mathcal{K}$ represents the set of basis functions used in the model. This approach is developed for a generic set of basis functions.

The error energy between the signal $f[m,n]$ and its approximation $g[m,n]$ controls the amount of approximation by the estimated signal in the entire area $\mathcal{L}$.

$$E_b = \sum_{(m,n)\in\mathcal{L}} b[m,n](f[m,n] - g[m,n])^2 \tag{4.3}$$

For the purpose of refinement, the samples close to the block E are of more importance than the ones that are far away. This non-uniform importance can be incorporated into error computation through the means of a weighting function $w[m,n]$. The weighting function has only positive values $\rho[m,n]$ in the blocks A-E and zero elsewhere. The weighting function significantly affects the refinement quality.

$$w[m,n] = \begin{cases} \rho[m,n] & (m,n) \in A,B,C,D,E \\ 0 & (m,n) \in F,G,H,I \end{cases} \tag{4.4}$$

The weighted instantaneous error energy E can now be written as:

$$E = \sum_{(m,n)\in\mathcal{L}} w[m,n](f[m,n] - g[m,n])^2 \tag{4.5}$$

To determine the expansion coefficients $c_{k,l}$ of the parametric model, the error E is minimized by taking partial derivatives with repect to the unknown coefficients and equating them to zero:

$$\frac{\partial E}{\partial c_{k,l}} \stackrel{!}{=} 0. \tag{4.6}$$

This gives rise to a set of linear equations:

$$\sum_{(k,l)\in\mathcal{K}} c_{k,l} \sum_{(m,n)\in\mathcal{L}} w[m,n]\varphi_{u,v}^2[m,n] = \sum_{(m,n)\in\mathcal{L}} w[m,n]f[m,n]\varphi_{u,v}[m,n]. \tag{4.7}$$

Since the number of known samples is less than the total number of points considered $M \times N$, the error minimization leads to an underdetermined system of equations. To overcome this problem, a successive approximation scheme is used. It is an iterative procedure, in which the signal in A-E is successively approximated in terms of one basis function per iteration. This involves the selection of a basis function and computation of the optimal expansion coefficient. Thus the algorithm approximates the signal in terms of dominant features through selected basis functions. This yields the parametric model $g[m,n]$ which contains the refined values. Finally, the block of interest in the motion predicted area is cut out of the model $g[m,n]$ and used in further modules of video compression.

Update of selected coefficient

When the modelling is done iteratively, a new model is generated in every iteration. In iteration ν, let the model be denoted as $g^{(\nu)}[m,n]$ and the set of basis functions selected upto iteration ν be denoted as $\mathcal{K}^{(\nu)}$.

$$g^{(\nu)}[m,n] = \sum_{(k,l)\in\mathcal{K}^{(\nu)}} c_{k,l}^{(\nu)}\varphi_{k,l}[m,n]. \tag{4.8}$$

In the beginning, for $\nu = 0$, $\mathcal{K}^{(\nu)}$ will be an empty set. The residual error at iteration ν can be expressed as

$$r^{(\nu)}[m,n] = (f[m,n] - g^{(\nu)}[m,n])b[m,n]. \tag{4.9}$$

Let $\varphi_{u,v}$ be the chosen basis function and Δc be the change in expansion coefficient, the residual error at the next iteration is then

$$r^{(\nu+1)}[m,n] = (r^{(\nu)}[m,n] - \Delta c \varphi_{u,v}) b[m,n]. \tag{4.10}$$

The new weighted error energy becomes:

$$E^{(\nu+1)} = \sum_{m,n \in \mathcal{L}} w[m,n](r^{(\nu)}[m,n] - \Delta c \varphi_{u,v}[m,n])^2 \tag{4.11}$$

In order to obtain Δc, the weighted residual error energy is minimized with respect to the unknown coefficient

$$\frac{\partial E^{(\nu+1)}}{\partial \Delta c} \stackrel{!}{=} 0. \tag{4.12}$$

This yields

$$\Delta c = \frac{\sum_{(m,n)\in\mathcal{L}} w[m,n] r^{(\nu)}[m,n] \varphi_{u,v}[m,n]}{\sum_{(m,n)\in\mathcal{L}} w[m,n] \varphi_{u,v}^2[m,n]} \tag{4.13}$$

The expansion coefficient is then updated by $c_{u,v}^{(\nu+1)} = c_{u,v}^{(\nu)} + \Delta c$ and the index (u,v) is included in the set of bases:

$$\mathcal{K}^{(\nu+1)} = \mathcal{K}^{(\nu)} \cup (u,v) \quad \text{if} \quad (u,v) \notin \mathcal{K}^{(\nu)}. \tag{4.14}$$

Selection of basis function

The basis function is selected according to error minimization criteria:

$$(u,v) = \underset{k,l}{\operatorname{argmax}} \, \Delta E^{(\nu+1)}. \tag{4.15}$$

where,

$$\Delta E^{(\nu+1)} = \Delta c^2 \sum_{(m,n)\in\mathcal{L}} w[m,n] \varphi_{k,l}^2[m,n] = \frac{(\sum_{(m,n)\in\mathcal{L}} w[m,n] r^{(\nu)}[m,n] \varphi_{k,l}[m,n])^2}{\sum_{(m,n)\in\mathcal{L}} w[m,n] \varphi_{k,l}^2[m,n]} \tag{4.16}$$

4.1.1 DFT basis functions

The 2D DFT basis functions are:

$$\varphi_{k,l}[m,n] = e^{j2\pi/Mmk} e^{j2\pi/Nnl} \tag{4.17}$$

The expressions derived in section 4.1 can be entirely implemented in the frequency domain when DFT basis functions are chosen. The detailed derivation of equations in frequency domain can be found in [1].

To ensure that the approximation $g^{(\nu)}[m,n]$ yields a real valued signal in each iteration, the parametric model is defined to take symmetry into account.

$$g^{(\nu)}[m,n] = \frac{1}{2MN} \sum_{(k,l)\in\mathcal{K}^{(\nu)}} (c_{k,l}^{(\nu)} \varphi_{k,l}[m,n] + c_{k,l}^{(\nu)*} \varphi_{M-k,N-l}[m,n]) \tag{4.18}$$

The equation for update of coefficient is:

$$\Delta c = \begin{cases} MN \frac{R_w^{(\nu)}[u,v]}{W[0,0]} & , (u,v) \in \mathcal{M} \\ 2MN \frac{R_w^{(\nu)}[u,v]W[0,0] - R_w^{*(\nu)}[u,v]W[2u,2v]}{W[0,0]^2 - |W[2u,2v]|^2} & , \text{else} \end{cases} \tag{4.19}$$

The basis function with index (u,v) is selected which maximizes

$$\Delta E^{(\nu)} = \begin{cases} 2\frac{R_w^{(\nu)}[k,l]^2}{W[0,0]} & , (k,l) \in \mathcal{M} \\ 2MN \frac{|R_w^{(\nu)}[u,v]|^2 W[0,0] - Re\{R_w^{*(\nu)}[k,l]^2 W^*[2k,2l]\}}{W[0,0]^2 - |W[2k,2l]|^2} & , \text{else} \end{cases} \tag{4.20}$$

4.1.2 DCT basis functions

DCT bases are periodic functions and provide a better concentration of energy in few coefficients compared to DFT, for certain limits of markov processes. Natural images are known to fall under these limits, and hence DCT is often used in image and video coding. Another advantage of DCT is that the basis functions are composed of only real values. It is also easy to integrate DCT based approach into video coders because most coders quantize DCT values of prediction error.

The DFT basis contains horizontal, vertical and diagonal orientations and provide phase relation additionally. But the DCT basis is composed only of horizontal and vertical orientations. For extracting the inherent structure in reconstructed regions, the DFT is theoretically better because of the presence of additional orientations [1].

Although DFT bases perform better for extrapolation [14], the refinement application demands additional properties of energy compaction. Hence DCT bases are also considered in this thesis.

The generic expression for the updation of weighted residual in FSA is

$$r_w^{(\nu+1)}[m,n] = r_w^{(\nu)}[m,n] - \Delta c \varphi_{u,v}[m,n] w[m,n] \tag{4.21}$$

In order to transform this equation to DCT domain, multiply it by $\varphi_{k,l}[m,n]$ and sum over all $(m,n) \in \mathcal{L}$.

$$\sum_{(m,n)\in\mathcal{L}} r_w^{(\nu+1)}[m,n]\varphi_{k,l}[m,n] = \sum_{(m,n)\in\mathcal{L}} r_w^{(\nu)}[m,n]\varphi_{k,l}[m,n] - \sum_{(m,n)\in\mathcal{L}} \Delta c \varphi_{u,v}[m,n]\varphi_{k,l}[m,n] w[m,n] \tag{4.22}$$

This can be written in terms of DCT coefficients of the weighted residual as

$$R_w^{(\nu+1)}[k,l] = R_w^{(\nu)}[k,l] - \Delta c \sum_{(m,n)\in\mathcal{L}} w[m,n]\varphi_{u,v}[m,n]\varphi_{k,l}[m,n] \tag{4.23}$$

To implement this equation completely in DCT domain the factor

$$\sum_{(m,n)\in\mathcal{L}} w[m,n]\varphi_{u,v}[m,n]\varphi_{k,l}[m,n]$$

has to be expressed in terms $W[k,l]$, the DCT coefficients of the weighting function.

Consider the orthogonal form of DCT4 basis functions:

$$\varphi_{u,v}[m,n] = \frac{2}{N}\cos(\frac{\pi}{N}(m+0.5)(u+0.5))\cos(\frac{\pi}{N}(n+0.5)(v+0.5)) \tag{4.24}$$

$$\varphi_{k,l}[m,n] = \frac{2}{N}\cos(\frac{\pi}{N}(m+0.5)(k+0.5))\cos(\frac{\pi}{N}(n+0.5)(l+0.5)) \tag{4.25}$$

The product $\cos(\frac{\pi}{N}(m+0.5)(u+0.5))\cos(\frac{\pi}{N}(m+0.5)(k+0.5))$ can be expressed as the sum of two cosine terms

$$\frac{1}{2}(\cos(\frac{\pi}{N}(m+0.5)(u+k+1)) + \cos(\frac{\pi}{N}(m+0.5)(u-k))) \tag{4.26}$$

Therefore the term $\sum_{(m,n)\in\mathcal{L}} w[m,n]\varphi_{u,v}[m,n]\varphi_{k,l}[m,n]$ simplifies into the sum of four cosine terms which in turn can be expressed as DCT2 of weighting function as

$$\frac{2}{4N}(W[u+k+1, v+l+1]+W[u+k+1, v-l]+W[u-k, v+l+1]+W[u-k, v-l]) \tag{4.27}$$

To compute the individual terms, the DCT2 of weighting function is first evaluated. It is then extended using periodicity properties to obtain the terms outside the first period. Note that this derivation uses DCT4 basis functions. However, in the process of expressing the new residual (which is calculated using DCT4), only the term with weighting function gets broken down to sum of DCT2 terms.

4.2 Performance evaluation of FSA

The H.264/AVC version JM10.2 was chosen as the underlying video coder for evaluating FSA for spatial refinement application. The block size was set at 16×16 and the motion compensated prediction was performed at quater-pel accuracy. This was then placed adjacent to three reconstructed blocks on the top and one on the left, constituting the known area in building the parametric model. Hence, the size of entire data was 48×48. In order to evaluate the frequency domain implementation of FSA, the data was zero-padded to form a block of size 64×64 and therefore $M = N = 64$. The simulations were performed on CIF video sequences of 'Vimto', 'Discovery', 'Crew', and 'Flower garden'.

The design of weighting function $w[m,n]$ for the purpose of error concealment is tackled in [2]. It uses an isotropic model with exponentially decaying values centered at the

middle of the lost block. The influence of the weighting function decreases radially symmetric with distance from the center. We define such a function as:

$$\rho[m,n] = \hat{\rho}^{\sqrt{(m-\frac{M-1}{2})^2+(n-\frac{N-1}{2})^2}} \tag{4.28}$$

The computation of the weighting function for error concealment is as follows:

$$w[m,n] = \begin{cases} \rho[m,n] & ,(m,n) \in \text{known area} \\ 0 & ,(m,n) \in \text{lost area} \end{cases} \tag{4.29}$$

In the reference implementation, FSA_{ref}, the same principle of isotropic weighting function was used but with a different fixed weight for the block area being refined. The new weighting function was defined as:

$$w[m,n] = \begin{cases} \hat{\rho}^{\sqrt{(m-\frac{M-1}{2})^2+(n-\frac{N-1}{2})^2}} & ,(m,n) \in A,B,C,D \\ 0.5 & ,(m,n) \in E \\ 0 & ,(m,n) \in F,G,H,I \end{cases} \tag{4.30}$$

Additionally, when weighting is applied to reconstruction error, it can be viewed as introducing orthogonality deficiency in basis vectors. It leads to a situation in which the basis vectors have non-zero components in the direction of other basis vectors. To counter this problem, a compensation scheme is proposed in [15]. The orthogonality deficiency compensation is based on the fact that other basis vectors have components in the direction of a selected basis vector and considering the projection of residual in the direction of selected basis vector as its expansion coefficient would be an overestimate. According to a fast implementation of orthogonality deficiency compensation [16], considering a part of projection length, instead of its whole length, would be sufficient to produce good results. In this thesis, a compensation factor of 0.5 is used to evaluate the performancnce of FSA.

4.2.1 Encoder Settings

For evaluating the improvement due to the proposed spatial refinement in H.264/AVC, the following changes were applied to the H.264/AVC codec:

- The insertion of Intra blocks in P frames was disabled
- Skip mode was disabled

All the H.264/AVC codec RD curves reported in this thesis are with the above mentioned modifications. The simulations were run for 100 frames. Some important configuration settings used for generating PSNR values are:

- Baseline profile
- IPPP mode (Intraperiod=0)
- B slices not used
- Sub-blocks not used
- RD optimization off

With these changes, spatial refinement was applied to luminance component to record the PSNR improvement. The algorithm was run for 200 iterations with $\hat{\rho} = 0.8$.

After performing refinement, the error between spatially refined signal and original video signal e_{sr} is compared with the error between motion predicted signal and original video signal e_{mc}. If $e_{sr} < e_{mc}$, the refined signal is taken as the new prediction, else the motion predicted signal is retained as the predicted video signal for further modules of encoder. This decision is communicated to the decoder using 1-bit side information for each block. The plots shown in this thesis include the overhead for transmitting this side information, unless stated otherwise.

The performance of FSA for *Vimto* and *Crew* sequences are recorded in Fig. 4.2 and Fig 4.3 respectively. The performance for *Discovery city* and *Flower garden* sequences

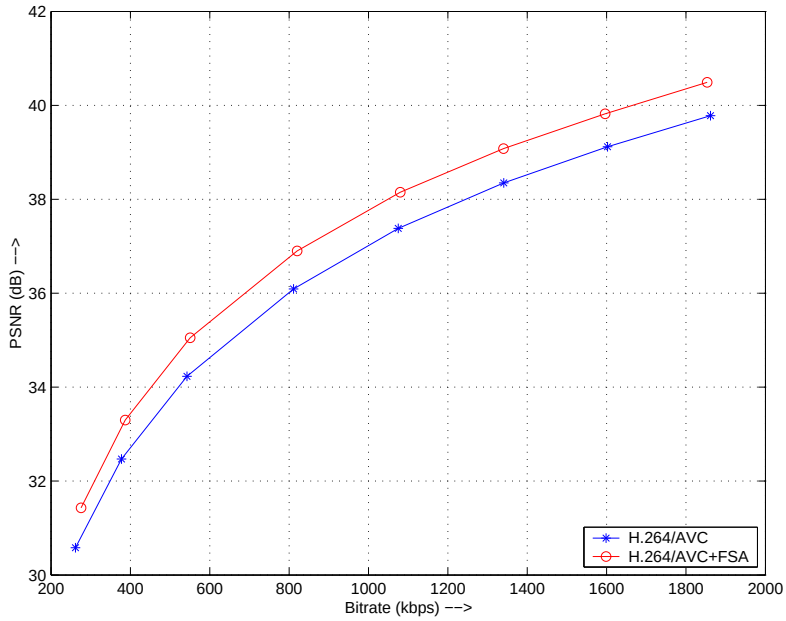

Figure 4.2: *FSA performance for Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations.*

are included in Appendix Fig. B.1 and Fig. B.2 respectively.

4.3 Termination criteria

Even when using sparse representations of observed data by means of linear combination of basis functions, it is possible to generate arbitrarily close approximations, and exact representations, when more and more basis functions are used in the modelling. This section discusses the importance of stopping the approximation process in order to avoid overfitting problems.

When the data being modelled is approximated too well, the block of interest resembles the initial motion compensated prediction data. In this case, the spatial refinement step would be of no benefit. If the approximation uses too few basis functions, the modelled data would differ signficantly from the initial data and could cause a loss in quality. Hence, we need to model the data upto a certain point, which yields a good combination

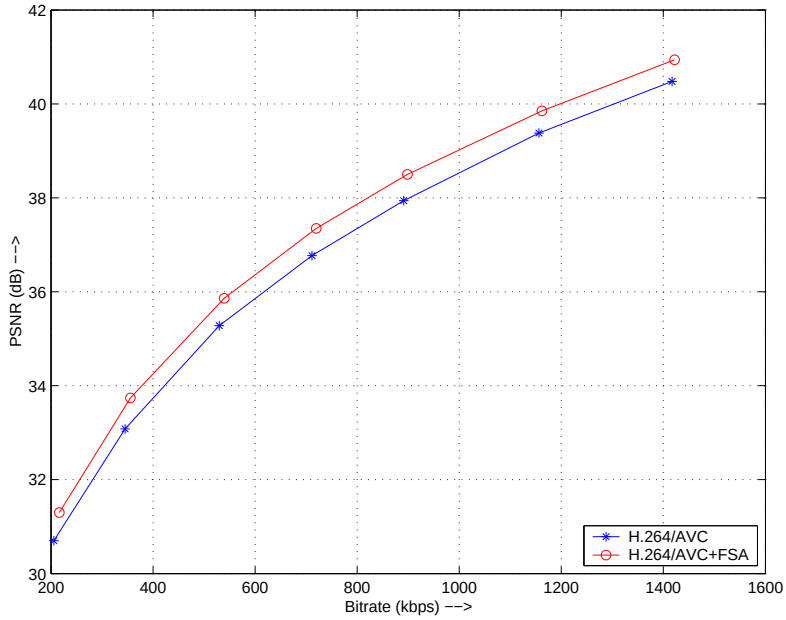

Figure 4.3: *FSA performance for Crew sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations.*

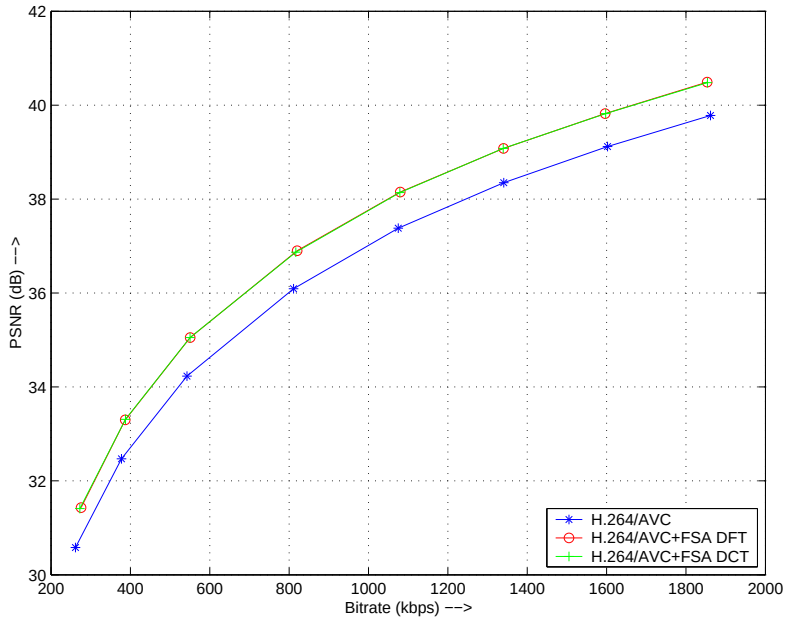

Figure 4.4: *FSA DCT performance for Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations; Green curve H.264/AVC + FSA DCT: Prediction using DCT based FSA 200 iterations.*

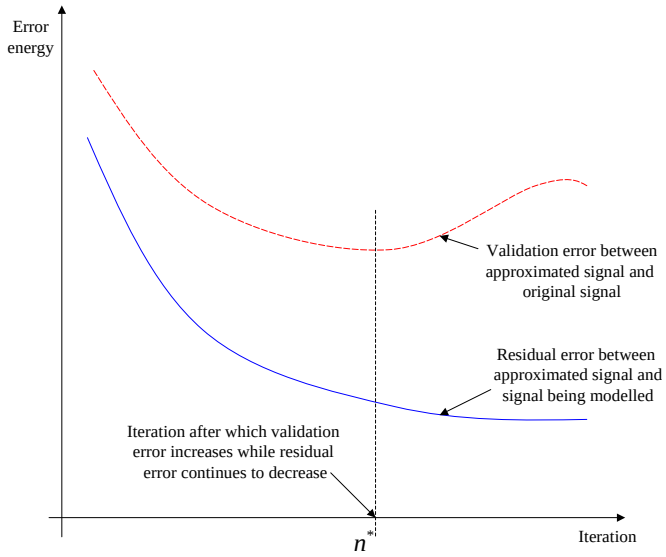

Figure 4.5: *Example of overfit. Blue curve continously decreases because approximation algorithm converges. Red curve (dashed) may increase after a certain point because of overfitting to motion compensated prediction signal.*

of spatial and temporal properties. It is important to note that the stopping rule depends on the data being modelled, the accuracy of motion compensation, spatial relationship between current block and surrounding blocks etc. Since these properties in video data are non-stationary, the stopping has to be adaptively determined for each block.

In statistics, overfitting is a problem that results due to too many modelling parameters. A false model may fit perfectly if the model has enough complexity in comparison to available data.

4.3.1 Oracle Assisted Stopping

To assess the importance of stopping criterion, the following experiment was performed:

- The basic setup as described in Section 4.2 was used with modifications to compute the spatial domain results after every iteration of FSA.
- The resulting data at the motion compensated prediction area was compared

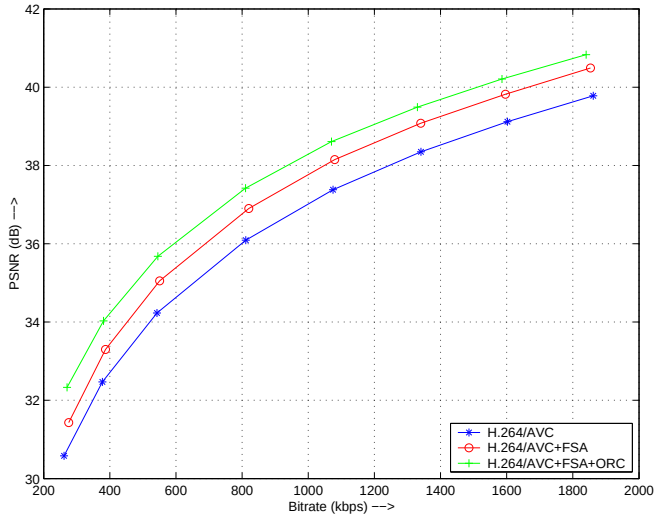

Figure 4.6: *Performance of Oracle assisted FSA for Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations; Green curve H.264/AVC + FSA + ORC: Prediction using Oracle assisted FSA with maximum 200 iterations.*

with the input video data of the original sequence and the mean squared error (MSE) was computed as a function of FSA iteration.

- After completing a fixed maximum number of iterations, the point of minimum MSE was obtained and the spatial data at that iteration was employed as the spatially refined data for further modules of the encoder.

This is an oracle assisted approach because the original data in the region of interest is not available at the decoder side, to compare at every FSA iteration, to decide the stopping. Hence, the encoder has to transmit the stopping iteration number in the bitstream which would be very expensive in terms of overhead bits. For example, in order to directly transmit 256 different stopping states, 8 additional bits per block would be required which would tremendously increase the datarate of the bitstream. In Chapter 5, the number of iterations is reduced to a low value and the overhead of transmitting this information is analyzed with respect to actual gain acheived.

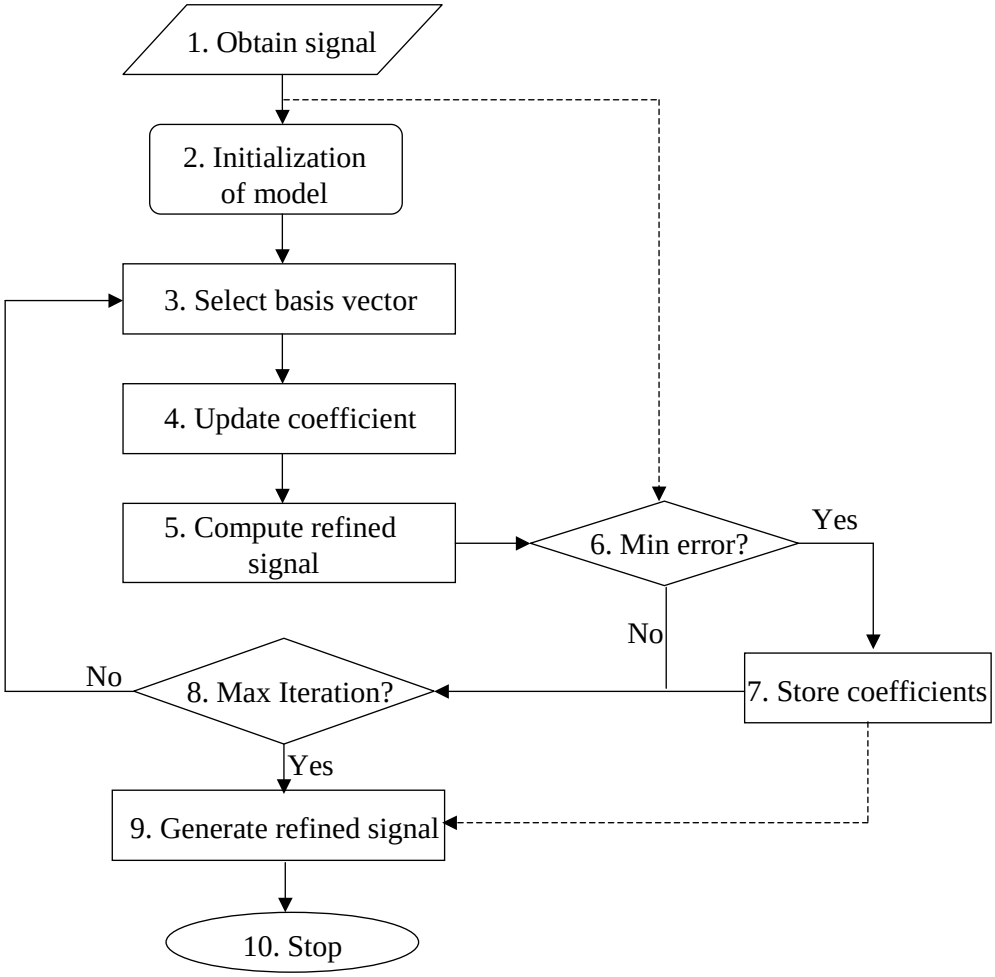

Figure 4.7: *Flowchart of oracle assisted FSA. The steps 5, 6, 7 are additional compared to FSA algorithm. It is oracle assisted because the original signal is unavailable to decoder at step 6, to decide the termination of the iterative algorithm.*

Chapter 5

Prediction Using Best Approximation

Over the last few decades a lot of research has taken place in the area of signal representation and sparse approximation. In approximation theory, it is proved that greedy schemes are well suited for sparse approximation [17]. As seen in Chapter 4, the FSA, which is a greedy scheme, provides a good improvement in PSNR over H.264/AVC. Hence, a logical step would be to evaluate other greedy algorithms for spatial refinement.

According to approximation theorists, algorithms like FSA and Matching Pursuit (MP) fall under the category of *Pure Greedy Scheme*. In 1992, Orthogonal Matching Pursuit (OMP) was proposed [3]. It improves the approximation performance of MP by adding least-squares minimization. A related algorithm was invented by Kaup and Aach for coding segmented images under the name Best Approximation [18, 19]. Such methods are classified as *Orthogonal Greedy Schemes*.

The anatomy of OMP is similar to that of MP but its behavior differs significantly from MP [20]. With this motivation, experiments are conducted on spatial refinement using Best Approximation, which is based on [18]. The original algorithm proposed

in the paper is computationally expensive. Suitable modifications are performed such that computational complexity is significantly reduced.

5.1 Introduction to Best Approximation

The framework of Best Approximation is similar to that of FSA. Consider the signal to be approximated as $f[m,n]$ and modelling using weighted linear combination of basis functions as mentioned in Equation 4.2. Assume that an appropriate basis function $\varphi_{u,v}[m,n]$ is selected according to 4.15. This is the greedy selection step in the algorithm.

In FSA, the residuum is just approximated by the selected basis function. This partial approximation is added to the already present approximation $g^{(\nu)}[m,n]$. The expansion coefficients of already selected basis functions are unaltered in FSA. The Best Approximation improves the performance of FSA by modifying the expansion coefficients of all the already selected basis functions in order to produce the *best* representation using *all* the selected basis functions.

The selected set of basis functions is denoted as:

$$\mathcal{K}^{(\nu+1)} = \mathcal{K}^{(\nu)} \cup (u,v) \quad \text{if } (u,v) \notin \mathcal{K}^{(\nu)}. \tag{5.1}$$

Assume that the expansion coefficients of all the selected basis functions are modified in the iteration $(\nu+1)$. The updated parametric model can be written as

$$g^{(\nu+1)}[m,n] = g^{(\nu)}[m,n] + \sum_{(u,v)\in\mathcal{K}^{(\nu+1)}} \Delta c_{u,v}\varphi_{u,v}[m,n]. \tag{5.2}$$

The residual error in the iteration $(\nu+1)$ is determined by

$$r_w^{(\nu+1)}[m,n] = r_w^{(\nu)}[m,n] - \sum_{(u,v)\in\mathcal{K}^{(\nu+1)}} \Delta c_{u,v}\varphi_{u,v}[m,n]w[m,n]. \tag{5.3}$$

The weighted energy of residual error can be expressed as

$$E^{(\nu+1)} = \sum_{(m,n)\in\mathcal{L}} w[m,n] \left\{ r_w^{(\nu)}[m,n] - \sum_{(u,v)\in\mathcal{K}^{(\nu+1)}} \Delta c_{u,v}\varphi_{u,v}[m,n]w[m,n] \right\}^2 . \quad (5.4)$$

The $\Delta c_{u,v}$ are computed by setting the partial derivative of $E^{(\nu+1)}$ with respect to all $\Delta c_{u,v}$ to zero.

$$\frac{\partial E^{(\nu+1)}}{\partial \Delta c_{u,v}} \stackrel{!}{=} 0 \quad \forall \ (u,v) \in \mathcal{K}^{(\nu+1)}. \quad (5.5)$$

The Equation 5.5 yields a system of linear equations for each coefficcient $\Delta c_{u,v} \quad \forall \ (u,v) \in \mathcal{K}^{(\nu+1)}$

$$\sum_{(k,l)\in\mathcal{K}_(\nu+1)} \Delta c_{k,l} \sum_{(m,n)\in\mathcal{L}} w[m,n]\varphi_{k,l}[m,n]\varphi_{u,v}[m,n] = \sum_{(m,n)\in\mathcal{L}} r_w^{(\nu)}[m,n]\varphi_{u,v}[m,n] \quad . \quad (5.6)$$

The minimization problem can be better analyzed from the perspective of Linear Algebra. The process of calculation of the expansion coefficients of all the selected basis vectors in order to best fit the original vector $\mathbf{f}$ can be considered as the projection of $\mathbf{f}$ onto the subspace spanned by $\{\varphi_{u,v} \quad \forall \ (u,v) \in \mathcal{K}^{(\nu+1)} \}$. The best projection is obtained if the residual in iteration $(\nu+1)$ is orthogonal to each basis vector in $\mathcal{K}^{(\nu+1)}$. Therefore, multiplying the Equation 5.3 by $\varphi_{k,l}[m,n]$ and summing over the entire area $\mathcal{L}$, we get the Equation 5.6 directly.

In OMP algorithm, the expansion coefficients for the selected basis vectors are calculated by solving the projection problem in a least squares sense,

$$\min \sum_{(m,n)} (f[m,n] - \sum_{(k,l)\in\mathcal{K}^{(\nu+1)}} c_{k,l}\varphi_{k,l}[m,n])^2 \quad (5.7)$$

where λ_j represents the index of basis function selected in iteration j.

The Equation 5.6 can be solved uniquely as long as the number of selected basis functions does not exceed the number of degrees of freedom in the signal being modelled.

The linear system of equations can be visualized well in matrix notation. Let the indices of the basis functions which belong to $\mathcal{K}^{(\nu+1)}$ be denoted from 0 to ν

$$\mathcal{K}^{(\nu+1)} = \{(u_0, v_0); (u_1, v_1); \cdots (u_\nu, v_\nu)\} = \{(k_0, l_0); (k_1, l_1); \cdots (k_\nu, l_\nu)\}. \tag{5.8}$$

A vector $\Delta\mathbf{c}$ of size $(\nu + 1) \times 1$ is formed from the update variables as follows

$$\Delta\mathbf{c} = \begin{pmatrix} \Delta c_{k_0,l_0} \\ \Delta c_{k_1,l_1} \\ \vdots \\ \Delta c_{k_\nu,l_\nu} \end{pmatrix} \tag{5.9}$$

The residuals are similarly summarized into another vector $\mathbf{r}_\mathbf{w}^{(\nu)}$

$$\mathbf{r}_\mathbf{w}^{(\nu)} = \begin{pmatrix} \sum_{(m,n)\in\mathcal{L}} r_w^{(\nu)}[m,n]\varphi_{u_0,v_0}[m,n] \\ \sum_{(m,n)\in\mathcal{L}} r_w^{(\nu)}[m,n]\varphi_{u_1,v_1}[m,n] \\ \vdots \\ \sum_{(m,n)\in\mathcal{L}} r_w^{(\nu)}[m,n]\varphi_{u_\nu,v_\nu}[m,n] \end{pmatrix} \tag{5.10}$$

The scalar products of the weighted basis functions are represented in a $(\nu+1)\times(\nu+1)$ matrix

$$\mathbf{W} = \begin{pmatrix} \sum_\mathcal{L} w[m,n]\varphi_{k_0,l_0}[m,n]\varphi_{u_0,l_0}[m,n] & \cdots & \sum_\mathcal{L} w[m,n]\varphi_{k_\nu,l_\nu}[m,n]\varphi_{u_0,l_0}[m,n] \\ \sum_\mathcal{L} w[m,n]\varphi_{k_0,l_0}[m,n]\varphi_{u_1,l_1}[m,n] & \cdots & \sum_\mathcal{L} w[m,n]\varphi_{k_\nu,l_\nu}[m,n]\varphi_{u_1,l_1}[m,n] \\ \vdots & \ddots & \vdots \\ \sum_\mathcal{L} w[m,n]\varphi_{k_0,l_0}[m,n]\varphi_{u_\nu,l_\nu}[m,n] & \cdots & \sum_\mathcal{L} w[m,n]\varphi_{k_\nu,l_\nu}[m,n]\varphi_{u_\nu,l_\nu}[m,n] \end{pmatrix} \tag{5.11}$$

The system of linear equations can now be expressed in matrix notation as

$$\mathbf{W}\Delta\mathbf{c} = \mathbf{r}_\mathbf{w}^{(\nu)}. \tag{5.12}$$

The unknown coefficients can be solved by a matrix inversion

$$\Delta\mathbf{c} = \mathbf{W}^{-1}\mathbf{r}_\mathbf{w}^{(\nu)}. \tag{5.13}$$

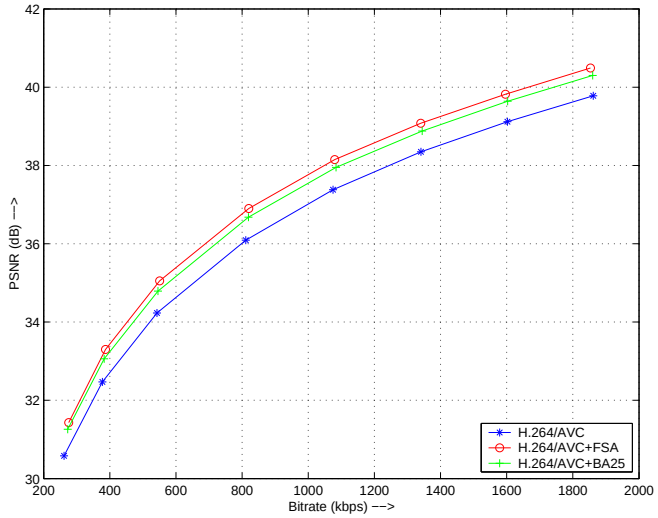

Figure 5.1: *Performance of Best Approximation, Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations; Green curve H.264/AVC + BA25: Prediction using DFT based Best Approximation, 25 iterations.*

The new expansion coefficients are obtained by updating all the selected coefficients $c_{u,v}^{(\nu)}$

$$c_{u,v}^{(\nu+1)} = c_{u,v}^{(\nu)} + \Delta c_{u,v} \quad \forall \ (u,v) \in \mathcal{K}^{(\nu+1)}. \tag{5.14}$$

5.1.1 Best Approximation Using DFT Basis Functions

In this section, the 2D DFT basis functions $\varphi_{k,l}[m,n] = e^{j2\pi/Mmk} e^{j2\pi/Nnl}$ are inserted in the generic equations for Best Approximation derived previously. The parametric model is same as in the case of FSA using DFT bases given in Equation 4.18. Refer [13] for details.

Fig. 5.1 compares the performance of H.264/AVC (blue curve) with FSA (red) & Best Approximation (Green) for Vimto sequence. It is evident that FSA performs better than Best Approximation for the bitrates considered. The difference between FSA & Best Approximation is low at lower bitrates but becomes higher with increasing

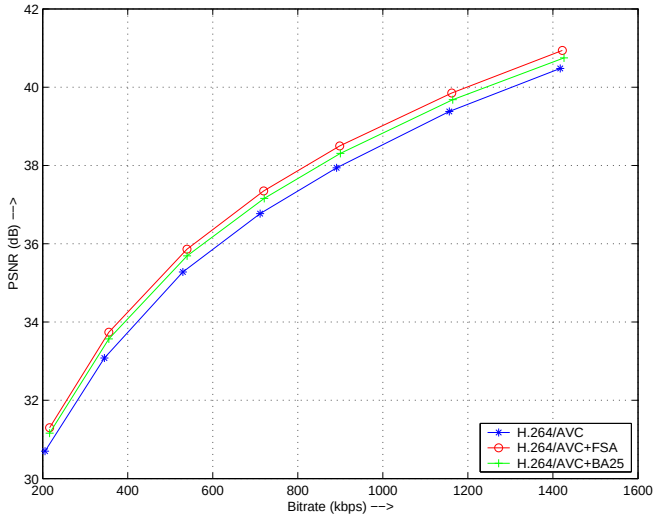

Figure 5.2: *Performance of Best Approximation, Crew sequence. Blue curve H.264/AVC: settings in Section 4.2.1, Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations, Green curve H.264/AVC + BA25: Prediction using DFT based Best Approximation, 25 iterations.*

bitrates. Even though the coefficients of all the selected basis vectors are updated, Best Approximation shows a lesser gain than FSA. The following section analyses the reason for this loss in performance.

Analysis of results

For the purpose of spatial refinement, the algorithm should exhibit not only good approximation properties but also good extrapolation ability for capturing the signal properties from neighboring blocks and extending it to the MC predicted area. Although, Best Approximation yields a good approximation of the entire area being modelled, it performs poorly when the model is used to produce refined samples in the area of interest [1]. It is similar to the better performance of FSA compared to Best Approximation for extrapolation purpose. The extrapolation property of Best Approximation is studied extensively in [13, 1]. The extrapolation may achieve better results compared to FSA in the first few iterations but the performance of FSA becomes better

with increasing number of iterations [13]. Moreover, Best Approximation has higher computational complexity than FSA because of the matrix inversion required in each iteration.

Even though the basis vectors are mutually orthogonal in case of DFT or DCT, there is orthogonality deficiency induced due to the introduction of weighting function. Another reason for the performance of Best Approximation is the non-orthogonality of the basis vectors. The DFT or DCT basis vectors correspond to frequency representation. According to [1], Best Approximation performs similar to FSA when the number of basis functions selected is less than or equal to the number of dominant frequencies present in the signal. But when the number of basis vectors selected exceeds the number of dominant frequencies, the other basis vectors with non-zero projections onto dominant directions start influencing the coefficients of dominant frequencies. The coefficients of these dominant frequencies are altered unduly in order to compensate the components of other basis vectors in this direction.

5.2 Best Approximation with Relaxation

According to approximation theory, the Best Approximation falls under *Orthogonal Greedy Scheme*. Such schemes have a potential to perform better than FSA because expansion coefficients of all the selected basis vectors are updated in each iteration. Nevertheless, because of the problems discussed in Section 5.1.1, it yields poorer refinement performance compared to FSA. In this section methods are devised to counteract the problems of computational complexity and poorer extrapolation property.

Computational complexity

The basic steps in Best Approximation are:

- Selection of basis vectors
- Projection of residual onto the subspace spanned by selected basis vectors

The computationally intensive matrix inversion step is a part of the projection process. The approximation is iterative because the selection of basis vectors is greedy in nature. The basic idea in reducing computational complexity is the fact that if the basis vectors to be selected is known a priori, the projection process has to take place only once. Therefore, it is desirable to estimate the basis vectors to be selected in an efficient way to avoid the matrix inversion multiple times. However, the pure greedy schemes select only one basis vector in each iteration.

Avoiding overfit of model

Another problem in the greedy selection of basis vectors in each iteration of Best Approximation is overfitting of the model. The selection of a basis vector that gives maximum reduction of residual energy tends to tune the model to reproduce preliminarily estimated samples, because expansion coefficient of all the selected basis vectors are updated in each iteration. This gives less benefits for extension of signal properties into the prediction area. This is an important reason for the poorer performance of Best Approximation compared to FSA for spatial refinement. It is necessary to device a scheme to avoid overfitting, in order to improve the overall refinement ability of Best Approximation.

Idea of Relaxation Parameter

In this section, a novel scheme to tackle both problems discussed above is presented. In the proposed approach, a relaxation parameter is introduced in the pure greedy

scheme which provides a significant reduction in computational complexity and at the same time avoids overfitting. The approach is similar to Weak Orthogonal Greedy Algorithms analyzed by Temlykov in [21]. The new scheme selects *all* the basis vectors that provide *at least* a specificed fraction of reduction in error as the best basis vector for a particular iteration.

Let the maximum residual error reduction obtained by searching through all the basis vectors be ΔE_{max}. In Best Approximation, only the basis vector providing this reduction is included into the set of chosen vectors. In the new approach, a relaxation parameter τ is introduced which takes a fractional value between 0 and 1. The selection of basis vectors into the set happens such that all the basis vectors that provide at least $\tau.\Delta E_{max}$ are included into the set of chosen vectors in a particular iteration. The new residual is computed by projecting the current residual onto the subspace spanned by the new set of vectors. This scheme is referred to as Best Approximation with Relaxation (BAR) in further sections of this thesis.

Let $\mathbf{f}$ denote the function to be approximated and $\mathbf{r}_i$ the residual after iteration i. At the initialization $\mathbf{r}_0 = \mathbf{f}$. Then, Best Approximation with relaxation paramter can be inductively defined as:

- Select all basis vectors $\boldsymbol{\varphi}_i$ that satisfy

$$\Delta E_{\boldsymbol{\varphi}_i} \geq \tau_i \max_{\boldsymbol{\varphi} \in \Phi} \Delta E_{\boldsymbol{\varphi}}$$

  where, τ_i is the relaxation factor in iteration i.

- $\mathbf{g}_i = P_{H_i}(\mathbf{f})$, where $H_i = \text{span}(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2.., \boldsymbol{\varphi}_\text{i})$ and P_{H_i} denotes projection onto the subspace spanned by H_i

- $\mathbf{r}_i = \mathbf{f} - \mathbf{g}_i$

The peroformance of this algorithm was evaluated with a fixed relaxation of $\tau = 0.5$. With this relaxation, there is a significant improvement in PSNR compared to Best Approximation.

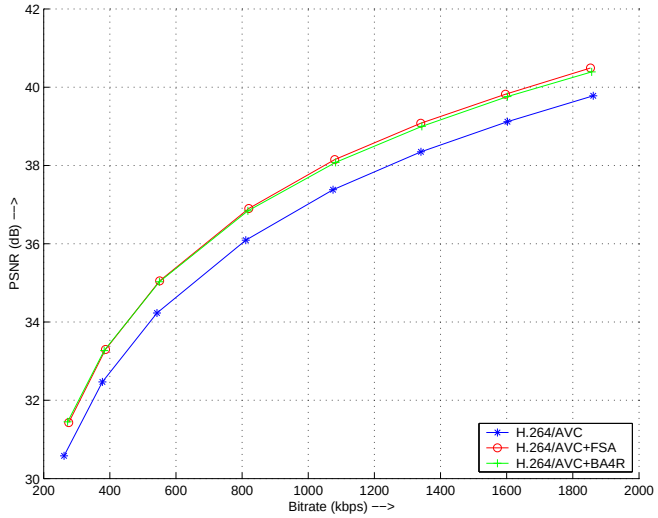

Figure 5.3: *Performance of Best Approximation with Relaxation, Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1, Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations, Green curve H.264/AVC + BA4R: Prediction using Best Approximation with Relaxation=0.5, 4 iterations.*

A variation of BAR, is to directly allow relaxation in terms of number of basis vectors chosen in each iteration, instead of relaxation in residual error reduction. In this scheme, the selection of basis vectors in each iteration can be limited to a specific number, in order to impose an upper limit on the number of chosen basis functions. This means that the computational complexity of the refinement process does not vary across different test cases, which is a desirable feature in video codecs. Such a scheme is denoted as Best Approximation with Simplified Relaxation (BASR).

5.3 Speed Improvements

The performance of FSA based prediction offers significant quality improvements over existing prediction schemes in H.264/AVC. Nevertheless, the computational complexity is considerably higher. The complexity problem is especially important since the algorithm has to be implemented at the decoder side too. Hence, one of the important

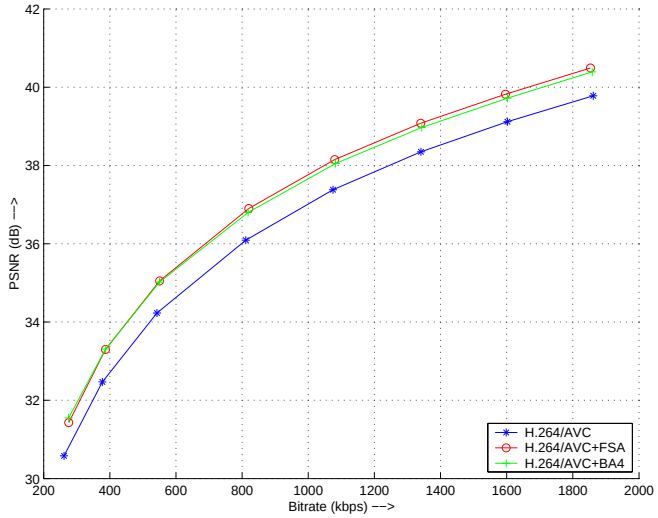

Figure 5.4: *Performance of Best Approximation with Simplified Relaxation, Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1, Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations, Green curve H.264/AVC + BA4: Prediction using Best Approximation with Simplified Relaxation, 4 iterations.*

goals of this thesis was to design a new algorithm that has similar or better performance compared to FSA but with significantly lower complexity.

At the starting of the thesis, the PSNR of reference FSA implementation was recorded for 200 iterations. The result of each iteration introduces dependencies on the next iterations thereby precluding a parallel execution of the algorithm. By moving to Best Approximation, one obvious advantage was the reduction in number of iterations from 200 to 25, but with each iteration being more complex than in the case of FSA. The increased complexity in each iteration is still admittable because it can be parallelized better within each iteration. With the addition of a relaxation factor, the total number of iterations was reduced to as low as 4, but with each iteration having slightly more complexity. Hence, a significant reduction in total number of iterations is achieved with the new algorithm. To evaluate the actual reduction in computational complexity, the MATLAB version of the algorithms were used to record the average time required to predict each macroblock, in Linux workstation running at 2.2GHz. The number of

iterations and computational time required are tabulated below.

Algorithm	Iterations	Avg. Time (sec)	Avg. Speedup factor
FSA	200	1.16	1.0 (Reference)
BA	25	0.265	4.3
BAR	4	0.052	22.3
BASR	4	0.042	27.6

Table 5.1: *Comparison of computational complexity of different algorithms. FSA - Frequency Selective Approximation, BA - Best Approximation, BAR - Best Approximation with Relaxation and BASR - Best Appoximation with Simplified Relaxation. The time taken for FSA is considered as reference to compute the speedup factors of other algorithms.*

From Fig. 5.3 and Tab. 5.1, it can be concluded that for Vimto sequence, Best Approximation with relaxation gives almost the same PSNR gain as FSA, with more than 22 times reduction in computational complexity.

5.4 Design of weighting function

An important advantage of FSA over Matching Pursuit is the possibility of non-uniform weighting of error samples in FSA. For error concealment application, the weight for the area of interest is set to zero as no data is available for the region. However, for the refinement application, valid data in the form of motion compensated prediction is available to the encoder. The relative importance of motion compensated predicted data and the data at adjacent blocks is directly controlled by the weighting function. This section addresses the aspect of design of weighting function for refinement application.

Constant weight for the region to be refined

In the first step of weighting function design, a fixed weighting parameter $w_0 > 0$ is used in the entire area to be refined shown as region-E in Fig. 4.1. Generally the image content becomes less correlated to the adjacent blocks with increasing distance from the region of interest. Such a scenario is best captured by an isotropic model [14] with its center at the midpoint of the region to be refined represented mathematically as

$$w[m,n] = \begin{cases} \hat{\rho}^{\sqrt{(m-\frac{M-1}{2})^2+(n-\frac{N-1}{2})^2}} & ,(m,n) \in A,B,C,D \\ w_0 & ,(m,n) \in E \\ 0 & ,(m,n) \in F,G,H,I \end{cases} \tag{5.15}$$

The algorithm was executed with different values of w_0 and the resulting PSNR were recorded in Fig. B.8 and Fig. B.9.

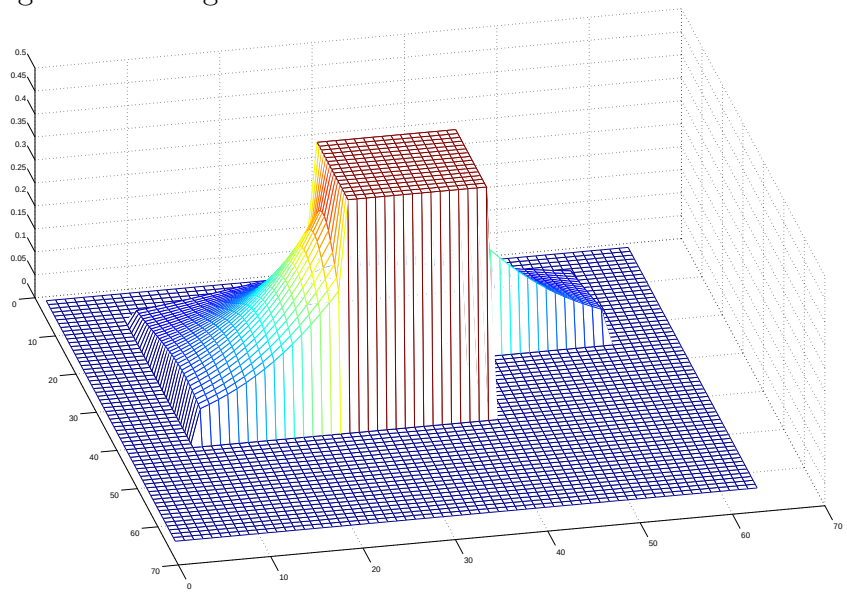

Figure 5.5: *Constant weighting in center block, Isotropic in known neighborhood, zero in unknown area.*

Exponentially decaying weight for the region to be refined

It is known that the process of motion compensation generally performs best at the center of the block considered. The prediction error usually is higher at block boundaries

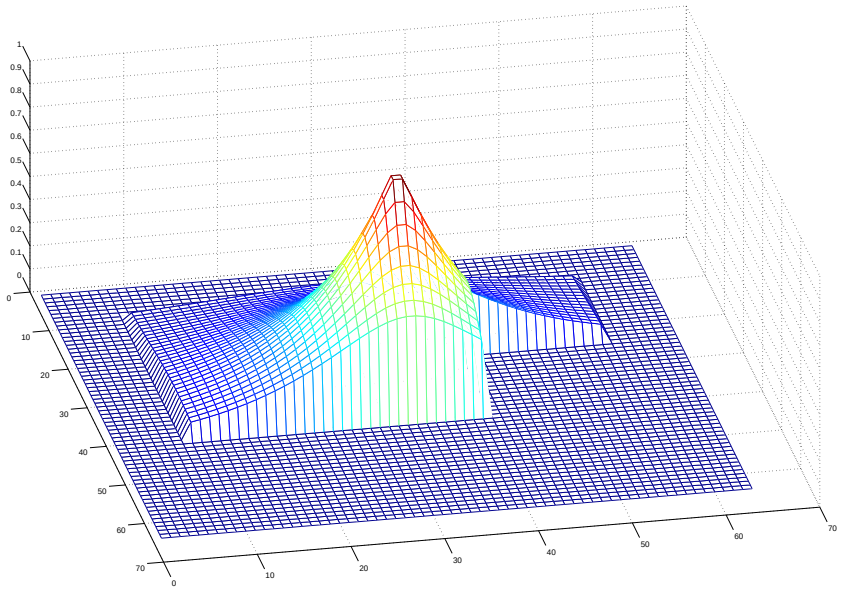

Figure 5.6: *Isotropic weighting in center block and known neighborhood, zero in unknown area.*

compared to the central area. Thus, we need to emphasize less on the approximation error near block boundaries. This can be achieved by extending the isotropic weighting model from the regions A,B,C,D to region-E. Thus, the weighting function becomes:

$$w[m,n] = \begin{cases} \hat{\rho}\sqrt{(m-\frac{M-1}{2})^2+(n-\frac{N-1}{2})^2} & ,(m,n) \in A,B,C,D,E \\ 0 & ,(m,n) \in F,G,H,I \end{cases} \tag{5.16}$$

5.5 Termination criteria

As in the case of FSA, the termination criteria plays an important role in Best Approximation for controlling the closeness of the refined picture to the original video. But unlike FSA, the characteristics of the termination iteration are different in Best Approximation due to two reasons:

- Best Approximation performs well when the number of basis vectors selected is close to the number of dominant frequencies in the signal [1]. This enables us to

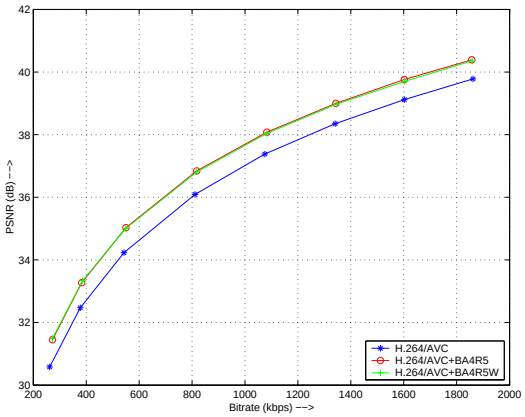

Figure 5.7: *Performance of Best Approximation with Relaxation, Isotropic Weighting, Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + BA4R5: Prediction using Best Approximation, 4 iterations, Relaxation=0.5; Green curve H.264/AVC + BA4R5W: Prediction using Best Approximation, 4 iterations, Relaxation=0.5, Isotropic Weighting.*

make an intelligent conjecture based on the estimate of dominant frequencies in the original signal.

- Since Best Approximation with Relaxation can be implemented in as low as 4 iterations, the optimal stopping iteration can as well be communicated to the decoder.

To analyze the importance of stopping criterion in Best Approximation, as in Section 4.3.1, an oracle assisted approach was employed. Best Approximation with Simplified Relaxation was chosen as the underlying algorithm. The resulting PSNR curve for optimal stopping iteration is depicted in Fig. 5.8. There is an average improvement of 0.3 dB for Vimto sequence.

In the next step, the optimal stopping iteration was communicated to the decoder, thus making the algorithm realizable in practise. Since there are four possible stopping states, a 2-bit side information is necessary to indicate the optimal stopping iteration. As in all the RD plots for spatial refinement, a 1-bit side information is required to

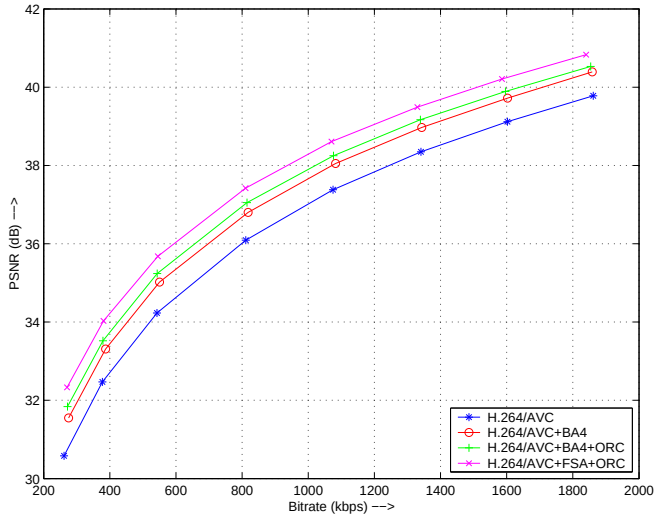

Figure 5.8: *Performance of Oracle assisted Best Approximation with Simplified Relaxation, Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + BA4: Prediction using Best Approximation with Simplified Relaxation, 4 iterations; Green curve H.264/AVC + BA4 + ORC: Prediction using Oracle assisted Best Approximation with Simplified Relaxation, maximum 4 iterations; Magenta Curve H.264/AVC + FSA + ORC: Prediction using Oracle assisted FSA, maximum 200 iterations.*

signal whether spatial refinement is to be performed for a given block. Hence, a direct implementation would require 3-bits of side information. But, the decision to perform spatial refinement can be treated as an additional state in the signalling of stopping iteration, say stop after zero iterations. In this case, we have 5 possible states for a 4 iteration algorithm. We can combine the information for 3 successive blocks to yield $5^3 = 125$ states. This can be communicated well with 7 bits. Therefore, the required side information is around $\frac{7}{3} = 2.33$ bits/macroblock. This signalling overhead is included in the RD curve of oracle assisted Best Appoximation to yield the Fig. 5.9. With much less complexity compared to FSA, the resulting PSNR performance of this approach, is even slightly better than FSA. The calculation of side information is that of a worst-case. Using entropy coding schemes, the side information overhead can be further reduced.

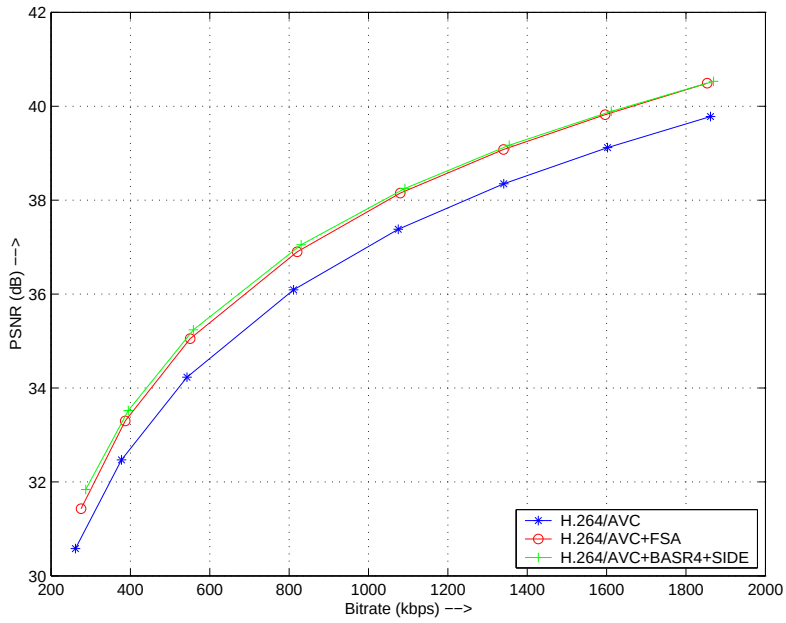

Figure 5.9: *Performance of Best Approximation with Simplified Relaxation and Side Information, Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations; Green curve H.264/AVC + BASR4 + SIDE: Prediction using Best Approximation with Simplified Relaxation and side information, upto 4 iterations, 2.3 bits/block side information.*

Chapter 6

Prediction Using Constrained Weighted Least Squares

In Chapters 4 and 5, it was established that spatial refinement (modelling the motion compensated block together with its neighboring blocks) produces a PSNR gain in resulting video. The approximation of the signal was accomplished by capturing the signal structure in terms of dominant basis functions and producing a sparse representation. Sparsity was imposed implicitly by the greedy selection of basis vectors to be included in approximation.

In this chapter, spatial refinement using approximation algorithms is converted into an optimization problem. The modelling of the signal is accomplished using expansion coefficients as described in Chapters 4 and 5. The squared error between the approximated signal and the original signal is minimized with explicit constraints imposed on the expansion coefficients. The technique of constrained least squares has been successfully applied to removing blocking artifacts in video [22].

The formulation of spatial refinement problem in terms of least squares approximation is shown in Section 6.1. The cost function is improved by adding a sparsity inducing constraint in Section 6.2. The optimization of cost function is performed using

Gradient Descent method in Section 6.2.1. Finally, in Section 6.3, the performance of the Constrained Weighted Least Squares (CWLS) method is evaluated for spatial refinement application.

6.1 Formulation of Weighted Least Squares Approximation

Let $f[m,n]$ denote the intensities of the samples of the entire area $\mathcal{L}$ shown as blocks A to I in Fig. 4.1. Let $c_{k,l}$ be the expansion coefficients used for modelling, then the weighted error criterion

$$J_1 = \frac{1}{2} \sum_{(m,n)\in\mathcal{L}} w[m,n] \left(f[m,n] - \sum_{k,l} \varphi_{k,l}[m,n] c_{k,l} \right)^2 \tag{6.1}$$

has to be minimized, where (k,l) includes all N^2 the basis functions.

In vector notation, consider $f[m,n]$ and $c_{k,l}$ as vectors in N^2 dimensional vector space denoted as vectors $\mathbf{f}$ and $\mathbf{c}$. Then, the basis functions become column vectors in a matrix $\mathbf{\Phi}$. If the weights are placed along the diagonal of a matrix $\mathbf{W}$, Equation 6.1 can be re-written in vector notation as

$$J_1 = \frac{1}{2}(\mathbf{f} - \mathbf{\Phi c})^T \mathbf{W} (\mathbf{f} - \mathbf{\Phi c}) \tag{6.2}$$

Assuming $\mathbf{\Phi}$ to be invertible and $\mathbf{W}$ to be a positive-definite matrix, it can be easily seen that the minima of Equation 6.2 occurs at $\mathbf{c}_{opt} = \mathbf{\Phi}^{-1}\mathbf{f}$ independent of weighting function. The convergence to $\mathbf{c}_{opt}$ would resynthesize the original signal $\mathbf{f}$, which is not the intent of spatial refinement step. In order to capture the inherent structure in the neighboring blocks and extend it to motion compensated block, some constraint has to be imposed on the expansion coefficients to make it suitable for refinement application.

6.2 Selection Of Constraints

One possible approach is to impose smoothness requirement and synthesize a homogeneous signal in the region to be predicted. In such a case, the details in the original signal could get undesirably blurred. As seen in Chapters 4 and 5, sparsity is a suitable constraint to capture inherent structure in signals. The good performance of FSA and Best Approximation reiterates this fact. The ideal measure of sparsity is the L_0 pseudo-norm which counts the number of non-zero values in the parameter vector. But this would involve numerical optimization of non-convex functional, making the process computationally complex. The convex L_1 norm was proposed as an alternative in [12]. The solution to this would involve a non-quadratic optimization and result in a high computational complexity.

In experiments on Mammalian Visual System, Olshausen and Field showed that some functions like $-\sum_i e^{-x_i^2}$ and $\sum_i \log(1+x_i^2)$ have sparsifying abilities [23]. The reason behind the choice of these functions is that, among states with equal variance, these functions favor the ones that have fewest number of non-zero coefficients.

Choosing $\sum_i \log(1+x_i^2)$ as constraint, we obtain the cost due to the new term as

$$J_2 = \frac{1}{2}\sum_{k,l} \log\left(1+\left(\frac{c_{k,l}}{\sigma}\right)^2\right) \tag{6.3}$$

where, σ is a normalizing constant.

Let λ control the relative importance between reconstruction error (J_1) and sparsity (J_2). Then, the overall cost function can be written as

$$J = J_1 + \lambda J_2 = \frac{1}{2}\sum_{(m,n)\in\mathcal{L}} w[m,n]\left(f[m,n]-\sum_{k,l}\varphi_{k,l}[m,n]c_{k,l}\right)^2 + \frac{1}{2}\lambda\sum_{k,l}\log\left(1+\left(\frac{c_{k,l}}{\sigma}\right)^2\right). \tag{6.4}$$

The spatial refinement task now becomes a minimization of the CWLS objective function J. The new expansion coefficients are calculated as a solution to this problem.

The region of interest is generated by using the optimized expansion coefficients in the synthesis equation.

A direct solution to this problem would compute the gradient of J with respect to $c_{u,v}$ and set it to zero to evaluate the optimal $c_{u,v}$. However, it results in N^2 cubic equations (because of the presence of derivative of log term) with N^2 unknowns. Instead of solving this large system of non-linear equations, an iterative cost minimization approach is followed in the next section.

6.2.1 Optimization Using Gradient Descent Method

The Gradient Descent (GD) is an optimization algorithm to find the minima of a function. It is also known as 'steepest descent' or the 'method of steepest descent'. GD is a popular iterative method for solving large system of equations. It computes the gradient of cost function to determine the update of coefficients. We note that gradient is a vector that, for a given $c_{u,v}$, points in the direction of greatest increase of the objective function J. Therefore, J decreases fastest if one goes from $c_{u,v}$ in the direction of the negative gradient of J with respect to $c_{u,v}$ as shown in Fig. 6.1. The update of coefficients is performed using a step size μ in the direction of negative gradient. If the objective function is convex, for proper selection of μ, it is proved that GD converges to a minima. For the current application, although the cost term (J_2) for small values of $c_{u,v}$ is not strictly convex, it can be assumed to be convex around the region of initialization, especially when added with a strictly convex error term (J_1).

Let the gradient contribution due to the cost of reconstruction error be denoted as g_1 and that due to the cost of constraint as g_2. Then, we have

$$g_{1,u,v} = \frac{\partial J_1}{\partial c_{u,v}} = \sum_{k,l} c_{k,l} \sum_{m,n} w[m,n]\varphi_{k,l}[m,n]\varphi_{u,v}[m,n] - \sum_{m,n} w[m,n]f[m,n]\varphi_{u,v}[m,n] \tag{6.5}$$

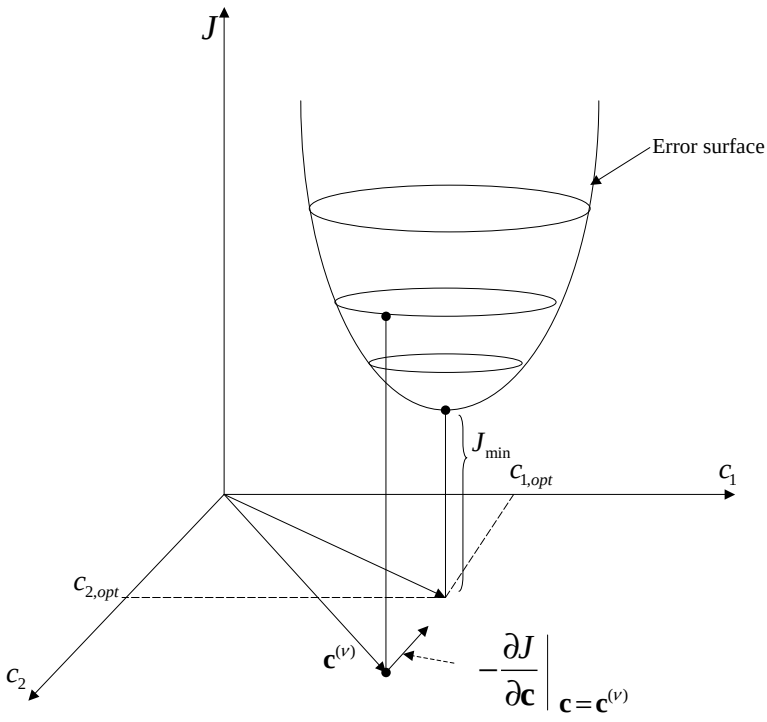

Figure 6.1: *Error Surface and Negative Gradient. The negative gradient at* $\mathbf{c}$ *points towards the direction at which there is a maximum decrement of error.*

and

$$g_{2,u,v} = \frac{\partial J_2}{\partial c_{u,v}} = \frac{c_{u,v}}{\sigma^2 + c_{u,v}^2}. \tag{6.6}$$

The overall gradient with respect to $c_{u,v}$ can be written as

$$g_{u,v} = g_{1,u,v} + \lambda g_{2,u,v}. \tag{6.7}$$

If we choose a step size $\mu^{(\nu)}$ in the direction of negative gradient, the updation of coefficient vector can be expressed as

$$c_{u,v}^{(\nu+1)} = c_{u,v}^{(\nu)} + \Delta c_{u,v}^{(\nu)} \tag{6.8}$$

where, $c_{u,v}^{(\nu)}$ is the value of $c_{u,v}$ in iteration ν and $\Delta c_{u,v}^{(\nu)} = -\mu^{(\nu)} g_{u,v}$. In each iteration, the updation happens over all the coefficients.

The next question to be addressed is the setting of the step size $\mu^{(\nu)}$. Generally, it is solved by employing a line search procedure in the direction of negative gradient and

choosing the step size that leads to a maximum decrement in J. However, for Equation 6.7, this leads to complicated mathematical expressions due to the presence of non-linear terms. Hence, the step size is fixed to a positive constant for implementing the GD algorithm.

6.3 Performance Evaluation Of CWLS

The CWLS approach was evaluated on data formed by the block to be refined and its three neighboring blocks, constituting 32×32 image samples. The 1D-vectorized form of this data is treated as a vector $\mathbf{f}$. The modelling is accomplished using DCT type 4 basis images of size 32×32.

$$\varphi_{u,v}[m,n] = \frac{2}{N}\cos(\frac{\pi}{N}(m+0.5)(u+0.5))\cos(\frac{\pi}{N}(n+0.5)(v+0.5)) \tag{6.9}$$

The 1D-vectorized form of the bases are placed in the columns of $\mathbf{\Phi}$ to form a 1024×1024 matrix.

A weighting similar to FSA algorithm was employed. The weight of the block to be refined was fixed to a constant value of 0.5 and the weights at the neighboring blocks were evaluated using an isotropically decaying model as described in Equation 5.15.

For GD algorithm, the coefficient vector $\mathbf{c}$ has to be given an initial value from which the search can start. An intuitive initial value for $\mathbf{c}$ is the DCT of $\mathbf{f}$ since $\mathbf{\Phi}$ is consitituted by DCT bases. Hence, at the start of optimization, the cost due to reconstruction error J_1 is zero but the cost of constraint J_2 is generally dominant. The optimization proceeds such that the overall cost function J is minimized in accordance with the chosen relative importance of the contributing cost functions. The values of λ between 8 and 16 were found useful for spatial refinement.

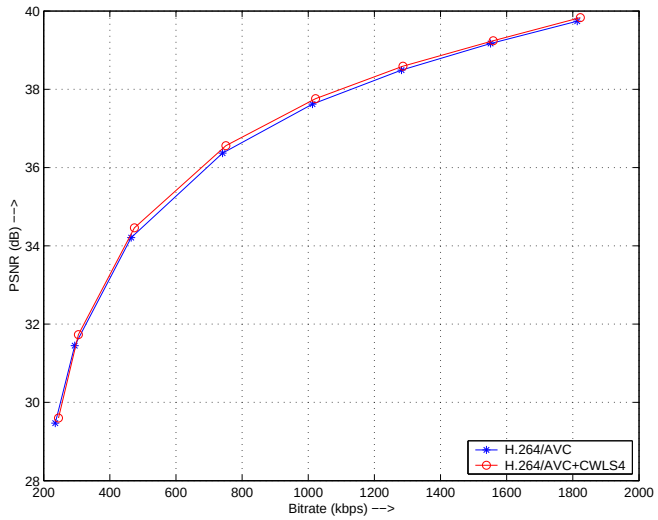

Figure 6.2: *CWLS performance for 25 frames of Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + CWLS4: Prediction using CWLS 4 iterations.*

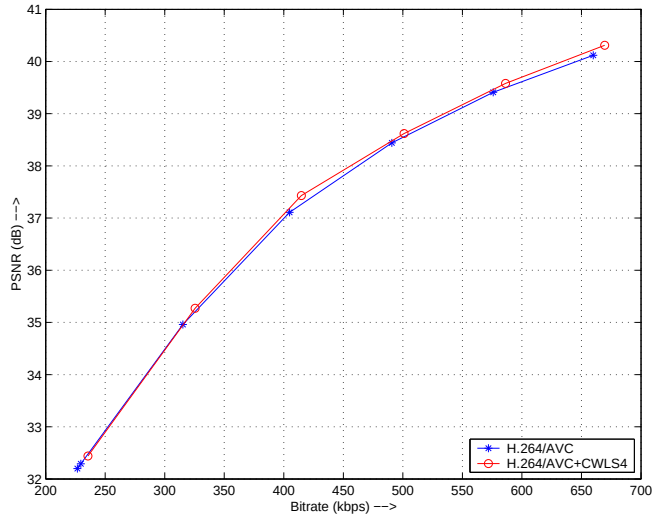

Figure 6.3: *CWLS performance for 25 frames of Discovery city sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + CWLS4: Prediction using CWLS 4 iterations.*

6.4 Analysis Of Results

The RD curves with CWLS based refinement for *Vimto* and *Discovery city* sequences are depicted in Fig. 6.2 and Fig. 6.3 respectively. As can be observed, the gain is lesser than FSA and Best Approximation algorithms. The performance of CWLS is significantly dependent on the ability of constraint term in sparsifying the representation.

In the experiments for analyzing constraints suitable for producing sparse representations by Olshausen and Field [23], the basis functions are also modified iteratively in order to get the signal energy concentrated in a few expansion coefficients. But such an adaptive modification of basis functions would lead to very high computational complexity. CWLS is designed with fixed basis functions and hence produces lesser gain. But the basic idea of imposing constraints to coefficients while minimizing reconstruction error can be explored further. For instance, other measures of sparsity like 'Kurtosis' are proposed in [24].

Chapter 7

Prediction Using Projections Onto Convex Sets

In Chapters 4 and 5, spatial refinement based on greedy approximation was formulated. Later, in Chapter 6, spatial refinement was framed as an approximation task with constraints imposed on expansion coefficients. In this chapter, spatial refinement is viewed as a projection problem where some common properties of video signals are exploited to produce refinement over preliminarily estimated data through motion compensation.

After motion compensation, the amount of improvement in prediction that can be achieved, depends on the *a priori* information that is known about video signals. For example, when some properties about spatial correlation are available, they can be used to enhance the approximation of region of interest. The use of these properties can be considered as *a priori* constraints. These constraints can be used by the encoder and decoder to implement algorithms to spatially refine the motion compensated data.

Specifically, the idea of spatial refinement of preliminarily estimated data can be viewed as employing two different information on the data to be refined, namely

- Temporal information
- Spatial information

Temporal information is inherently used because the initial estimation of data to be filled in current block is done through motion compensation. Spatial information can be utilized to modify the motion compensated prediction data to fit smoothly with the samples in the surrounding blocks. Using this idea, the spatial refinement algorithm can be redefined as the process of converging to a point where both these properties are satisfied to a certain extent.

Projections Onto Convex Sets (POCS) would be ideally suited for this purpose. It has been applied successfully for error concealment application in [25]. In Section 7.1 of this chapter, the theory of POCS is elaborated. Later, in Section 7.2 the spatial and temporal requirements are formulated into convex constraints so that it can be applied to POCS. The performance of POCS is evaluated in Section 7.3. Finally, suitability of POCS for the purpose of spatial refinement is analyzed.

7.1 Description of POCS method

Before POCS algorithm is introduced, the concept of a convex set needs to be elucidated. In Euclidean space, an object is convex if for every pair of points within the object, every point on the straight line segment that joins them is also within the object. Therefore, if C is a convex set, then for any $u_1, u_2, ..., u_N$ in C, and any non-negative numbers $\lambda_1, \lambda_2, ..., \lambda_N$ such that $\lambda_1 + \lambda_2 + ... + \lambda_N = 1$, the vector $\sum_{k=1}^{N} \lambda_k u_k$ is in C. A vector of this type is known as a convex combination of $u_1, u_2, ..., u_N$.

POCS is an algorithm for computing the point of intersection of some convex sets, using a sequence of projections onto those sets. The method is especially useful when

an analytical formula exists for carrying out the projections.

Suppose A and B are convex sets in $\mathcal{R}^N$ and $\mathbf{P}_A$ and $\mathbf{P}_B$ denote projection on A and B, respectively. The algorithm starts with an arbitrary $\mathbf{x}_0 \in A$, and then alternately projects onto A and B:

$$\mathbf{y}_k = \mathbf{P}_B \mathbf{x}_k, \quad \mathbf{x}_{k+1} = \mathbf{P}_A \mathbf{y}_k, \quad k = 0, 1, 2, ... \tag{7.1}$$

This generates a sequence of points $\mathbf{x}_k \in A$ and $\mathbf{y}_k \in B$.

It is proved by Cheney and Goldstein [26] that, if $A \cap B \neq 0$, then the sequences $\mathbf{x}_k$ and $\mathbf{y}_k$ both converge to a point $\mathbf{x}^* \in A \cap B$. The number of iteration before it actually converges could be infinitely many but the sequences $\mathbf{x}_k$ and $\mathbf{y}_k$ satisfy $d(\mathbf{x}_k, B) \to 0$, and $d(\mathbf{y}_k, A) \to 0$, where d calculates the distance of the point from the set. POCS is also useful when the sets do not intersect. In this case POCS yields a pair of points in A and B that have minimum distance [26].

If the number of convex sets k is greater than two, then we can find a point of intersection of the sets by projecting onto A_1, then $A_2, \cdots$, then A_k and then repeating the cycle of k projections.

7.2 Formulation of constraints

POCS has been applied to various image restoration, extrapolation problems where some *a priori* information is available. The goal of this section is to formulate convex constraints using some typical properties of video sequences, so that POCS can be applied to spatial refinement.

Some video characteristics that are employed are:

- Smoothness
- Sparsity of transform coefficients

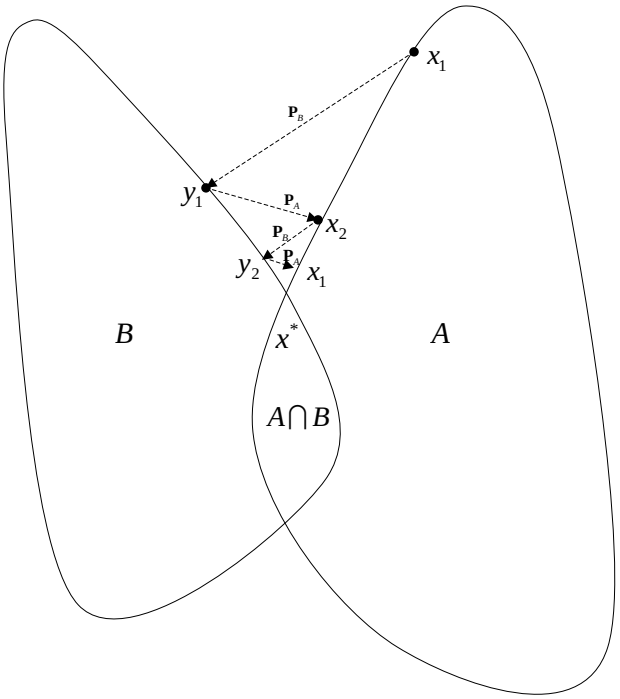

Figure 7.1: *Signal Space Illustration of POCS with Intersecting Sets. When the sets are intersecting convergence happens at a point in the region of intersection of both sets.*

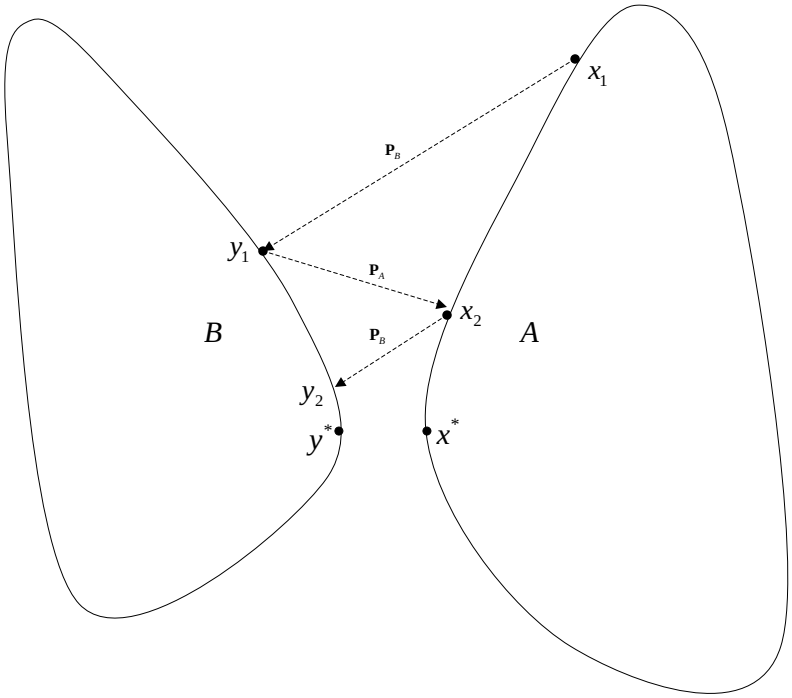

Figure 7.2: *Signal Space Illustration of POCS with Non-intersecting Sets. When the sets are non-intersecting, it yields a pair of points in A and B that have minimum distance.*

- Consistency with known values

Common constraints such as space-limiting, band-limiting, nonnegativity, and bounded energy are proved to be convex sets [27]. The following constraints and projection operators can be designed to characterize the properties mentioned above:

- Spatial domain constraint: We define a class of signals C_1 that takes on a prescribed set of known values. Consider N-dimensional vectors $\mathbf{x}$ with some components equal to known values. This can be expressed as

$$C_1 = \{\mathbf{x} \in \mathcal{R}^N; x_i = k_i, i \in I\} \tag{7.2}$$

  where, x_i is the i-th component of vector $\mathbf{x}$ and k_i are the known values in index set I. An operator $\mathbf{P}_1$ that projects onto this convex set can be stated as:

$$[\mathbf{P}_1\mathbf{x}]_i = \begin{cases} k_i & , i \in I \\ x_i & , \text{otherwise} \end{cases} \tag{7.3}$$

  where, $[\mathbf{P}_1\mathbf{x}]_i$ is the i-th projection coefficient. Therefore, the projection $\mathbf{P}_1$ forces the data at specific locations to assume known values.

- Frequency domain constraint: We define a class of signals C_2 that takes on a prescribed set of transform coefficients. This set encompasses all signals $\mathbf{y}$ in the N-dimensional complex space $\mathcal{C}^N$ with some transform coefficients equal to known values. This class can be mathematically described as:

$$C_2 = \{\mathbf{y} \in \mathcal{C}^N; [\mathbf{T}\mathbf{y}]_i = z_i, i \in I\} \tag{7.4}$$

  where, $\mathbf{T}$ is a linear operator that transforms the signal $\mathbf{y}$ into frequency domain, $[\mathbf{T}\mathbf{y}]_i$ is the i-th transform coefficient, z_i are the known values and I is the index set of known values. We can now define the projection operator onto this convex set C_2 as:

$$[\mathbf{T}\mathbf{P}_2\mathbf{y}]_i = \begin{cases} z_i & , i \in I \\ [\mathbf{T}\mathbf{y}]_i & , \text{otherwise} \end{cases} \tag{7.5}$$

  Therefore, the projection $\mathbf{P}_2$ forces the transform coefficients at specific locations to assume known values.

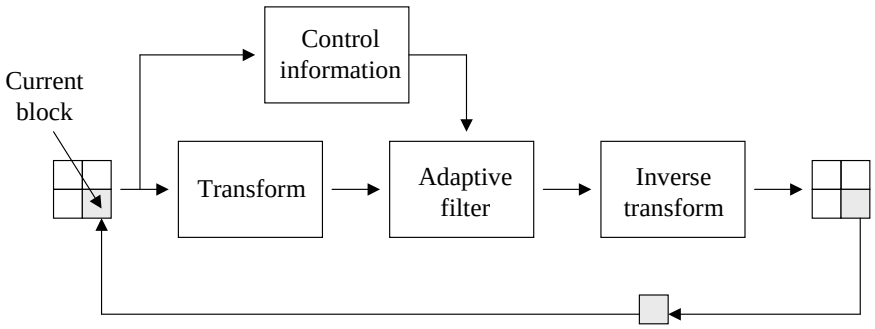

Figure 7.3: *Schematic of POCS Applied to Video.*

7.3 Performance evaluation of POCS

The POCS approach was evaluated on data formed by the block to be refined and its three neghboring blocks, constituting 32×32 image samples. DCT was used to transform this data into frequency domain.

For the projection onto a subspace in the frequency domain, a mask was generated with indices corresponding to frequencies of weak amplitudes as 0 and of other frequencies as 1. To capture the dynamics of different signals, the identification of weak amplitudes was designed based on the mean of all amplitudes. Let μ be the mean of coefficients in DCT domain. All the frequencies that have amplitude less than $\rho.\mu$ were selected as weak, where ρ is a fraction between 0 and 1.

The set of all signals that have an amplitude of zero at masked frequencies is designated as C_1. This set uses the properties of smoothness and sparsity in the frequency domain. In spatial domain, the set of all signals that take on known reconstructed values at neighboring blocks is designated as C_2.

The projection onto C_1 was performed by setting the coefficients of the masked frequencies to zero. After the projection in frequency domain, the new spatial domain samples were calculated. In the spatial domain, the projection onto C_2 is carried out by replacing the new spatial domain samples with the known values at the neighboring blocks.

This procedure is performed iteratively upto a pre-defined number of iterations. Finally, new samples in the region of interest are taken from the resulting spatial domain data as the spatially refined samples. The PSNR performance of POCS is recorded in Fig. 7.4 and Fig. 7.5.

It can be noticed that the PSNR improvement due to POCS is much lesser compared to greedy algorithms. This behavior can be attributed to the fact that thresholding of frequency coefficients removes energy from the signal. Ideally, this reduction has to be compensated by the modification of coefficients of non-masked frequencies. Although POCS satisfies spatial constraints, it retains the coefficients of non-masked frequencies without any compensation for masked coefficients. Hence, this results in a signal that could be missing some image details and produces lesser gain compared to greedy schemes.

The performance of POCS could be improved by modifying the projection operator in frequency domain. Instead of just thresholding the weak frequencies to zero, we could best approximate the signal in the subspace spanned by the non-masked frequencies in each iteration of POCS. Hence, each iteration in this approach would be similar to Best Approximation scheme. Nevertheless, each iteration would have similar computational complexity compared to Best Approximation, resulting in much higher overall complexity.

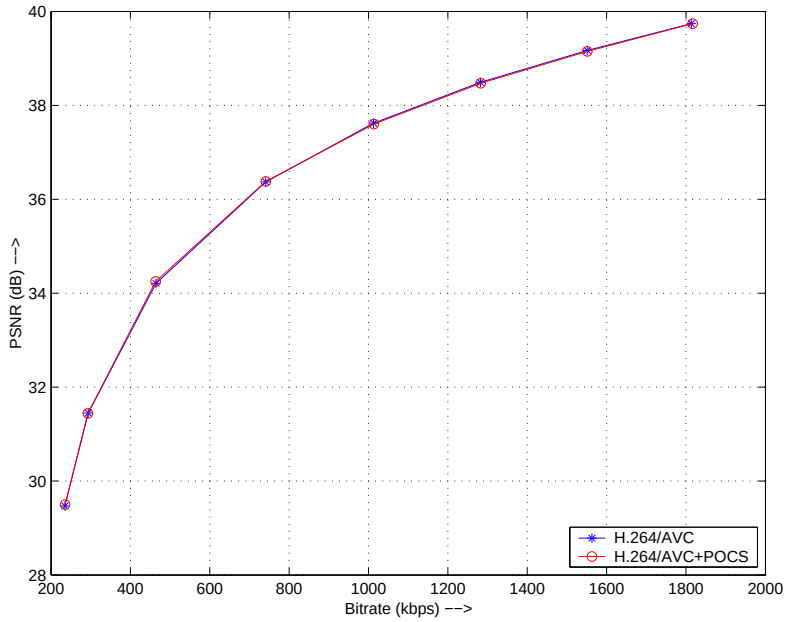

Figure 7.4: *POCS performance for Vimto sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + POCS: Prediction using POCS 10 iterations.*

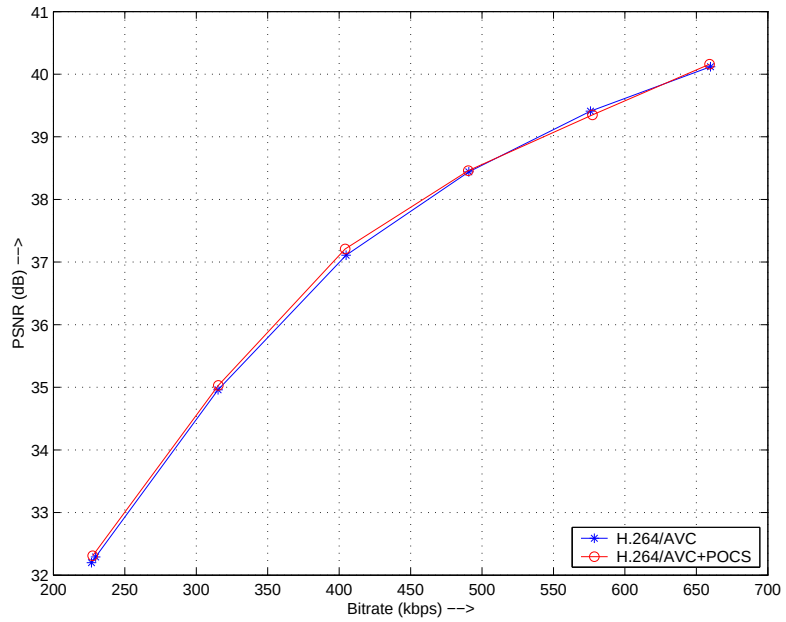

Figure 7.5: *POCS performance for Discovery city sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + POCS: Prediction using POCS 10 iterations.*

Chapter 8

Summary and Future Work

Among the different modes of communication, humans have a special place for visual communication. Emergence of various application scenarios have been fuelling the quest for better video compression algorithms. The challenges in video compression have grown with the convergence of multimedia and communication. Video-conferencing, Digital broadcasting, Internet streaming are typical examples of such convergence applications. Apart from compression efficiency, the challenges in modern video coders are computational complexity, error-robustness, memory efficiency, etc.

The main focus of this thesis was on improving the compression efficiency through better prediction, while keeping the computational complexity under control. Prediction schemes in existing video coders utilize either only temporal or only spatial information along with RD optimization. There have been attempts at combining spatial and temporal information in one-step, thereby enabling a spatio-temporal prediction. The complexity of such algorithms are high because of the high volume to 3D data to be handled.

At the start of this thesis, a novel idea for realizing spatio-temporal prediction by spatially refining the motion compensated prediction block was implemented using FSA algorithm. The results were encouraging, as significant improvements were observed

for some test-cases. But, computational complexity of the iterative FSA algorithm was high. In this thesis, different algorithms for spatial refinement of motion compensated prediction data have been explored.

The extension of FSA to Best Approximation initially showed a performance decrement (Sec. 5.1.1). An improvement to Best Approximation, in terms of relaxation in the selection of basis functions, provides significant reduction in computational complexity and increases quality (Sec. 5.2). The importance of terminating the iterative algorithm was analyzed and experimentally proved (Sec. 5.5). The gain due to optimal stopping was considerably high both for FSA and Best Approximation. Further research can focus on designing optimal stopping of Best Approximation. One key idea is to recognize the fact that Best Approximation is known to perform well as long as the number of selected basis functions is in the range of the number of dominant frequencies in the signal [13]. Hence, a means of identifying dominant frequencies would be useful for stopping the iterative approximation. For instance, the number of dominant frequencies can be identified using data from previous frame motion compensated block and its neighbors.

In order to design a non-interative algorithm for spatial refinement, the approximation problem was reformulated as an error minimization problem but with some additional conditions on model parameters, and a direct solution was attempted. Sparsity was found to be an effective strategy from Chapter 4 and 5. Hence a measure of sparsity based on log function was introduced in the optimization problem as a constraint. But it resulted in a large system of non-linear equations, again forcing an iterative solution. Gradient Descent was used to solve the minimization problem. The performance of constrained optimization depends directly on the ability of constraint to capture important structure in the signal. To improve this algorithm, other measures of sparsity can be experimented. For example, in [24], a measure of sparsity based on Kurtosis is proposed.

Finally, spatial refinement is framed as a projection problem known as POCS. It is based on the fact that common properties of spatial and frequency domain can be defined well in terms of convex sets. The signal to be approximated is iteratively projected onto these sets to satisfy these properties to a certain extent. The projection employed in current implementation is a simple thresholding of small coefficients. It could be improved by using Best Approximation within each iteration of POCS. Moreover, the generation of frequency domain mask significantly affects the final quality. Therefore, methods can be explored for improved generation of masks.

From the experiments in this thesis, it can be concluded that, prediction can be improved by using spatial properties from neighboring blocks after motion compensation and hence a higher compression efficiency for video data can be achieved.

Appendix A

Abbreviations

AVC	Advanced Video Codec
BA	Best Approximation
BAR	Best Approximation with Relaxation
BASR	Best Approximation with Simplified Relaxation
CWLS	Constrained Weighted Least Squares
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
FSA	Frequency Selective Approximation
GD	Gradient Descent
MB	Macroblock
MC	Motion Compensation
MP	Matching Puruit
OMP	Orthogonal Matching Pursuit
POCS	Projections Onto Convex Sets
PSNR	Peak Signal to Noise Ratio
RD	Rate-Distortion

Appendix B

Additional PSNR plots

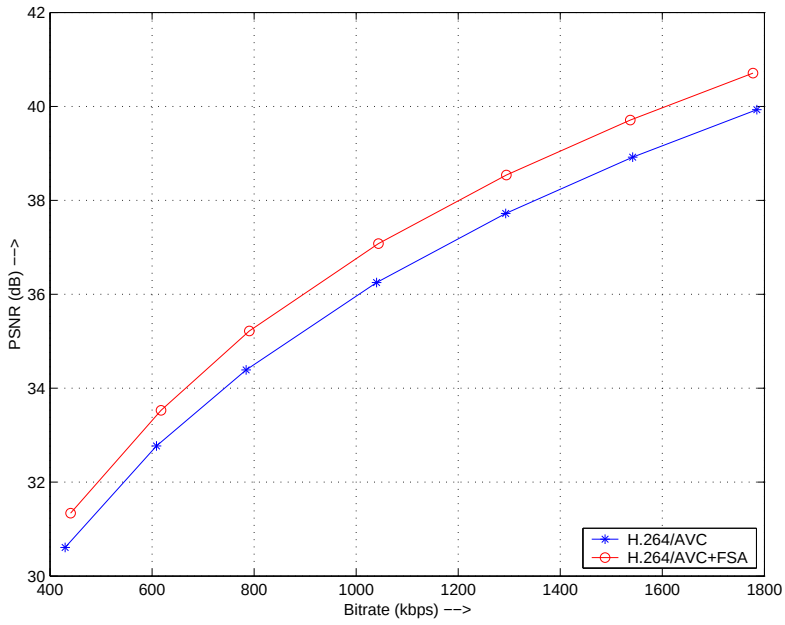

Figure B.1: *FSA performance for Discovery city sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations.*

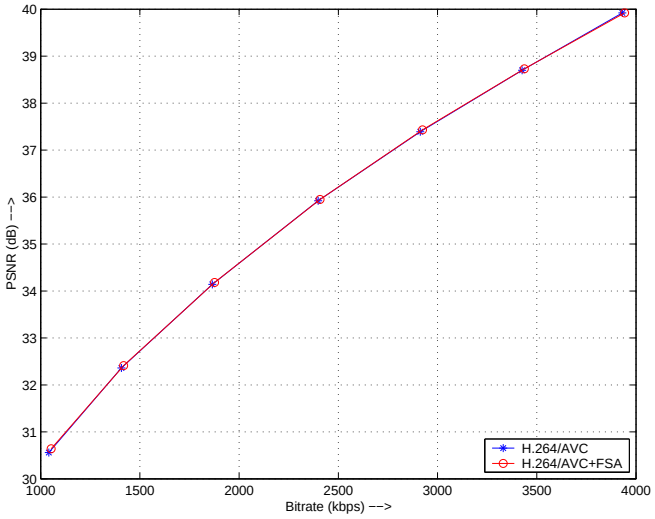

Figure B.2: *FSA performance, Flower Garden sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations.*

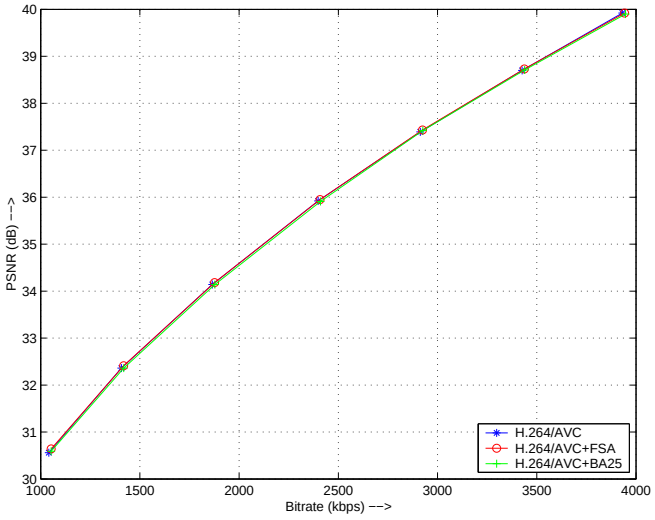

Figure B.3: *Performance of Best Approximation, Flower Garden sequence. Blue curve H.264/AVC: settings in Section 4.2.1; Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations; Green curve H.264/AVC + BA25: Prediction using DFT based Best Approximation, 25 iterations.*

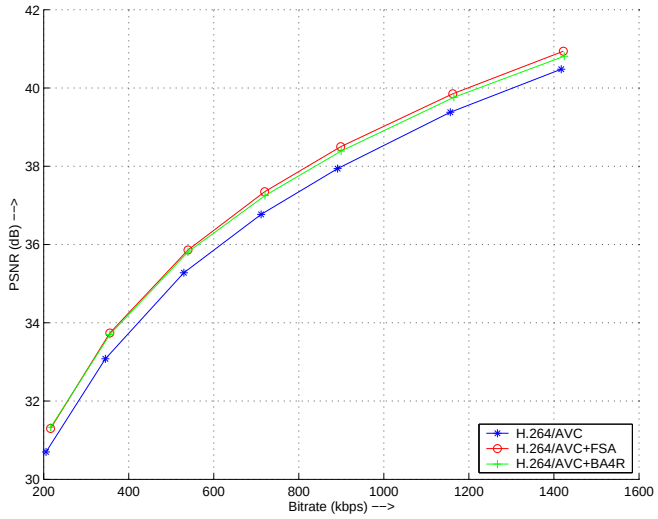

Figure B.4: *Performance of Best Approximation with Relaxation, Crew sequence. Blue curve H.264/AVC: settings in Section 4.2.1, Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations, Green curve H.264/AVC + BA4R: Prediction using Best Approximation with Relaxation=0.5, 4 iterations.*

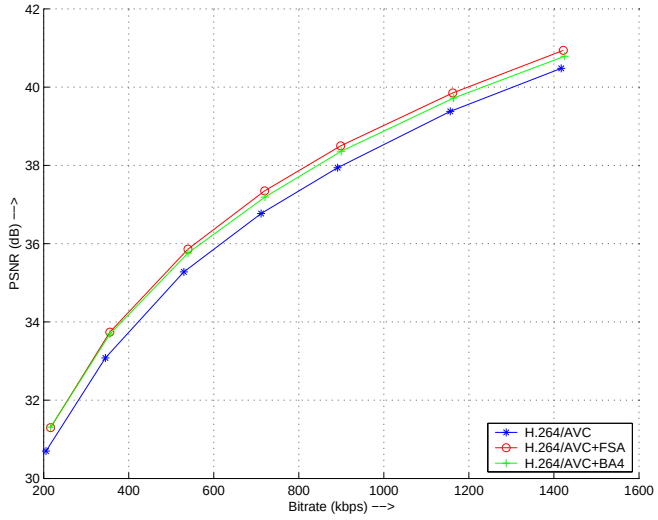

Figure B.5: *Performance of Best Approximation with Simplified Relaxation, Crew sequence. Blue curve H.264/AVC: settings in Section 4.2.1, Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations, Green curve H.264/AVC + BA4: Prediction using Best Approximation with Simplified Relaxation, 4 iterations.*

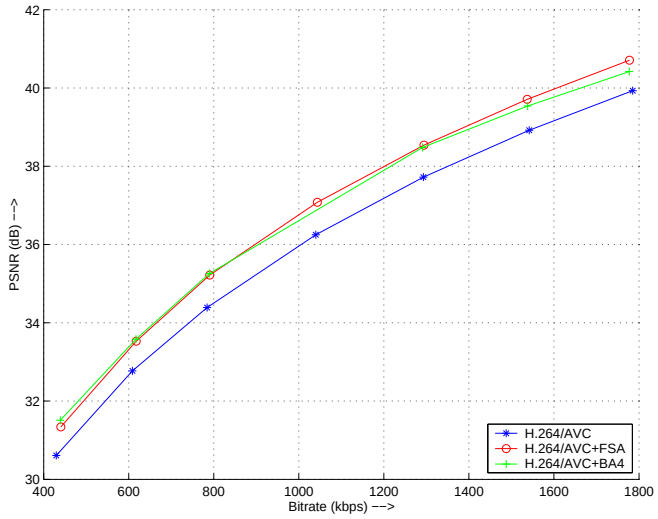

Figure B.6: *Performance of Best Approximation with Simplified Relaxation, Discovery city sequence. Blue curve H.264/AVC: settings in Section 4.2.1, Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations, Green curve H.264/AVC + BA4: Prediction using Best Approximation with Simplified Relaxation, 4 iterations.*

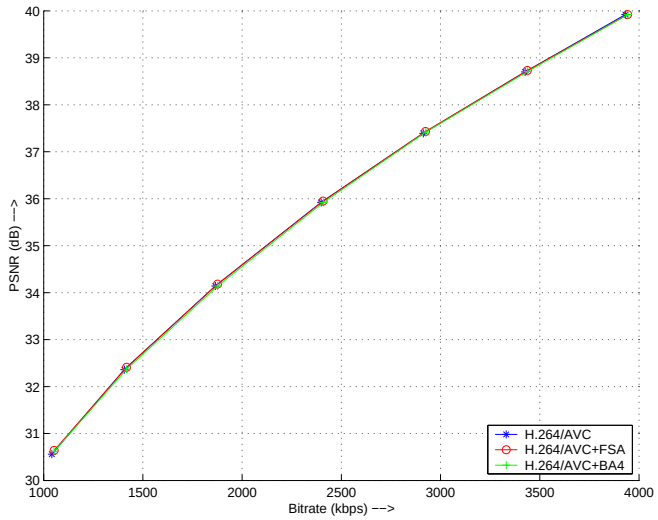

Figure B.7: *Performance of Best Approximation with Simplified Relaxation, Flower Garden sequence. Blue curve H.264/AVC: settings in Section 4.2.1, Red curve H.264/AVC + FSA: Prediction using FSA 200 iterations, Green curve H.264/AVC + BA4: Prediction using Best Approximation with Simplified Relaxation, 4 iterations.*

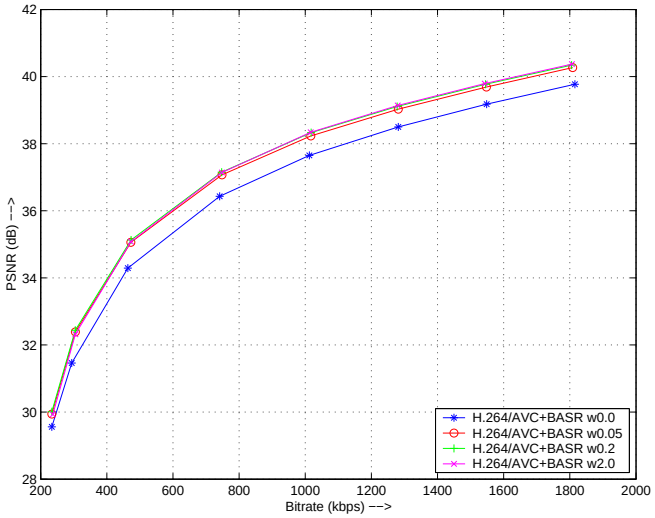

Figure B.8: *Influence of weighting factor, Vimto sequence. H.264/AVC + Best Approximation with Simplified Relaxation.*

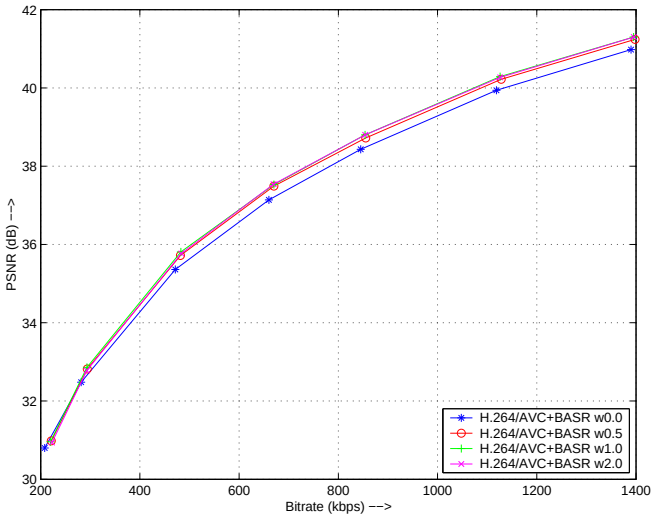

Figure B.9: *Influence of weighting factor, Crew sequence. H.264/AVC + Best Approximation with Simplified Relaxation.*

List of Figures

2.1 Hybrid Video Coder 14
2.2 Example Intra-Prediction 15
2.3 Motion Compensated Prediction 16
2.4 Spatial Refinement Schematic 18
2.5 RD Optimizated Decision 19

4.1 Adjacent blocks in a frame 34
4.2 FSA performance for Vimto sequence 43
4.3 FSA performance for Crew sequence 44
4.4 FSA DCT performance for Vimto sequence 44
4.5 Example of overfit 45
4.6 Performance of Oracle assisted FSA for Vimto sequence 46
4.7 Flowchart of oracle assisted FSA 47

5.1 Performance of Best Approximation, Vimto sequence 53
5.2 Performance of Best Approximation, Crew sequence 54
5.3 Performance of Best Approximation with Relaxation, Vimto sequence . 58
5.4 Performance of Best Approximation with Simplified Relaxation, Vimto sequence 59
5.5 Constant weighting in center block 61
5.6 Isotropic weighting in center block 62

5.7 Performance of Best Approximation with Relaxation and Isotropic Weighting, Vimto sequence 63
5.8 Performance of Oracle assisted Best Approximation with Simplified Relaxation, Vimto sequence 64
5.9 Performance of Best Approximation with Simplified Relaxation and Side Information, Vimto sequence 65

6.1 Error Surface and Negative Gradient 71
6.2 CWLS performance for Vimto sequence 73
6.3 CWLS performance for Discovery city sequence 73

7.1 Signal Space Illustration of POCS with Intersecting Sets 78
7.2 Signal Space Illustration of POCS with Non-intersecting Sets 78
7.3 Schematic of POCS Applied to Video 80
7.4 POCS performance for Vimto sequence 82
7.5 POCS performance for Discovery city sequence 82

B.1 FSA performance for Discovery city sequence 88
B.2 FSA performance, Flower Garden sequence 89
B.3 Best Approximation Performance, Flower Garden sequence 89
B.4 Performance of Best Approximation with Relaxation, Crew sequence . 90
B.5 Performance of Best Approximation with Simplified Relaxation, Crew sequence 90
B.6 Performance of Best Approximation with Simplified Relaxation, Discovery city sequence 91
B.7 Performance of Best Approximation with Simplified Relaxation, Flower Garden sequence 91
B.8 Influence of weighting factor, Vimto sequence 92
B.9 Influence of weighting factor, Crew sequence 92

Bibliography

[1] K. Meisinger. Selective signal extrapolation and its applicatin in image and video communications. *PhD thesis, Chair of Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg*, 2007.

[2] A. Kaup, K. Meisinger, and T. Aach. Frequency selective signal extrapolation with applications to error concealment in image communication. *Int. J. Electron. Commun.*, 59:147–156, 2005.

[3] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proc. 27th Annu. Asilomar Conf. Signals, Systems and Computers*, November 1993.

[4] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. Advanced video coding for generic audiovisual services. *ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC*, March 2005.

[5] M.T. Orchard and G.J Sullivan. Overlapped block motion compensation: an estimation-theoretic approach. *IEEE Transactions on Image Processing*, 3:693–699, 1994.

[6] B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1998.

[7] D. J. C. McKay. Information theory, inference and learning algorithms. *Cambridge University Press, ISBN-13: 9780521642989*, 2003.

[8] B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal of Computing*, 24:227–234, 1995.

[9] J. H. Friedman and W. Stuetzle. Projection pursuit regressions. *J. Amer. Statist. Soc.*, 76:817–823, 1981.

[10] A. Kaup and T. Aach. A new approach towards description of arbitrarily shaped image segments. In *IEEE International Workshop on Intelligent Signal Processing and Communication Systems, Taipei, Taiwan*, March 1992.

[11] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993.

[12] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing*, 20:33–61, 1999.

[13] J. Seiler. Schaetzung unbekannter bildbereiche in videosignalen mittels zeitlich-oertlicher selektiver extrapolation. *Diploma Thesis. Chair of Multimedia Communications and Signal Processing, University Erlangen-Nuremberg, Germany*, 2006.

[14] K. Meisinger and A. Kaup. Spatial error concealment of corrupted image data using frequency selective extrapolation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Montreal*, volume 3, pages 209–212, May 2004.

[15] J. Seiler, K. Meisinger, and A. Kaup. Orthogonality deficiency compensation for improved frequency selective image extrapolation. In *Proc. Picture Coding Symposium, Lissabon, Portugal*, November 2007.

[16] J. Seiler and A. Kaup. Fast orthogonality deficiency compensation for improved frequency selective image extrapolation. In *Proc. International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, Nevada*, April 2008.

[17] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50:2231–2242, 2004.

[18] A. Kaup and T. Aach. Coding of segmented images using shape-independent basis functions. *IEEE Transactions on Image Processing*, 7(7):937–947, 1998.

[19] A. Kaup. Modelle zur regionenorientierten bildbeschreibung, phd thesis. *RWTH Aachen, VDI Verlag, Fortschritts-Berichte VDI Reihe 10, no. 381*, 1995.

[20] J. A. Tropp. Topics in sparse approximation. *PhD thesis, The University of Texas at Austin*, 2004.

[21] V.N. Temlyakov. Weak greedy algorithms. *Advances in Computational Mathematics*, 12:213–227, 1999.

[22] A. Kaup. Adaptive constrained least squares restoration for removal of blocking artifacts in low bit rate video coding. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Munich*, volume 5, pages 2913–2916, April 1997.

[23] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[24] D.J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

[25] H. Sun and W. Kwok. Concealment of damaged block transform coded images using projections onto convex sets. *IEEE Trans. Img. Proc.*, 4:470–477, 1995.

[26] W. Cheney and A. Goldstein. Proximity maps for convex sets. In *Proceedings of the AMS*, volume 10, pages 448–450, 1959.

[27] D. C. Youla and H. Webb. Image restoration by the method of convex projections: Part 1-theory. *IEEE Trans. Med. Imag.*, pages 81–94, 1982.