MEE 08:19

Improving Intra Pixel prediction for H.264 video coding

Senay Amanuel Negusse

This thesis is presented as part of Degree of
Master of Science in Electrical Engineering

Blekinge Institute of Technology
May 2008

Supervisor: Dr. Kenneth Andersson[1]
Examiner : Dr. Benny Lövström[2]
1. Multimedia Technology , Ericsson research
2. Blekinge Institute of Technology (School of Engineering, Department of Signal Processing)

Improving Intra Pixel prediction for H.264 video coding

Abstract

H.264 is the latest most efficient video coding standard developed by joint collaboration between ITU-T's Study Group 16 Question 6 (Video Coding Experts Group VCEG) and ISO's MPEG forming a group known as the Joint Video Team (JVT). The JVT has added additional features to the previous video coding standards (MPEG-4 and H.263) which has allowed the current H.264 to provide a much more efficient compression ability giving a bit rate reduction of about 50 % of the previous standards for the same video quality.

This thesis is among the ongoing studies that deal with trying to improve the intra prediction scheme of H.264. It aims to modify the directional extrapolation scheme of the standard intra prediction and add a combined usage based on the local structure of previously reconstructed pixels surrounding the area to be coded. A new method of searching the direction of the local structure of an area using optimal filters is deployed. The method is based on the least squares algorithm. A modified coder structure is implemented in an attempt to obtain improved coding efficiency using the new prediction. The result based on visual inspection and objective evaluation of the new modified prediction against standard H.264 intra prediction is presented.

Key words: H.264, Intra-prediction, wiener filter, local structure

Foreword

I would like to express my deepest gratitude to Dr. Kenneth Andersson at Ericsson's multimedia research group, for all his guidance and encouragement through out my thesis, dedicating his time and energy. I would also like to thank my examiner at Blekinge Tekniska Högskola, Department of Signal Processing, Dr. Benny Lövström who has been following my work and helping me on my report.
I would not want to pass without thanking Jonatan at Ericsson's multimedia research group for his help and many others for making my stay enjoyable.
And finally, Yuan Chun (Clark), a fellow student at Chalmers Tekniska Högskola, who has been my office mate, I would like to thank him for his friendship, and wish him success in his future career.

Table of Contents

1. Introduction

1.1 Thesis background

In video coding typically the decoder fully obeys the encoder. In this thesis it is investigated if the coding efficiency of H.264 can be improved by giving more freedom to the decoder to operate on previous reconstructed pixels. More specifically intra coding and how to make the decoder aid in intra prediction is addressed here.

State of the art intra prediction as of H.264 makes a directional extrapolation of the neighbouring pixels on an adjustable block size to predict the current block. A recent proposal in ITU-T Study Group 16 Question 6 (Video Coding Experts Group VCEG) has showed substantial improvement in rate-distortion performance and is based on template matching [6]. It goes away from the standard concept and gives flexibility to the decoder. The template consists of pixels surrounding a target pixel. A candidate with the same shape as the template is displaced among previously reconstructed pixels. The candidate with best match with the template is selected. The prediction of a target pixel is made from the pixel near the candidate region that corresponds to the target pixel. Another recent work, *enhanced intra-prediction using context-adaptive linear prediction* [8], uses two windows of reconstructed pixels from two previous frames to train coefficients (inter-frame training) that would be used in the neighbouring reconstructed pixels of the current pixel to adaptively follow the local structure by doing the inter-frame training for each pixel in the 4x4 block to be predicted. This method, although it has shown a good performance in rate distortion up to -17.95% bit saving for the tested "*foreman CIF*" sequence, requires a considerably high computational complexity in both the encoder and the decoder, requiring nine linear equations per pixel in a 4x4 block.

In contrast to intra prediction by template matching this thesis will operate only on the nearby previously reconstructed pixels thereby omitting the search of the position of the candidate template. More over, in contrast to the LSP method the training region is here only in the current frame and uses much fewer free parameters which reduces complexity significantly. It will be shown that by estimating the local structure and texture from the neighbouring pixels of a target block of pixels, a continuation of the local structure can be produced by linear weighted combination of the neighbouring pixels making up a prediction of the target block. The least squares estimator is used for estimating the local structure and produce the optimal weights that would be used to predict the target block. The aim will be to insert the new method as an additional mode to the H.264 standard intra prediction modes for intra 4x4 luminance prediction. Complexity of the algorithm was also put under consideration while maintaining improved coding efficiency.

This thesis was implemented and tested on a proprietary H.264 codec in C language developed by Ericsson's multimedia research group. Microsoft Visual C++ compiler was used for the implementation of the codec. To view test video sequences and make objective and subjective evaluation, Ericsson's VIPS viewer was used along with MATLAB.

1.2 Organization of report

This report is organized as follows. **Chapter2** provides as a beginning the basics of video samples and the H.264 codec. **Chapter 3** specifies the problem this thesis looked at and the proposed implementation for study**. Chapter 4 & 5 ,** deal with method of estimation of local structure used in this thesis and features of the new block type along with results for the statistical analysis with regard to its percentage of selection and the gain in rate distortion. **Chapter 6** will do an evaluation of the algorithm implemented in the thesis and finally **Chapter 7** contains recommendation for future work.

2. Fundamentals of video coding and H.264/AVC

2.1 Video Samples and Frames

Video signals are three dimensional, two spatial dimensions along with the temporal dimension. Spatial samples are taken from a frame or picture at a single point in time. Temporal samples refer to periodical capture of a scene by the camera forming sequence of frames. A single frame or picture of a video scene is a temporal sample. The higher the temporal sampling rate the closer the video appears like a natural scene and the more band width is required. Temporal video samples can form a complete frame referred to as progressive sampling or a sample of half the frame data in which case it is known as interlaced sample. Video camera sensor decomposes the light to represent a sample in terms of the components of RGB (red green blue).
One major advantage that is utilized in video compression is the fact the human beings are less sensitive to colors, which allows for the color samples to be represented with less resolution and giving higher resolution to the more perceived samples, the luminance. These samples are commonly referred to as the YCbCr color space samples. See [1] for an explanation how the luma sample, Y, and the chroma samples Cb & Cr are derived from RGB color images.

Sampling Formats: As explained above, human visual perception is more sensitive to luminance Y value than the chromas Cb & Cr. It is therefore common in modern compression techniques to have chroma sub sampling to represent chroma values with less resolution than lumas and thus making more bandwidth available for luma samples. Below are three patterns of sampling in MPEG-4 and H.264

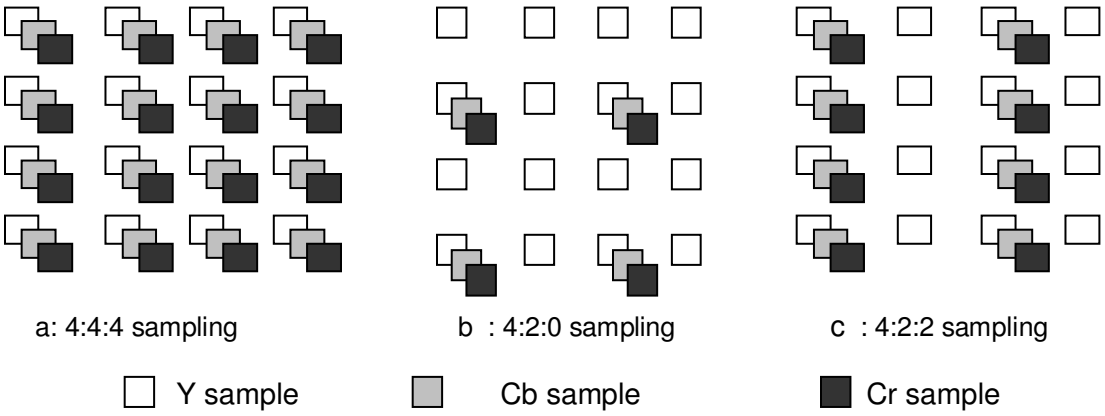


Figure2.1:H.264 Sampling formats

The first pattern of sampling, 4:4:4, shows a sampling with all the Luma and chroma sampled with equal resolution that is without any sub-sampling of the chroma samples, the notation '4:4:4' denote the sampling rate of each YUV component horizontally. As can be seen from Figure2.1.a, for every four luma pixel in the horizontal direction, there are U & V chroma samples available. In the 4:2:2 pattern, shown in Figure2.1.c, the numbers indicate that for every four luma samples horizontally, there are two chroma, U & V, samples; the chromas are sub-sampled to have half resolution of the luma in the horizontal direction while still having the same vertical resolution. The third pattern, 4:2:0, most commonly used in commercial DVDs, sub-samples the chromas, Cb & Cr, to quarter of the number of samples in Y, each having half the horizontal and vertical resolution of Y. Unlike the two formats discussed above, the numbers 4:2:0 do not actually have clear relation to the

pattern and tend to be confusing; they were chosen as a “code” to differentiate the formatting pattern from 4:4:4 and 4:2:2.
The test video sequences used in this thesis have a .YUV extension name which implies to storing colors as Y, U, and V values; Y stores the luminance (luma) and U and V store the chrominance (Cb & Cr) components, which can be used to store image color information more accurately than the typical RGB format.
Video Formats: Video Formats refer to the resolution of samples that a frame in a video can have. Video frames can be represented in various formats depending upon the applications. Different set of video formats are standardized used for compression and test of algorithms .The Common Intermediate Format (CIF) is the original basis format from which other sets of formats are derived as shown below.

Format	Description	Resolution
QCIF	Quarter CIF	176x144
4CIF	4 x CIF	704x576
16CIF	16 x CIF	1408x1152

Table2.1: H.264 formats

2.2 Video quality measurement

Analysis and evaluation of compression systems and algorithms in video coding is not an easy task. There usually exists a difficulty while making a test involving human perception which arises from not having consistent results. A video image could go under two types of evaluations to assess its quality, a subjective quality test and objective quality test.

Subjective quality measurement: refers to an assessment of video quality made based on human beings’ perceptual ability to make a non-metric decision. Mean opinion score is one of the means how subjective measurements are taken. As the name implies, a group of people are made to watch sequences of video, or images, and an average of their opinion regarding their quality is registered to indicate the result. Lots of factors could affect the result of an assessment of video quality made by human visual system leading to time consuming processes and inconsistent result.

Objective quality measurement: is a consistent and inexpensive means of by which processed video quality is measured using a standardized algorithm. The most common objective quality measurement used in current video and image compression quality assessment is the peak signal to noise ratio, “PSNR” referring to the peak signal power in the image and the power of the corrupting noise. The decoded and reconstructed picture quality is usually measured by its PSNR value. Let S(i, j) be the original picture sample of picture S and Sr(i,j) a sample from its reconstructed version Sr , then the distortion power is computed by calculating the mean squared error between S & Sr as below

$$MSE = \frac{1}{MN}\left[\sum_{i=0}^{i=M}\sum_{j=0}^{j=N}\left(S(i,j) - Sr(i,j)\right)^2\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

where M & N represent the resolution of the picture along the height and width. Therefore the PSNR in logarithmic scale is defined as

$$PSNR_{db} = 10\log_{10}\frac{(2^n - 1)^2}{MSE} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

(2^n - 1) refers to the highest possible signal in the picture and n is the number of bits per sample.
When analyzing image quality using PSNR computation, it is assumed that the original reference image is 100% distortion less.

2.3 H.264 Codec

H.264 advanced video coding is the latest standard which has achieved a compression efficiency of about 50% for the same picture quality in the previous standard, MPEG-2 and H.263**.** It is currently the most efficient and highly reliable standard available for video communication, broadcasting and storage applications which are currently the most widespread in the market and implemented in both existing and future networks. H.264 has provided great flexibility due to its representation of video in the Video Coding Layer (VLC) and Network Abstraction Layer (NAL) which formats the video contents for a suitable transmission across various types of networks.

2.3.1 Macroblocks and Slice groups

Block based video compression systems, as H.264, partition a frame into blocks of 16x16 luma pixel size and 8x8 pixels of chroma, referred to as a “macroblock” (MB). Macroblocks in a frame are the basis upon which video compression standards are implemented. A picture can be split into one or more group known as slice groups by ordering the macroblocks into families based on their prediction types.

- I-Slice types, containing intra predicted macroblocks
- P-Slice types, containing inter and intra predicted macroblocks
- B-Slice, with macroblocks predicted by two motion compensation vector along with P type macroblocks

By coding each slice in a frame independent of one another error propagation is avoided, more over, slices give the advantage of robust error control by localizing error replacing it by a new slice produced by interpolation of neighbouring slices.

2.3.2 The H.264 Codec structure

There are two data flow paths in the H.264 codec, the *forward* and the *reconstruction* path. With the exception of the in-loop de-blocking filter, all the functional blocks shown below are present in all the previous standards. The blocks in grey show functional blocks which come across the reconstruction path, emulating the process in the decoder to have identical reference frames for prediction in both encoder and decoder. Note from the block below that the reference encoder decoded and reconstructed frame is un-filtered prior to being used as a reference for intra prediction.

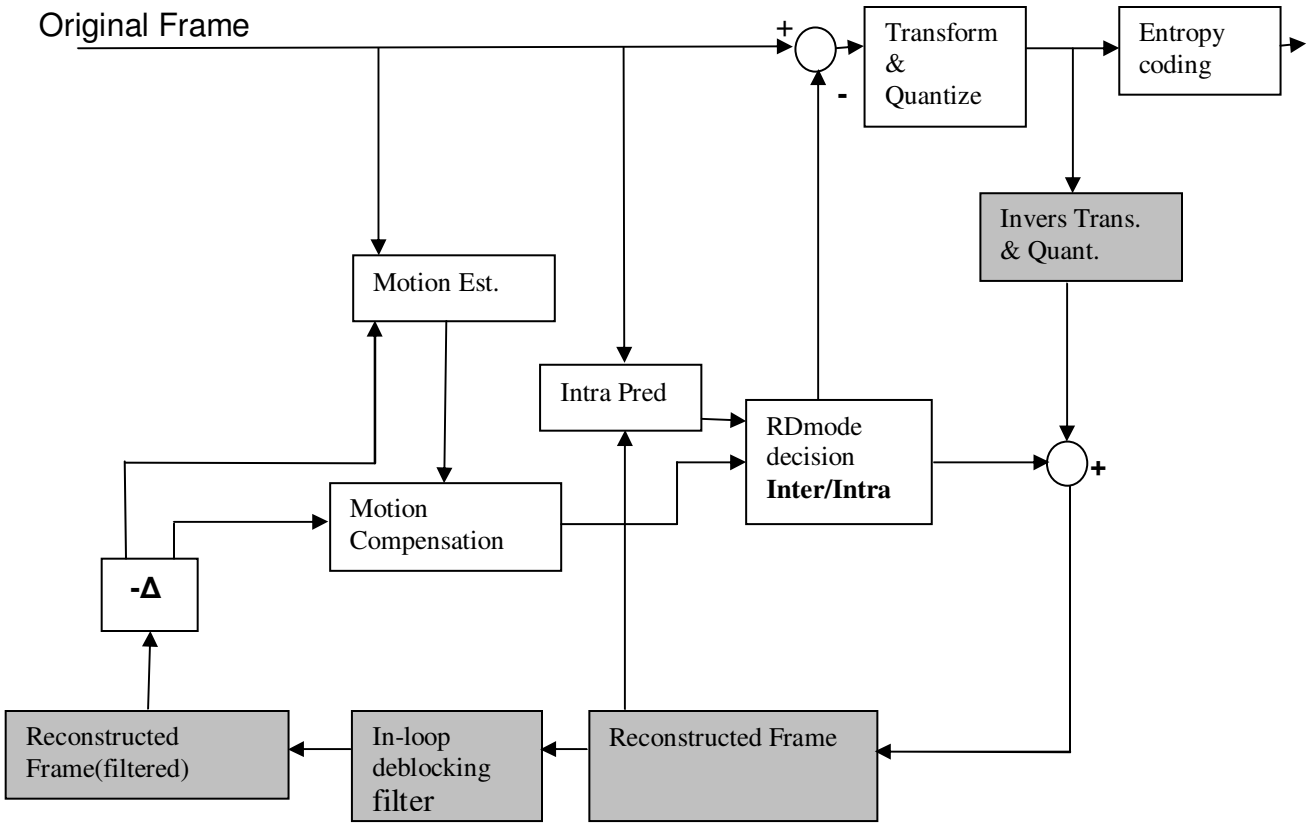
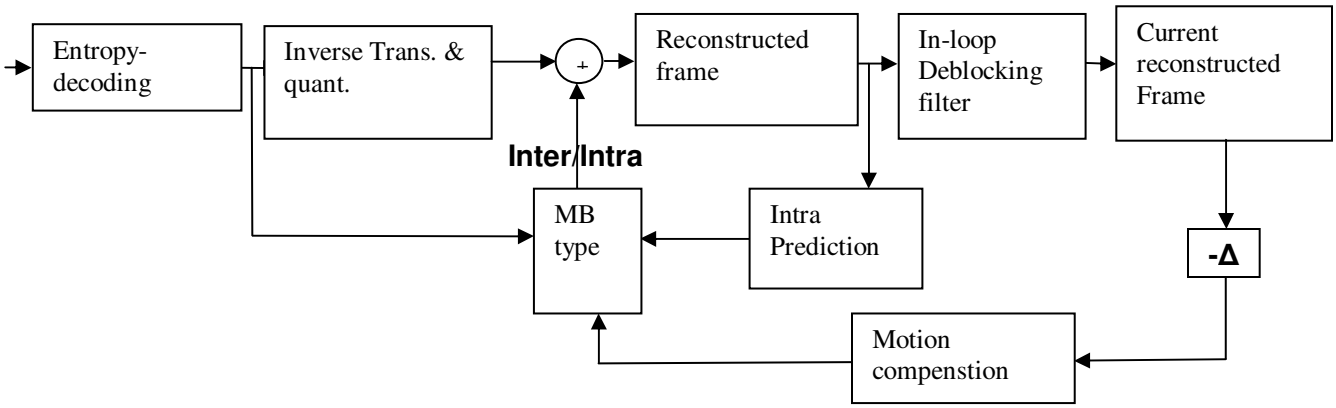


Figure2.2: H.264 Encoder block

In the decoder, after entropy coding and inverse transformed and quantized, Macroblock header information is also sent along with the residual video data to guide the decoder which prediction scheme to use(inter/intra) to produce a prediction of a current macroblock from a reference reconstructed frame identical to the one in the encoder



Decoder block

Figure2.3: decoder block, which is the scope of the JVT standardization, by restricting the bit stream, syntax and decoding order

2.3.3 Motion estimation and compensation

Inter-prediction generates macroblocks from previously encoded and reconstructed frames by using block based motion vectors obtained by motion estimation and matching in a localized search area in a reference frame. One factor that makes H.264 different from the previous standards is availability of variable block size for the motion estimation. As can be seen below, there various sub-macro block partition types for what is known as tree structured motion compensation. Division of macro blocks into sub blocks provides flexibility in motion search for a greater accuracy. A 16x16 macroblock can further be divided in to 16x8 ,8x16, 8x8 sub-blocks and further on in an 8x8 sub-macro block basis 8x8 ,4x8,8x4 and 4x4 sub-blocks, with each with each block having independent motion compensation vector and thus improving prediction accuracy.

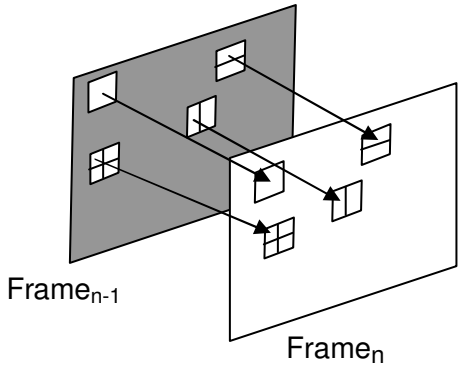
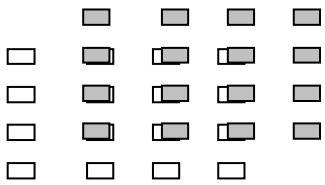
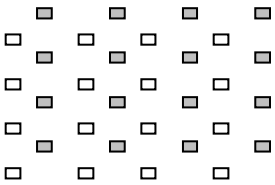


Figure2.4: Motion estimation for 16x16, 16x8, 8x16 & 8x8 blocks

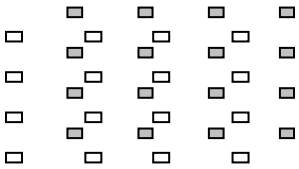
An obvious disadvantage to partitioning the macro block is that it requires as many motion vectors as there are partitions per macroblock to make motion compensation, and thus increasing the overhead in bits. An improvement in prediction accuracy was also made by using sub-pixel motion compensation (half-pixel and quarter pixels) by interpolating adjacent pixels at the expense of computational complexity.



a: Integer pixel motion compensation



b: Half pixel motion compensation



C: Quarter pixel motion compensation

Figure2. 5: Motion compensation for 16x16, 16x8, 8x16 & 8x8 blocks

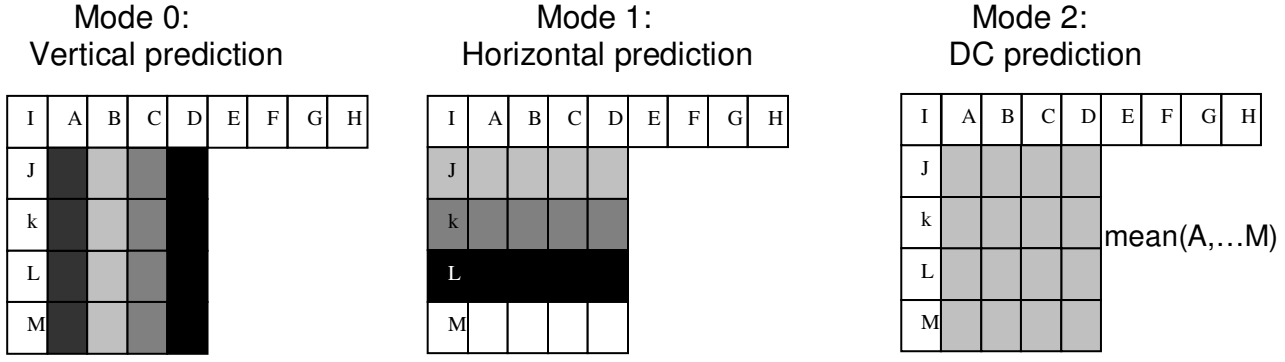
2.3.4 Intra prediction

Spatial redundancy within an image is the basis for image compression and standards such as JPEG. H.264's Intra-prediction approach can be viewed as an image compression technique that is applied to video frames along with temporal prediction for high quality video compression. Comparison of H.264's intra prediction with that of JPEG2000 has shown a better performance in PSNR for compression of monochromatic images with the only back side being the blocky artifact inherent to block based transform compressions [14]. Intra-predicted macroblocks do not refer to any other frame beside the current one for a reference. Neighbouring pixels around the current macroblock are extrapolated to make prediction.

Figure2.6: Intra-prediction of macroblock from neighbouring block

There are two types of intra prediction for the luma components in a frame, INTRA-16x16, which performs the prediction of the entire macro block all at once applied in smooth areas of the current frame, and INTRA - 4x4 which predicts the macro block in sixteen 4x4 block partition within a macroblock. INTRA-8x8 intra-prediction type is used to predict the chroma components of a frame. A macro block is partitioned into four 8x8 sub-blocks for prediction of the chroma samples all with identical prediction mode.

There are four directional prediction modes for INTRA-16x16 predictions which use reconstructed pixels above and to the left of the macro block to make four types of extrapolations. The prediction modes of INTRA 16x16 is also shared by INTRA 8x8 for chroma prediction with identical prediction modes for all the 8x8 sub-blocks in the macroblock. Standard INTRA 4x4 derive its prediction using nine modes as shown below. Extrapolation of neighbouring reconstructed pixels is performed in nine different directions with to try and see the best direction that the current 4x4 follows.



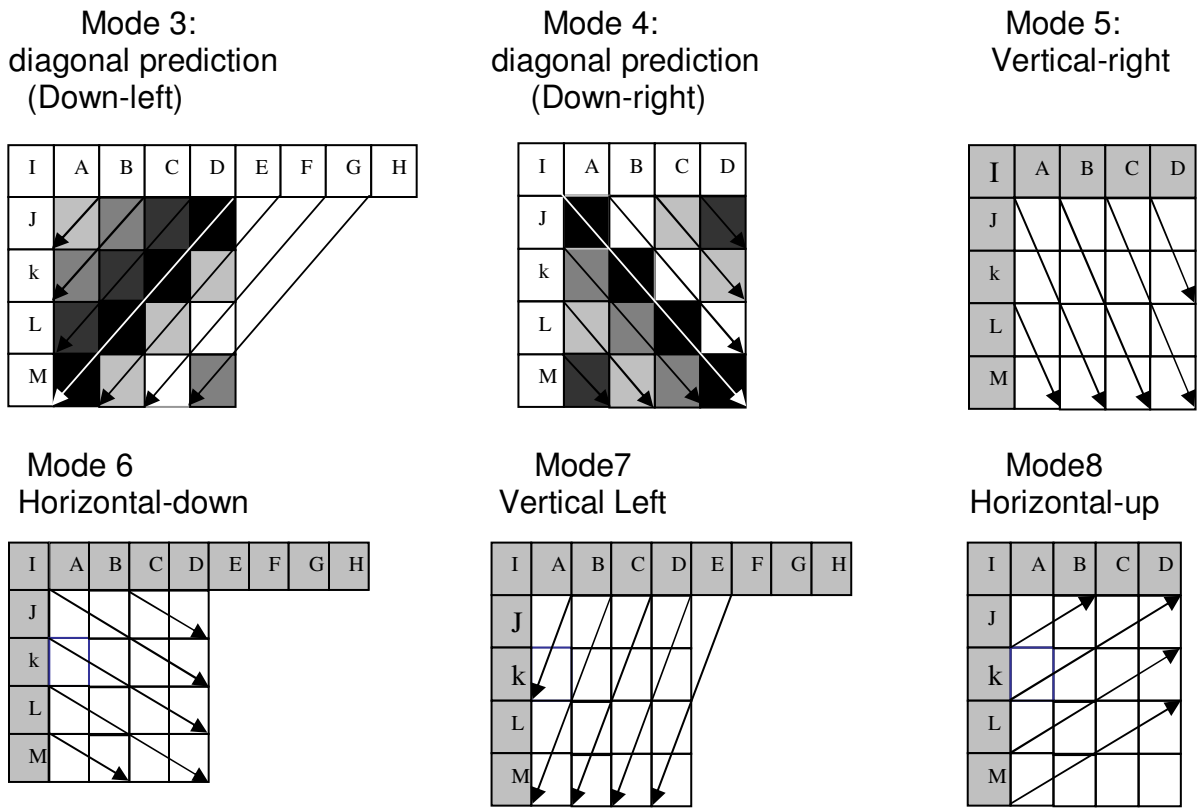


Figure2.7: Intra 4x4 prediction modes

Modes 0 and 1 copy the reference pixel values of the immediate neighbors downwards vertically and to the right horizontally respectively; while Mode 2 takes the mean of all the neighbouring pixels to predict the current block taking homogenous luma value. Mode 3 to 8 use low pass unity magnitude extrapolating filters of like , [1/4,1/2,1/4] for three immediate neighbouring previously decoded support pixels and [1/2,1/2] for two support pixels, to determine the reference pixel values in the block .Modes 0, 1 & 2 of intra 4x4 are shared by INTRA 16x16 luma blocks and INTRA 8X8 chroma block types; one more mode, mode 3, is included for intra 16x16 luma and 8x8 chroma blocks, which uses weighted combination of horizontal and vertical adjacent pixel for prediction, known as planar prediction, as shown below.

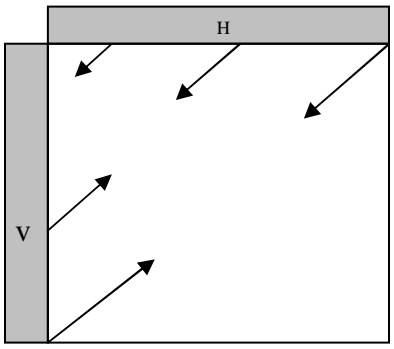


Figure2. 8: Planar prediction (mode 3) for intra 16x16, modes 0, 1, 2 are same as Intra 4x4

2.3.5 Block Transform, quantization and entropy coding

In hybrid video coding, temporal predictions are further improved by removing the spatial correlation in the residue signal. Since H.264 has improved both the temporal (inter) and spatial prediction of a block, the use of 8x8 transform block size in other standards was replaced by a 4x4 block DCT like integer transform (matrix H shown below) and thus reducing noisy block artifacts.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

The use of integer transform gives H.264 codec a major advantage in such a way that an exact inverse transform can be obtained in the decoder. In addition to that, integer transforming makes it computationally easier for implementation of the algorithm using additions and shifting only. In order to avoid division and multiplication, transform and quantization are performed as a single operation by incorporating the transformation inside the quantizer. But the general formula for quantizing the transformed coefficients is as shown

$$Z_{xy} = \text{round}\ (Y_{xy} / \text{Qstep}) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (3)$$

Currently a set of 52 quantization parameters is available in H.264. A one step increase in the quantization parameter implies an approximate increase of 12% in Qstep. Therefore starting from quantization parameter Qp=0 which represents Qstep 0.625 it goes on up to QP 51 with Qstep 224.

H.264 implements Context-Adaptive Variable Length Coding (CAVLC), which is implemented in this thesis, and Context-Adaptive Binary Arithmetic Coding (CABAC), implemented in the main profile of H.264. CAVLC as the name implies uses variable length coding scheme and codes the transformed coefficients by exploiting inter symbol redundancy switching VLC tables for various syntax elements depending on previously sent coding symbols. CABAC achieves considerably higher compression efficiency by combining an adaptive binary arithmetic coding technique with context modeling. Although CABAC gives a more efficient compression than CAVLC, the ease of processing of CAVLC makes it more desirable for use. (See [1] for detail on CABAC & CAVLC)

2.3.7 In-loop de-blocking filter

In-loop de-blocking filter is implemented in the H.264 standard only to remove the blocky artifact common in block based video coding systems**.** Unlike in the previous standards which uses de-blocking filter in the decoder only, H.264 implements In-loop de-blocking in both the coder and decoder for each 4x4 block. An adaptive filtering strength is implemented on pixels at the edge of each 4x4 block since edges are reconstructed with less accuracy than interior pixels.

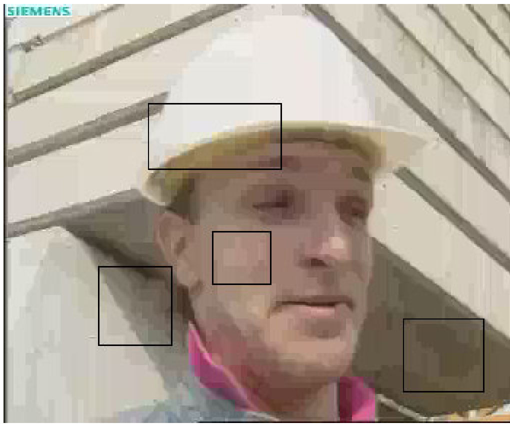


Figure2. 9a: unfiltered reconstructed frame **Figure 2.9b**: Filtered reconstructed frame

De-blocking filter works by smoothing the edgy appearance between adjacent 4x4 blocks. If the pixels in neighbouring adjacent blocks happen to be representing a sharp area in an image, then there would naturally be a considerably large absolute difference between the pixels, and hence use of the filter would be avoided. However, if their absolute difference is relatively large with respect to their neighbors, it would be assumed that the difference is caused by the blocky artifact and thus the filter is applied.

2.4 Rate-distortion optimization in H.264

Availability of two prediction types with different macroblock types, and prediction modes for each, requires an optimal means of selecting the best prediction mode to code a macroblock region in a frame. Standard H.264 uses lagrangian optimization method in order to counterbalance the compromise between cost of bits and quality of video for a given quantization parameter in an equation shown below

$$\text{Min}(J) = \text{Min}(D + \lambda . R) \quad \text{……………………………………………}(4)$$

where D is a measure of distortion energy and R is cost of bits per macroblock. Lambda (λ) is the lagrangian constraining parameter which is a function of the quantization parameter (QP) given as

$$\text{Lambda}(\lambda) = 0.85 * 2^{(QP-12)/3} \quad \text{…………………………………… (5)}$$

Distortion measurement: Two means of distortion energy measurement are used in standard H.264 and in this thesis as well; the sum of squared difference (SSD) and the sum of absolute difference (SAD), both of which refer to the difference between the original and the predicted value of pixels. D above is usually taken as SSD while making a rate distortion computation. See [11] for details on rate distortion optimization.

3. Motivation and Proposed approach

3.1 Analysis of intra mode selection

Since part of the objective is dealing with addition of new mode/modes to the nine standard intra predictions and possibly replacement, it is worth looking at how modes are selected and ordered for a later reference on how to re-position them. As mentioned in the previous chapter, the rate distortion cost between INTRA predicted macroblocks and INTER predicted macroblocks is computed to determine the best mode for prediction of the current macroblock. But prior to that, a rate-distortion (RD) computation of each mode in intra prediction is done to determine the best mode for prediction. SSD of each prediction for all intra modes and cost of bits including the number of bits required to signal the mode is computed in an extensive rate distortion computation to determine the mode with the least RD cost. The order of prediction as is put in the previous section (mode 0 for vertical, mode 1 for horizontal……etc...) is set in the standard based on the average percentage of choice of modes for various test sequences. The percentage mode distribution for the *foreman CIF* sequence is shown below for QPs 22, 30, 37.

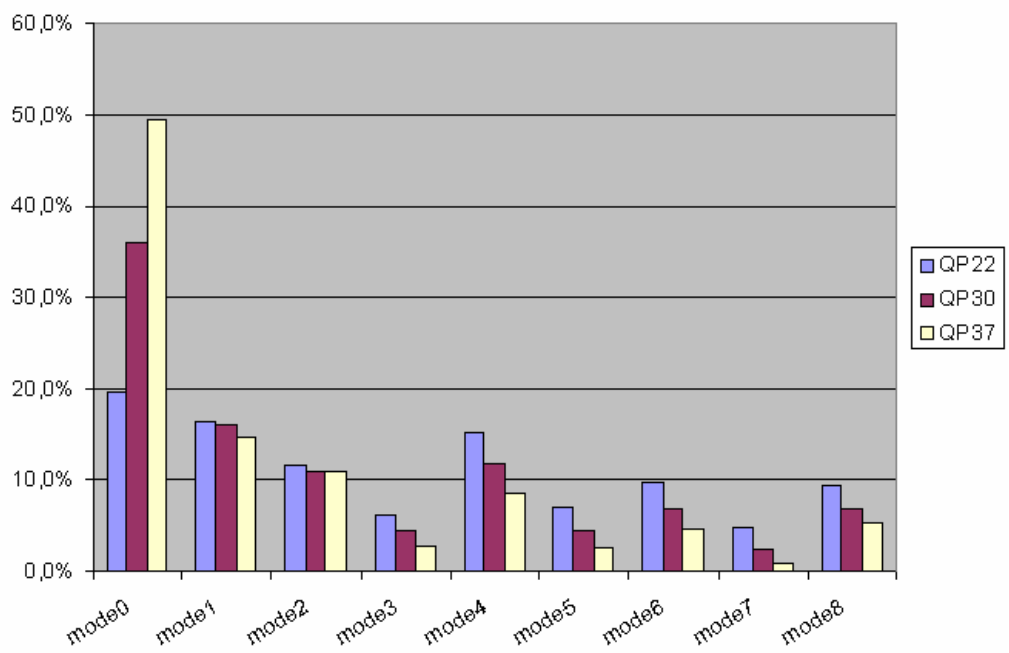


Figure3.1: percentage choice of each mode for foreman CIF sequence

Observe that the higher quantization parameter the more dominant the vertical, horizontal and DC modes are, implying that much of the image texture is lost in higher QPs and thus smoothening the reconstructed picture.

Signaling the choice of mode of prediction to the decoder adds a large amount of bits to the total encoded video file. Therefore, standard Intra prediction exploits the correlation between neighbouring modes to reduce the total overhead cost of signaling the prediction modes.

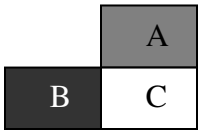


Figure3.2: A 4x4 target block C with neighbouring blocks A & B, which help to predict the prediction mode of C

Assuming 4x4 block “C” to be the current block to be Intra predicted, then the coder and decoder simultaneously perform prediction of the mode based on prediction modes of adjacent blocks (A & B) to determine the most probable mode of C. Exploiting the fact that local structure in a small area tend to have similar orientation and thus similar directional prediction, a one bit flag is sent for a predicted mode or else four (three for the rest of the modes and one to set off the flag to zero) bits if the mode is other than the predicted. Below it is shown a simple flow chart for mode prediction algorithm from neighbouring blocks used in H.264.

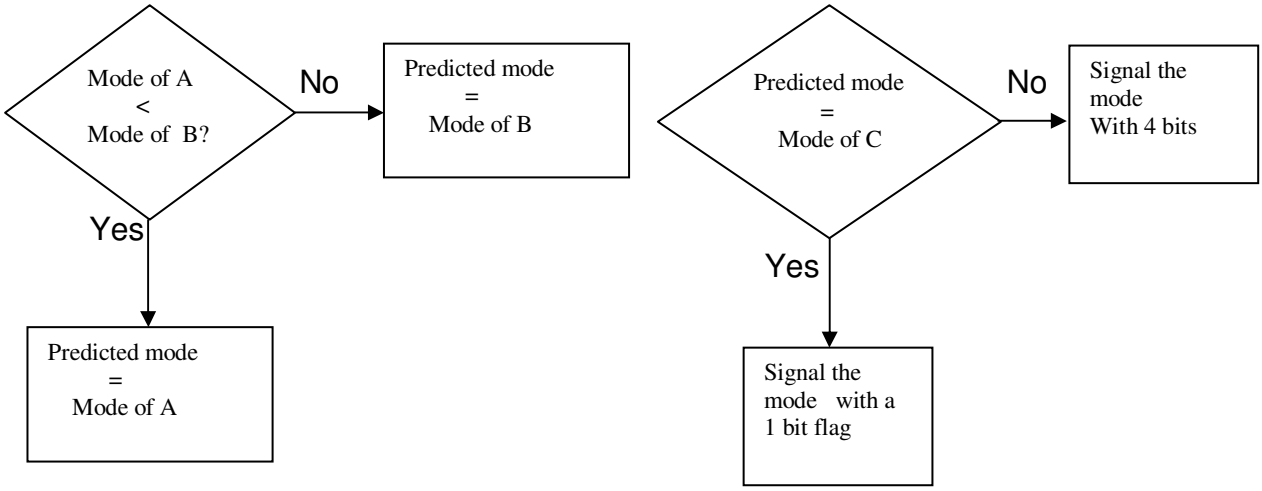


Figure3.3: Assigning bits for signaling modes of prediction

One bit for flagging whether a mode is predicted or not and if not three bits more for representing the actual mode is enough to accommodated nine modes.
However in a case, such as in this thesis, where it is desired to combine a new prediction mode with the standard ones, the signaling of the modes is modified in an optimum way to accommodate an additional new mode.
Prediction of a block by extrapolation of the neighbouring pixels has the advantage of being computationally easy. However, RD computation requires prediction in all available modes be coded and reconstructed. To alleviate the extensive time consumed in the RD computation different fast RD computation algorithms for intra prediction are being developed [12].

3.2 Motivation

Inter prediction generally performs much better than Intra-prediction; the percentage that the rate distortion computation chooses a macroblock to be coded using inter prediction is usually high.

Sequence	Inter prediction	Intra 16x16	Intra 4x4
Foreman CIF	99.12 %	0.43%	0.45%
Shuttle Start 720	81.45%	2.19%	16.36%

More over, Intra 16x16 prediction is used to code smooth areas of a picture, which in another way round could also mean that, an area predicted and reconstructed by intra-16x16 is smooth, as can be seen from the foreman frame below.

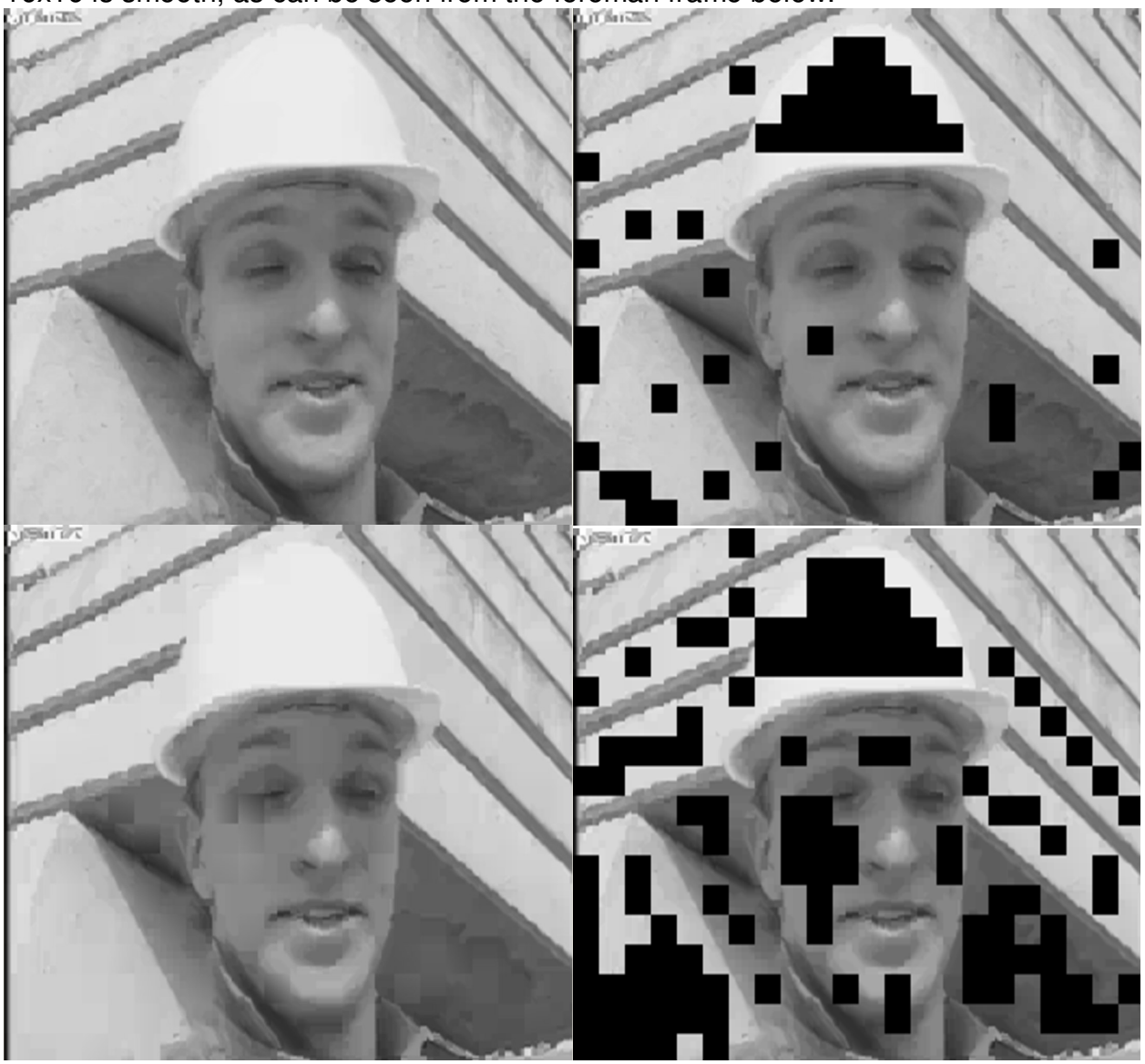


Figure3.4: Foreman CIF Standard prediction for QP 22 (Top row), and QP 32 (bottom row), with the black macroblocks showing Intra 16x16 coded areas

The larger the QP used, the more macroblocks which are encoded using Intra 16x16. Therefore, any hope of capturing a local texture/structure of a picture should concentrate on small area. The more localized an area in a picture is, the more correlated the pixels around that area are, intra 4x4 prediction have therefore the capability to hold more structural and textural information than intra 16x16. However, currently standard intra prediction only uses pixels from the immediate neighbouring reconstructed blocks and does not take in enough statistical information to exploit more spatial correlation and make a more accurate estimation of the target area.

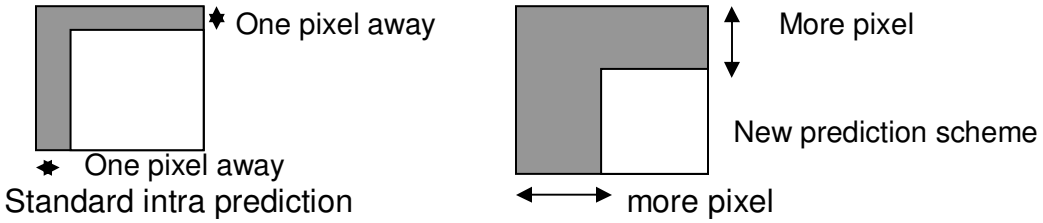


Figure3.4: demonstration of referencing neighbouring pixels for prediction in standard prediction and new prediction method to be introduced

Instead of only using the immediate neighbouring pixels above and to the left, it is desired to try and see if the performance of Intra prediction can be improved taking in more information from the reconstructed pixels by including pixels beyond the immediate neighbors (pixels in the above and left block) and exploiting the statistical correlation between rows and columns to check the direction of the local structure to make a prediction based on the estimated local structure.
At this point, it should be obvious why it was necessary to turn off all other modes except intra prediction. It would be almost impossible to observe the attribute of any modification on intra prediction when it is running along with inter prediction.

3.3 Proposed approach to implementation

As a start, the concept of motion estimation and compensation was though of as a way of capturing the textural/structural shift in a local area.

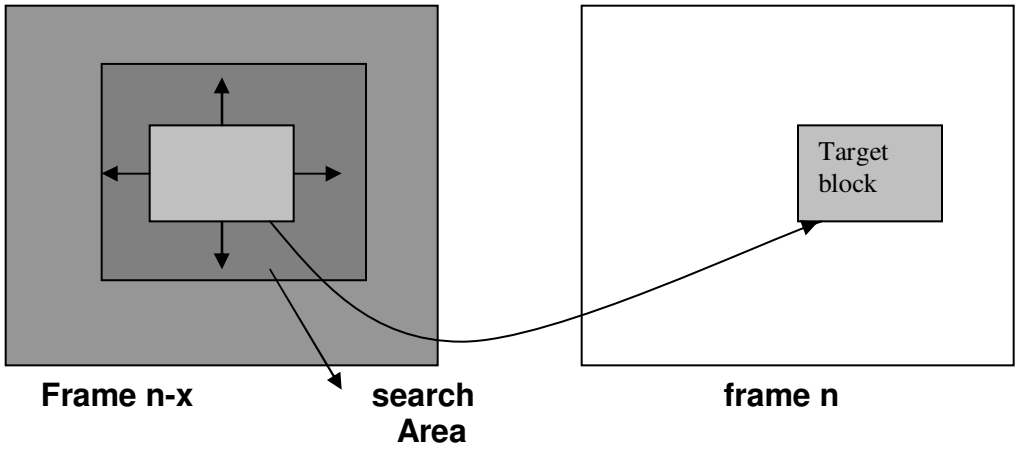


Figure3. 5: Matching block search

What happens in a motion estimation is that a search for a matching block of the current target block is done in temporally neighbouring frame (previous or future frames) by shifting blocks in their surrounding neighborhoods in a search area by half pixel, integer pixel or quarter pixel and after finding the match with the lowest absolute difference (error) with the current block, motion vectors are assigned to the block specifying its shift or displacement in the previous frame.
In a similar manner, it was thought of that adjacent rows or columns of pixels could be shifted sideways to have a close match of their immediate adjacent row of pixels or column of pixel. Therefore, by aiming at a target row or column, an adjacent row/ column could be

shifted sideways or upwards/downwards to match the target row or target column respectively.

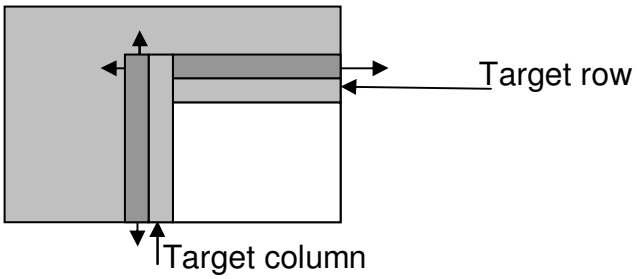


Figure3.6: Row/column shift search

The dominant structure would determine the direction of shifting for the forthcoming rows or columns of pixels to be predicted. The idea of shifting the adjacent arrays by half pixel, quarter pixel or integer pixel as in motion estimation was too restrictive and did not give much freedom to the modeling of local structure. Therefore, the method studied and implemented in this thesis is to generate a new intra prediction mode which exploits the local neighbouring decoded and reconstructed pixels to find the local structure of the image area and make prediction of the current 4x4 block by following the direction of the estimated local structure. The decoder is also made to search the local structure in the same manner as the coder since it will also have neighbouring reconstructed pixels and thus obtaining the same estimation and the same prediction for the current 4x4 block. Giving the search a degree of freedom, the least squares algorithm and Optimal Wiener filtering was used as the basic tool to train coefficients to the local structure of an image and make prediction of a new 4x4 block based on the trained coefficients by filtering the neighbouring reconstructed pixels.

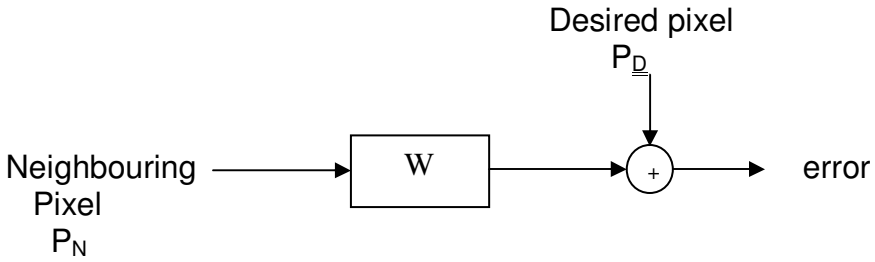


Figure3.7: Elementary wiener filtering diagram

Minimizing the square of the error ($P_D - P_N$) would lead to the computation of the wiener-hopf equation given as $W = R_N^{-1} r_{DN}$, where R_N and r_{DN} are the auto correlation matrix of the neighbouring pixels and the cross correlation vector between the desired pixels and neighbouring pixels respectively.

3.4 Limitation

This thesis addresses H.264's baseline profile which uses CAVLC. Furthermore only coding using "I" frames will be evaluated and only modification of the luma 4x4 intra prediction is considered.

4. Block prediction by estimation of local structure

4.1 Introduction

Standard Intra prediction for 4x4 blocks as mentioned in the previous chapter uses nine directional extrapolations from the neighbouring pixels to predict the target 4x4 blocks in a 16x16 macroblock. Different extrapolations of pixel values is tried out to decide which direction best estimates the structure of the 4x4 block with regard to the original in terms of distortion and cost of bits to represent the residual. In this chapter, it will be shown that by estimating the local structure in a more flexible way a new block type can be predicted according to the local structure of previously decoded pixels.

4.2 Least squares estimation of local structure

Local image orientation estimation is a basic tool in image processing and compression. In general the local image orientation is similar in nearby pixel positions. This feature is exploited in this thesis by predicting pixel values in the direction of the local structure. Taking this in a 4x4 target block, assuming there are neighbouring reconstructed blocks available, a weighted combination of the neighbouring pixels can be constructed to get an approximation of a target pixel. This assumption is the basis to model the structural shift between immediate rows and column of pixels in a block and use the model to make an extension of the structure or texture and thus make the prediction of the target 4x4 blocks.
Let the original pixel be I, then a linear weighted combination of neighbouring pixels can be constructed as below,

$$\hat{I} = I_{1*}W_1 + I_{2*}W_2 + \ldots\ldots + I_{n*}W_n \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

Where $\hat{I}$ is the approximated pixel and
$I_1, I_2 \ldots\ldots I_n$ neighbouring reconstructed pixels
$w_1, w_2 \ldots\ldots w_n$ are the estimator weights

Below is an artificial frame texture for demonstration

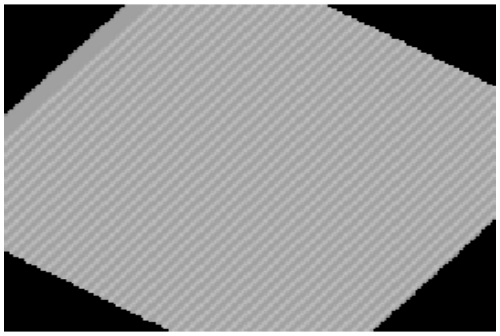


Figure4.1: Artificial frame of texture

What you see in the figure is a simple wall of distinct texture with a certain orientation. Now, the aim would be to obtain an approximation of the above picture constructed by 4x4 blocks which are predicted from their neighbouring pixels above and to the left using the original pixels (previously coded and reconstructed pixels are used when this is implemented in H.264 encoder). Below is shown a target 4x4 block along with neighbouring pixels of two rows (R_1 & R2) above and two rows to the left (C1 & C2).

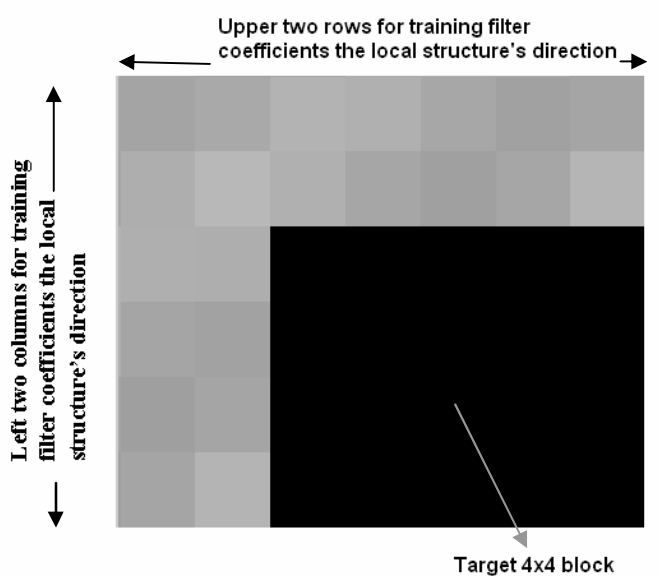


Figure4.2: a 4x4 block with neighbouring two rows and two columns

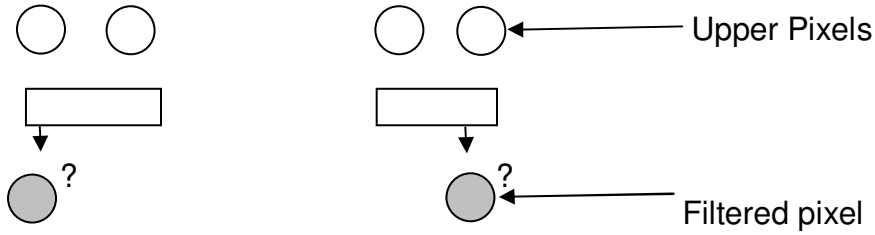
The shift in intensity between the rows or columns is what basically gives the character of the texture. In this sense, a row of luma pixels Ř1 can be constructed that is as close to R1 in least square error sense by shifting pixels in R0 side ways. In the same manner, column of pixels Ĉ1 which approximates C1 can be constructed by shifting the pixels' intensity of C0 upwards or downwards. Therefore, predictor coefficients (w_1, w_2……. w_n) can be generated using the least squares algorithm which works on minimizing the squared difference between pixels of R1 & their approximated pixels in Ř1, where Ř1 is
Ř1= [$\text{P̃}1_0$ $\text{P̃}1_1$ $\text{P̃}1_2$ $\text{P̃}1_3$ $\text{P̃}1_4$ $\text{P̃}1_5$]**,** which is an array of approximated pixels of R1 and
Ĉ1= [$\text{Ũ}1_0$ $\text{Ũ}1_1$ $\text{Ũ}1_2$ $\text{Ũ}1_3$ $\text{Ũ}1_4$ $\text{Ũ}1_5$] is an array of approximated pixels of C1

$P0_0/U0_0$	$P0_1/U1_0$	$P0_2$	$P0_3$	$P0_4$	$P0_5$	R0	Row0
$P1_0/U2_1$	$P1_1/U1_1$	$P1_2$	$P1_3$	$P1_4$	$P1_5$	R1	Row1
$U0_2$	$U1_2$	Target 4x4 block					
$U0_3$	$U1_3$						
$U0_4$	$U1_4$						
$U0_5$	$U1_5$						
C0	C1						
Column0	*Column1*						

And $\text{P̃r}1_N$ is modeled by the linear combination of $\text{P̃r}1_N = \sum_{i=0}^{i=k} P0_i\ W_{ri,}$ where k is the length of the predictor coefficients vector W_r for row filtering.

By calculating the minimum of squared difference, min $((P1_i - \text{P̃r}1_i)^2)$ for shifting a row of pixels to the right and min $((P1_i - \text{P̃l}1_i)^2)$ for shifting to the left, using wiener-Hopf's equation for optimum filter**,** it is possible to get the coefficients that can best approximate the shift between R1 & R0. Doing the same way for the column $\text{Ũu}1_i$ for upwards shift and $\text{Ũd}1_i$ for downward shift, we can get the coefficients that can best approximate the shift between C1 & C0. It should be noted that left/right shift for row filtering and up/down shift for column

filtering was necessary to capture local structure with a two tap filter and therefore requires that the optimization be done twice for both row wise and column wise optimization.



Calculating the minimum of the sum of the error as below

$$\min\left(\sum_{i=1}^{i=5}(P1_i - \text{P}\text{r}1_i)^2\right)=\min\left(\sum_{i=1}^{i=5}(P1_i - \sum_{n=0}^{n=k} P0_{i+n-1}\, W_n)^2\right), \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7)$$

for a right shift, where "$\text{P}\text{r}1_i$ "denote right shifted pixels of Row0

$$\min\left(\sum_{i=0}^{i=4}(P1_i - \text{P}\text{l}1_i)^2\right)=\min\left(\sum_{i=0}^{i=4}(P1_i - \sum_{n=0}^{n=k} P0_{i+n}\, W_n)^2\right), \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(8)$$

for a left shift, where "$\text{P}\text{l}1_i$ " denote left shifted pixels of Row0

In the same way, computing the minimum squared difference for upward and downward shift of the column, we would end up with four sets of coefficients for this particular example where a two tap predictor is used (K+1=2). This would derive the coefficients from the wiener equation

$$\begin{aligned}
W_{Rr} &= |R0rR0r\ |_{NxN}^{-1} * |R1rR0r\ |_{Nx1} && \text{For right shift}\\
W_{Rl} &= |R0_lR0_l\ |_{NxN}^{-1} * |R1_lR0_l\ |_{Nx1} && \text{For Left shift}\\
W_{CU} &= |C0_UC0_U\ |_{NxN}^{-1} * |C1uC0u|_{Nx1} && \text{For Upward shift}\\
W_{Cd} &= |C0_dC0_d\ |_{NxN}^{-1} * |C1_dC0_d\ |_{Nx1} && \text{For Down ward shift}
\end{aligned}$$

.....(9)

Where, $|R0rR0r|_{NxN}$ and $|R0_lR0_l|_{NxN}$ are NxN auto-correlation matrices of Row0 for a right shift and left shift analysis respectively, and; $|R1rR0r|_{Nx1}$ and $|R1_lR0_l|_{Nx1,}$ cross correlation between Row1 and the Row0 for right and left shift of a row wise shift analysis all of which derive W_{Rr} and W_{Rl} . Similarly, $|C0uC0u|_{NxN}$ and $|C0_dC0_d|_{NxN}$ are NxN auto-correlation matrices of column0 for up ward and down wards shifts; $|C1uC0u\ |_{Nx1}$ and $|C1_dC0_d|_{Nx1}$ is the cross correlation between Column1 and the Column0 in the column wise analysis which derives W_{CU} & W_{Cd} . In short, pixels in R1 & C1 will be our sets of desired pixels that will be approximated by using the optimal filters on pixels in R0 & C0 respectively. Note that the wider the search area, the less correlated the pixels are. It was therefore important to limit the correlation distance. For this case, N is equal to two.

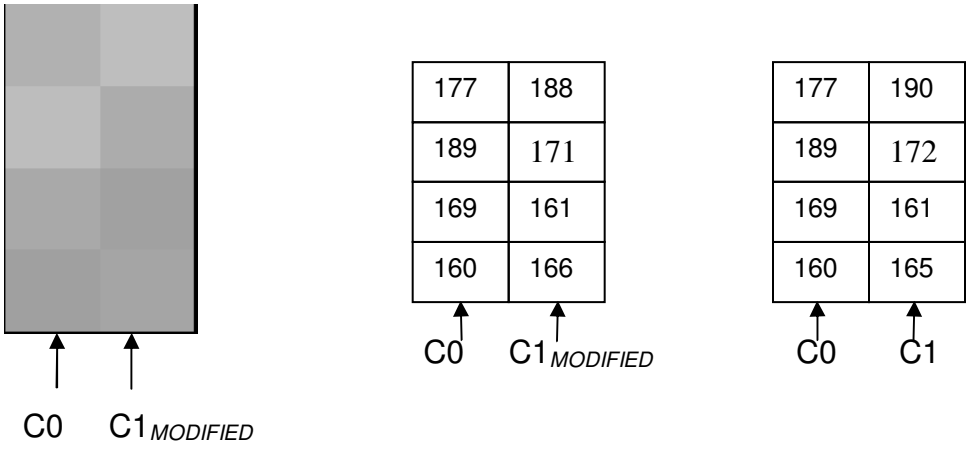
Once the coefficients, which carry information regarding structure orientation (or "structural shift" to be more precise) between rows and columns, are derived the next task would be to decide which coefficients among the four, W_{Rr} , W_{Rl}, W_{CU} & W_{Cd}, best estimates the structure. By taking the lesser of the SSD as shown below between R1 & Ř1 and the SSD between C1 & Č2, it is possible to make the decision of which coefficient to pick to make the prediction of the entire 4x4 block. The sum of squared difference (SSD) simply goes by

$$
\begin{aligned}
\text{SSD\_right shift} &= \sum_{i=0}^{i=n} (P1_i - \text{Pr}1_i)^2 \\
\text{SSD\_left shift} &= \sum_{i=0}^{i=n} (P1_i - \text{Pl}1_i)^2 \\
\text{SSD\_upward shift} &= \sum_{i=0}^{i=n} (U1_i - \text{U}_u 1_i)^2 \\
\text{SSD\_downward shift} &= \sum_{i=0}^{i=n} (U1_i - \text{U}_d 1_i)^2
\end{aligned}
\qquad \text{.....(10)}
$$

Thus the minimum of, *SSD_right shift, SSD_left shift, SSD_upward shift, SSD_downward shift*, shall decide which predictor coefficient to pick up and which direction to shift. Based on the chosen coefficient, the prediction shall be carried on starting from the immediate previously reconstructed row or column sequentially in the current block by filtering consecutive rows or consecutive columns to the target block.

Figure4.3: Demonstration of new prediction scheme based on filtering

Using the target 4x4 block from the picture above as an example, it can be shown that with a two tap prediction filter, it is possible to predict the target block. Taking the above 4x4 target block, an approximation of the pixels in R1 shall be derived based on the optimal filter coefficient generated, and in the same way, an approximation of C1 shall also be generated. In this particular 4x4 block case, W_{CU} was found to be [0.0986, 0.9047].



Therefore the lowest SSD in this case is $C1_i - C1_{MODIFIED} = 6$

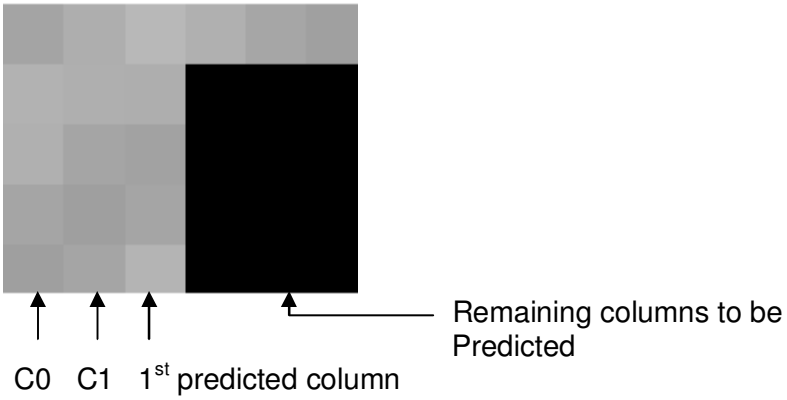


Figure4.4*:* prediction of the first row in the 4x4 target block

For this particular target block, the predictor coefficients W_{CU} seems to have given the best approximation of the pixels in C1 in terms of least SSD compared to the remaining three sets of filters. Therefore, the target block shall be predicted sequentially column by column starting by filtering C1 using predictor coefficients W_{CU} to predict the first column which in turn is filtered to approximate the second column of the target block and so on. In this manner, the whole frame in Figure4.1 can be re-constructed by 4x4 blocks predicted the same way as the above. You can see below a frame constructed from 4x4 blocks predicted using two tap prediction filters from the neighbouring pixels.

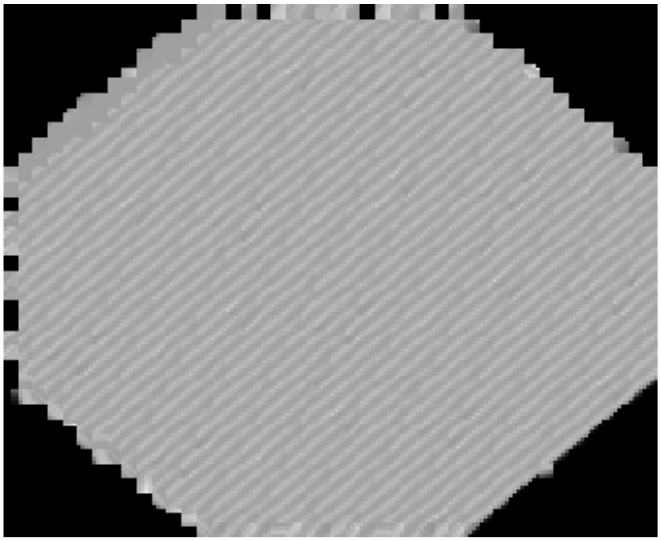


Figure4.5: Predicted picture using two predictor coefficients

When applying the above predictor algorithm in the H.264, the reconstructed neighbouring pixels are used to derive the predictor coefficients and generate the new 4x4 target blocks. In other words R1, R2, C1&C2 shall consist of reconstructed pixels. Below is the prediction from H.264 intra 4x4 prediction compared with the prediction from a two tap local structure filter. The prediction error is then encoded using H.264 transform and quantization for both cases. It shall be noted that the predicted frame produced by the two tap filter uses no side information for the intra prediction.

Figure4.6.a*:* standard predicted frame using all nine modes.(QP 22,PSNR=29.07dB)

Figure 4.6.b: A frame produced using two tap prediction filter.(QP 22,PSNR =22.90dB)

4.2.1 Improving prediction of edges

It can be seen from the predicted frame above (e.g. marked regions in Figure4.5b) that areas on edges have a very high prediction error and considerable noise. In taking two immediate rows or two columns for the estimation, there is a shortcoming to it that it does not take enough picture information to make accurate enough estimation of the local structure. In addition to this, taking the SSD of only one row and its approximate or one column and its approximate may not include structures which are not passing through the row or column and thus potentially giving a misleading result. Another reason for bad prediction is the short filter deployed. Therefore, in order to increase the accuracy of prediction and reduce the prediction noise three majors were taken,

a. Include more picture statistics in the analysis of local structure estimation.

b. Use a three tap filter,

c. Allow optimization of two free parameters by restricting the sum of the three filter coefficients to sum up to one (i.e., [a, 1-(a+b), b]) in order to avoid too much DC shift within a block when doing sequential row by row filtering or column by column filtering.

More picture statistics is added by including two more rows/columns in the optimization. The predictor coefficients can be derived by the least squares estimator which is set to minimize the square of total sum of the error between rows and their approximated versions ($R1_{modified,}$ $R2_{modified,}$ and $R3_{modified}$), however, note that only two parameters are optimized . Let W_r be vector of coefficients , $[a_r,1-(a_r+b_r),b_r]$, which estimates the shift between rows, and W_c , $[a_c,1-(a_c+b_c),b_c]$, be an estimator of the shift between columns, then pixels in R1,R2,R3 can be approximated from immediate pixels in the above, R0,R1,R2 respectively as shown below

$O0_0$	$O0_1$	$O0_1$	$O0_3$
$O1_0$	$O1_1$	$O1_2$	$O1_3$
$O2_0$	$O2_1$	$O2_2$	$O2_3$
$O3_0$	$O3_1$	$O3_2$	$O3_3$

Original 4x4 block

$P0_0/U0_0$	$P0_1/U1_0$	$P0_2/U2_0$	$P0_3/U3_0$	$P0_4$	$P0_5$	$P0_6$	$P0_7$	R_0
$P1_0/U01$	$P1_1/U1_1$	$P1_2/U2_1$	$P1_3/U3_1$	$P1_4$	$P1_5$	$P1_6$	$P1_7$	R_1
$P2_0/U0_2$	$P2_1/U1_2$	$P2_2/U2_2$	$P2_3/U3_2$	$P2_4$	$P2_5$	$P2_6$	$P2_7$	R_2
$P3/U0_3$	$P3_1/U1_3$	$P3_2/U2_3$	$P3_3/U3_3$	$P3_4$	$P3_5$	$P3_6$	$P3_7$	R_3
$U0_4$	$U1_4$	$U2_4$	$U3_4$	Target 4x4 block				
$U0_5$	$U1_5$	$U2_5$	$U3_5$					
$U0_6$	$U1_6$	$U2_6$	$U3_6$					
$U0_7$	$U1_7$	$U2_7$	$U3_7$					
C0	C1	C2	C3					

$Eh1 = R1 - R1_{modified}$
$Eh2 = R2 - R2_{modified}$
$Eh3 = R3 - R3_{modified}$

$Ev1 = C1 - C1_{modified}$
$Ev2 = C2 - C2_{modified}$
$Ev3 = C3 - C3_{modified}$

$$R1_{modified} = \left[\sum_{i=0}^{i=k} P0_i W_i \quad \sum_{i=0}^{i=k} P0_{i+1} W_i \quad \sum_{i=0}^{i=k} P0_{i+2} W_i \quad \ldots\ldots\ldots \quad \sum_{i=0}^{i=k} P0_{i+n} W_i \right]$$

$$R2_{modified} = \left[\sum_{i=0}^{i=k} P1_i W_i \quad \sum_{i=0}^{i=k} P1_{i+1} W_i \quad \sum_{i=0}^{i=k} P1_{i+2} W_i \quad \ldots\ldots\ldots \quad \sum_{i=0}^{i=k} P1_{i+n} W_i \right]$$

$$R3_{modified} = \left[\sum_{i=0}^{i=k} P2_i W_i \quad \sum_{i=0}^{i=k} P2_{i+1} W_i \quad \sum_{i=0}^{i=k} P2_{i+2} W_i \quad \ldots\ldots\ldots \quad \sum_{i=0}^{i=k} P2_{i+n} W_i \right]$$

In the same manner the predicted columns

$$C1_{modified} = \left[\sum_{i=0}^{i=k} U0_i W_i \quad \sum_{i=0}^{i=k} U0_{i+1} W_i \quad \sum_{i=0}^{i=k} U0_{i+2} W_i \quad \ldots\ldots\ldots \quad \sum_{i=0}^{i=k} U0_{i+n} W_i \right]$$

$$C2_{modified} = \left[\sum_{i=0}^{i=k} U1_i W_i \quad \sum_{i=0}^{i=k} PU_{i+1} W_i \quad \sum_{i=0}^{i=k} U1_{i+2} W_i \quad \ldots\ldots\ldots \quad \sum_{i=0}^{i=k} U1_{i+n} W_i \right]$$

$$C3_{modified} = \left[\sum_{i=0}^{i=k} U2_i W_i \quad \sum_{i=0}^{i=k} PU_{i+1} W_i \quad \sum_{i=0}^{i=k} U2_{i+2} W_i \quad \ldots\ldots\ldots \quad \sum_{i=0}^{i=k} U2_{i+n} W_i \right]$$

…………………………………………………. (10)

where k+1 is coefficient length and $R1_{modified}$, $R2_{modified}$, $R2_{modified}$ are arrays containing approximated pixels of rows R1, R2, and R3 respectively, which goes the same for $C1_{modified}$, $C2_{modified}$ & $C3_{modified}$ containing approximated pixels of C1,C2 and C3. Computation of the predictor coefficients therefore starts by setting coefficients that would

minimized sum of the mean squared error between the row and their approximated versions and the columns and their approximated versions as below,

$$Er_{min-1} = \min ((Eh1)^2 = (P1_{(n+1)} - \sum_{i=0}^{i=k} P0_{(n+i)} W_{ri})^2 \text{ for the row wise shift modeling}$$

and the over all minimum would be evaluated as

$$E_{min\_row} = \sum_{i=1}^{i=3} (Er_{min-i})$$

$$Ec_{min-1} = \min ((Ev1)^2 = (U1_{(n+1)} - \sum_{i=0}^{i=k} U0_{(n+i)} W_{ri})^2 \text{ for the column wise shift modeling}$$

and the over all minimum would be evaluated as

$$E_{min\_col} = \sum_{i=1}^{i=3} (Ec_{min-i})$$

..... (11)

where Er_{min-1} & Ec_{min-1} are least squared errors of the second row and its approximation; the second column and its approximation which would also be done for the subsequence rows and columns and finally take the over all sum of the least squared errors form each rows and columns for generation of optimal coefficients. In this way auto correlation and cross correlation computation are made to include larger picture area by trying to minimize the squared error between rows and between columns in the block above and the block to the left respectively.

$$R_M = |R_0 R_0|_{NxN} + |R_1 R_1|_{NxN} + |R_2 R_2|_{NxN}$$
$$C_M = |C_0 C_0|_{NxN} + |C_1 C_1|_{NxN} + |C_2 C_2|_{NxN} \quad \text{..... (12)}$$

where R_M and C_M are 2x2 matrices (since only two parameters are optimized) made by summing the auto-correlations of the rows R_0, R_1, R_2 and C_0, C_1, C_2 respectively . In the same manner, the cross correlation would be

$$R_v = |R_1 R_0|_{Nx1} + |R_2 R_1|_{Nx1} + |R_3 R_2|_{Nx1}$$
$$C_v = |C_1 C_0|_{Nx1} + |C_2 C_1|_{Nx1} + |C_3 C_2|_{Nx1} \quad \text{..... (13)}$$

R_v and C_V, are cross correlation vectors obtained by summing the cross correlations between each consecutive rows and each columns respectively. The predictor coefficients are then derived from the wiener-Hopf equation as,

$$W'_r = R_M^{-1} * R_v$$
$$W'_C = C_M^{-1} * C_v \quad \text{..... (14)}$$

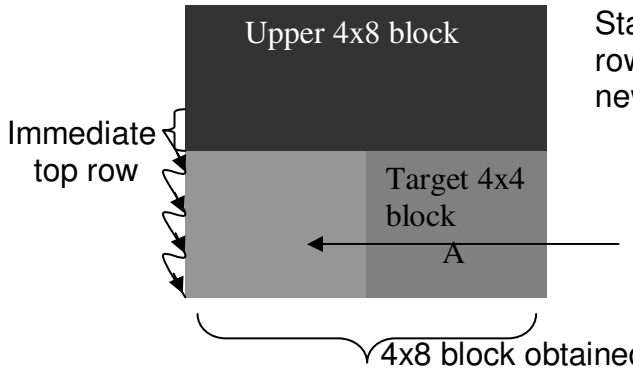
where W'_r and W'_C are vectors of two optimal parameter $[a_r, b_r]$ and $[a_C, b_C]$. The over all predictor filters for both row and column would then be $W_r = [a_r, 1-(a_r+b_r), b_r]$ and $W_C = [a_C, 1-(a_C+b_C), b_C]$ respectively.

It can be noted that by using a three tap filter only one analysis for the row and column case respectively is required compared to two for the two tap filter. Moreover, increasing

the filter length even further would increase the accuracy of prediction, however, that would also increase computation.

4.2.2. Optimal selection of predictor coefficient

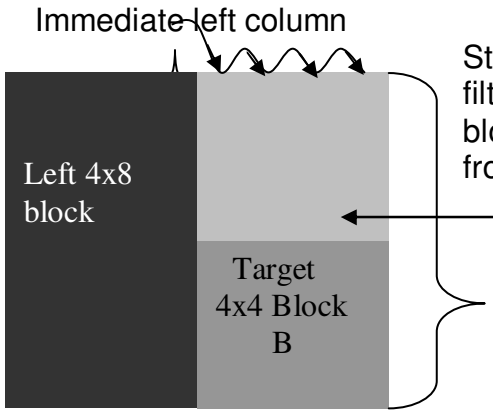
After obtaining the optimal predictor coefficients, for a row wise prediction using W_r, the prediction proceeds as shown below



Starting from the immediate top row, filtering rows consecutively downwards, a block of 4x8 newly predicted pixels is obtained from W_r.

4x4 block, Re-production of the left reconstructed block

And for the column wise prediction using Wc, the prediction would go on as below



Starting from the immediate left column, filtering columns consecutively to the right, a block of 8x4 newly predicted pixels is obtained from Wc.

4x4 block, Re-production of the top reconstructed block.

In this case, for three tapped predictor, we have two predictions for the same target block, A & B, produced by Wr & Wc respectively and for two tapped predictor, we would have four predictions. Note that along with the target block, the left and top reconstructed blocks are also having a re-production using the predictor filters Wr and Wc respectively. It was always necessary to bear in mind what information the decoder shall have available to make an identical decision as in the encoder.

Selection of prediction based on neighbouring reconstructed 4x4 block

After obtaining the predictor coefficients, Wr and Wc for predictor size "l+1", and generating the prediction from both sets of coefficients, it would be necessary to make an optimal decision on which prediction to pick up. The decoder only have neighbouring reconstructed blocks to make this decision from, therefore, the computation would be carried on as below,

$$\text{SSDr} = \sum_{n=0}^{n=3} \left(\sum_{i=0}^{i=3} \left(Ui_n - \sum_{k=0}^{k=l} Pn_{i+k} * Wr_k \right)^2 \right)$$

..... (15)

$$\text{SSDc} = \sum_{n=0}^{n=3} \left(\sum_{i=0}^{i=3} \left(Pi_n - \sum_{k=0}^{k=l} Un_{i+k} * Wc_k \right)^2 \right)$$

where Ui_n is previously predicted and reconstructed pixel to the left of the target block, *and* Pi_n previously predicted and reconstructed pixel at the top of the target block. This would result in SSD computation between the left reconstructed block and its re-produced version and the SSD between the Top block and its approximated version, the lesser of the two SSDs shall point to the predictor coefficient and the right target block which gives a more accurate approximation of the area. Standard H.264 uses original picture frame as a reference in the encoder to decide which mode of prediction best achieves higher quality in the rate distortion optimization and then sends this mode to the decoder as side information. In order to compare the block predicted by the new method with that produced by one standard intra prediction modes, it was fair to modify the RD computation and turn the constraining factor off so that the mode selection would only regard minimization the square of prediction error, i.e in equation (4), lambda is to zero.

Min (J) = Min (D)

This is because at this point the newly predicted block considers distortion only for generation of predictor coefficients, and therefore, comparison with the standard prediction should be in an identical state. Below is prediction of a frame from the foreman sequence again,

Figure4.7.a: standard predicted frame using all nine modes with full rate distortion.(QP 22,PSNR=29.1)

Figure4.7.b: standard predicted frame using all nine modes with full rate distortion but lambda set to zero (non-constrained optimization).(QP 22, PSNR=30.04)

Figure4.7.c:A frame produced using the above method with three tap predictor (QP22PSNR=26.48dB)

As can be seen by comparison from Figure4.5b, the prediction in Figure4.6.c has been improved to 26.48dB for three tap predictor with two free parameters. No information is required to be sent to the decoder regarding the choice of predictor filter since the predictor is capable of determining that from the neighbouring reconstructed blocks the same way as in the encoder.

Selection of prediction based on the original 4x4 block

As mentioned before, when calculating the SSD to select between predictors Wr and Wc the neighbouring reconstructed blocks were used. However, an alternative option is to send the decoder side information regarding which analysis to use for optimal filter generation and use the original reference block to compute the SSD and make the choice between Wr and Wc , which would improve the quality of the predicted frame.

$$\text{SSDr} = \sum_{n=0}^{n=3} \left(\sum_{i=0}^{i=3} (On_i - Rn_i)^2 \right)$$

$$\text{SSDc} = \sum_{n=0}^{n=3} \left(\sum_{i=0}^{i=3} (On_i - Cn_i)^2 \right) \qquad \text{..... (16)}$$

where On_i is a pixel from the original block and Rn_i & Cn_i are pixels from the newly predicted target block using the row and column (Wr & Wc)analysis respectively.

Figure4.8:A frame produced using three tap predictor using the original target block as reference for filter selection (QP 22,PSNR =27.68dB)

4.3 Prediction of border pixels

The order of prediction inside a 16x16 macro block is shown on the sketch below. A macroblock is divided into four 8x8 sub macroblocks (BlockNr 0,BlockNr 1, BlockNr 2, BlockNr3) and each 8x8 block is further divided into four 4x4 blocks(0,1,2,3). Intra 4x4 prediction therefore follows the sequence of block numbers 0,1,2,3 as a prediction order.

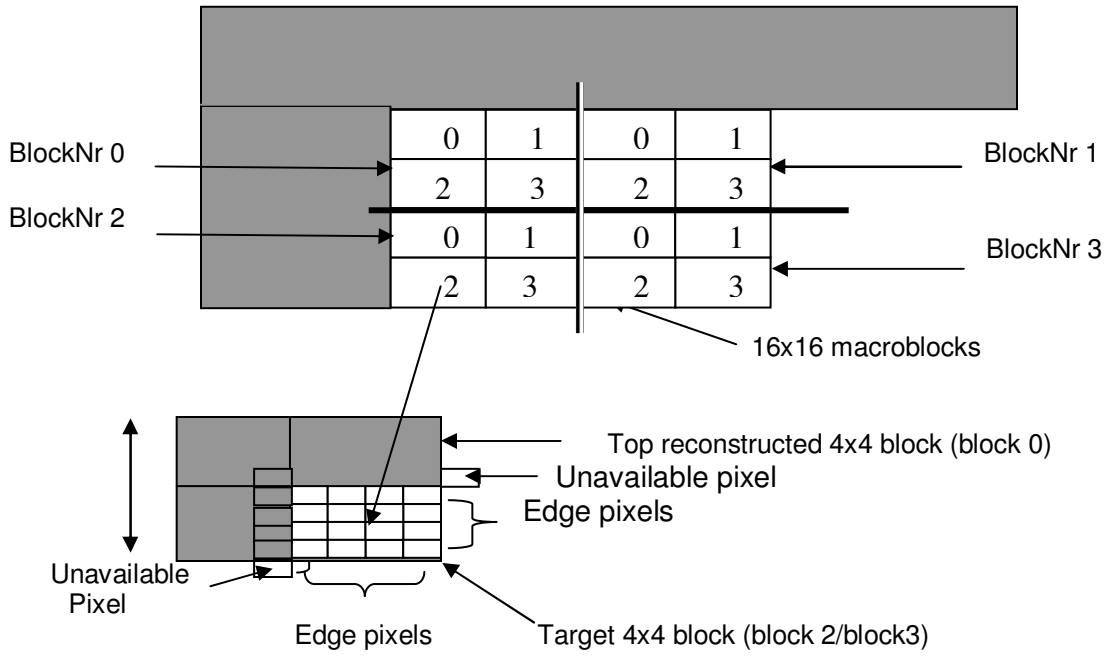
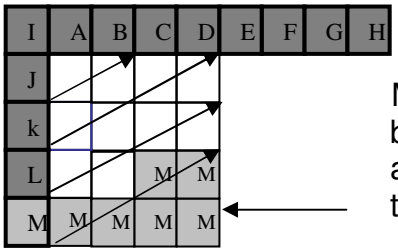


Figure4.9: Macroblock partitioning and coding order

A problem arises when trying to predict 4x4 blocks which are on edge of the macro block (*blocks 2 & 3 in BlockNr 2 &* 3). Using a two or three tap filter to predict a pixel requires that

all neighbouring pixels are available, however predicting the edge pixels poses a problem since there will be an ambiguity due to the unavailable pixel to filter. What the standard prediction normally does in this instant is to simply copy the neighbouring pixels to fill in the pixels at the edge of the 4x4 block.



Mode 8 intra-prediction: pixels in the last row of the 4x4 block and two pixels in the row above are all identical and are copied from the last pixel in the left column of the target block (M).

Figure4.10: Intra pred. mode 8

This ambiguous prediction of pixels brings about a mis-alignment in the structure going horizontally from left to right up as can be seen on the plot below. The black 4x4 blocks are there to indicate those which are predicted by mode 8 of intra prediction.

Figure4.11.a: Standard intra-predicted frame showing an area (in the rectangle) with miss alignment

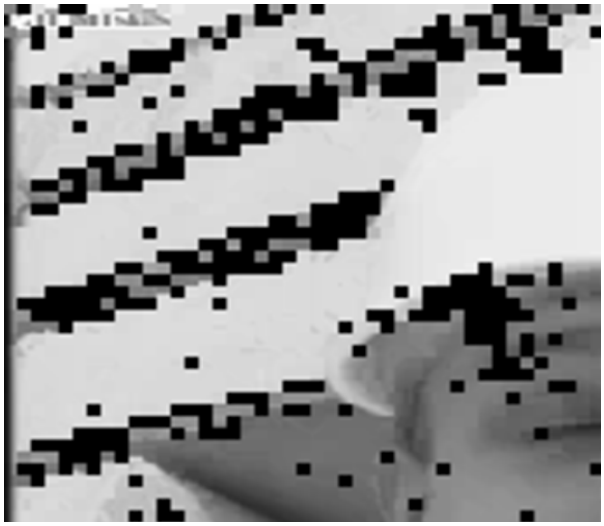


Figure4.11.b: The black 4x4 blocks showing blocks predicted by mode 8

The area below the current macroblock to be intra predicted is a black area (not yet coded and reconstructed) therefore, the unavailable pixels in Figure4.8 are temporarily copied from the pixel above and then the prediction filter would proceed predicting the pixels on the edge. Taking the MATLAB generated image below, a modified predicted frame of the original is produced. It can be seen from the figure, bottom left below, the effect of copying adjacent pixels on the predicted frame, the texture of the image is mainly due to these ambiguous pixels.

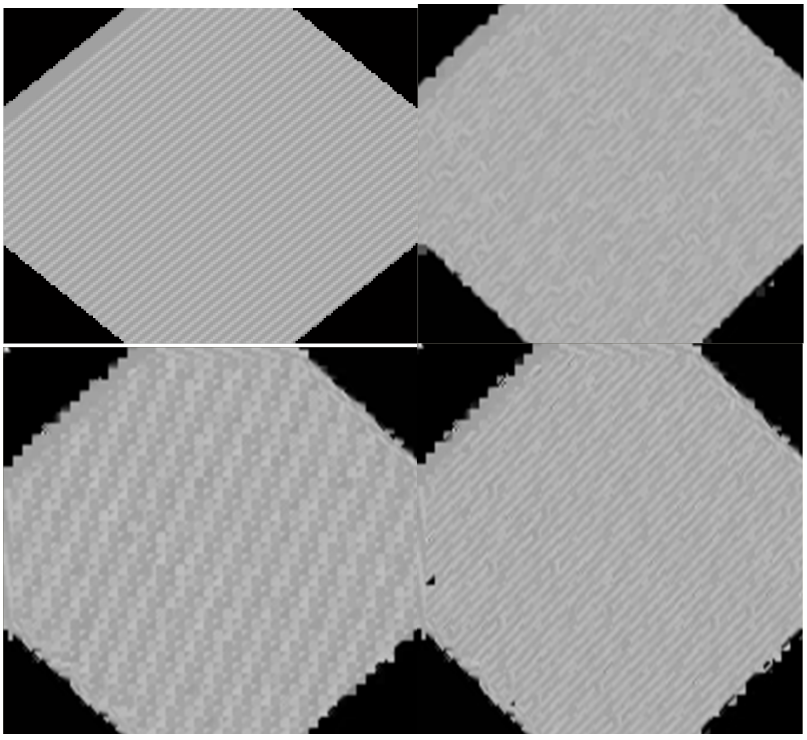
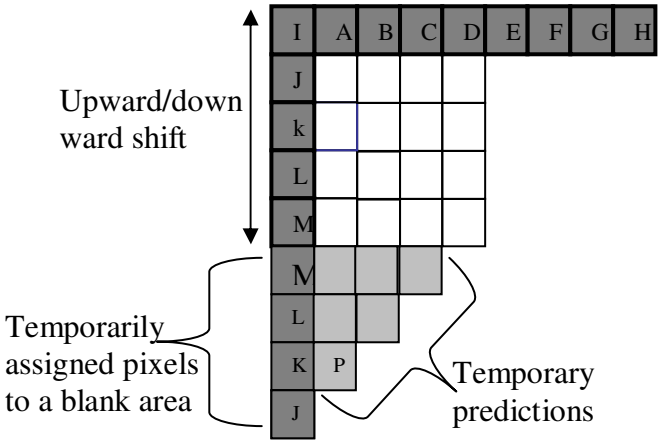


Figure4.12: (Top left) an artificial frame texture, (Top right) H.264 predicted frame, and (Bottom left) a frame predicted with three tap predictor with the above method on which the effect of prediction of edges can greatly be observed. (Bottom Right) frame shows enhancement of prediction of border pixels.



One way to get around this problem for textures with higher frequency would be instead of simply copying the pixel "M", making a mirror image of the entire column or making an exact copy of the entire column and projecting an upward temporary diagonal prediction which shall aid with the prediction of the pixels on the edge. It can be seen from Figure4.11 (bottom right) that the predicted frame is enhanced and its texture is close to the original by using pixels from the left column to temporarily take the place of the unavailable pixels to the left bottom of the current 4x4 block.

Figure4.13: Mirroring the left column

$P=L*wc_0 + K*wc_1 + J*wc_2$, where wci are the prediction coefficients for column wise prediction.

But still, for areas with low frequency pixels as in the foreman sequence, the effect of this method mirroring pixels is not visible on the frame very well as we can in Figure4.13 since not much new detail would be added between two blocks. This can be seen below.

Figure4.14: Prediction without (Left, PSNR 26.48dB) and modified predicted (Right, PSNR 26.52dB) with edge blocks using mirrored pixels, showing not much difference

It should be noted that this problem appears mainly for structure tilted from bottom left to right up, since those are the areas which require information from the blank area which is not yet coded for their prediction. And the pixel which is used to temporarily fill in the blank area could potentially cause a large prediction error which would propagate diagonally in the 4x4 block.

4.4 Prediction noise

While searching for the best estimate of a row or a column and generating the optimum coefficients, an autocorrelation matrix of the neighbouring pixels is derived. During this process, there usually is a case in which a bad behaviour of the matrix (for homogenous surfaces and sharp edges) brings about an unstable filter which leads to over amplification or attenuation of the predicted luma pixel especially occurring when there isn't enough neighbouring pixel information. A singular autocorrelation matrix implies that the surface where the matrix is derived from is homogenous surface (DC), in which case the predictor, whether it is row filter or column filter, would be set to an averaging filter ([1/3, 1/3, 1/3] for the three tap predictor) if the singularity appears during the row wise or column wise analysis respectively. However, in these areas there also appear matrices which are not completely singular but close to singularity. The slightest surface change in a homogenous area could give a relatively small determinant value which leads unstable predictor filter.

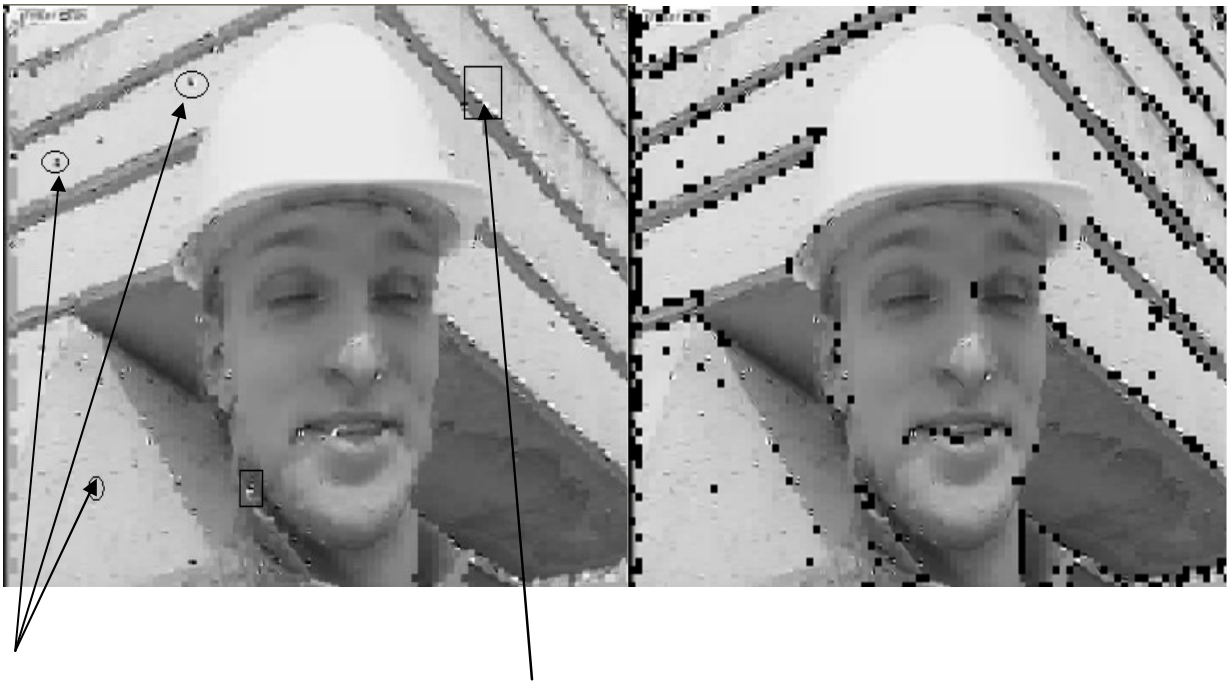


Figure4.15: Modified predicted noisy frames (Left fig), (Right fig)the right one showing localized 4x4 blocks with DC gradient greater than a threshold value.(in this case T=20, two tap filter with one free parameter)

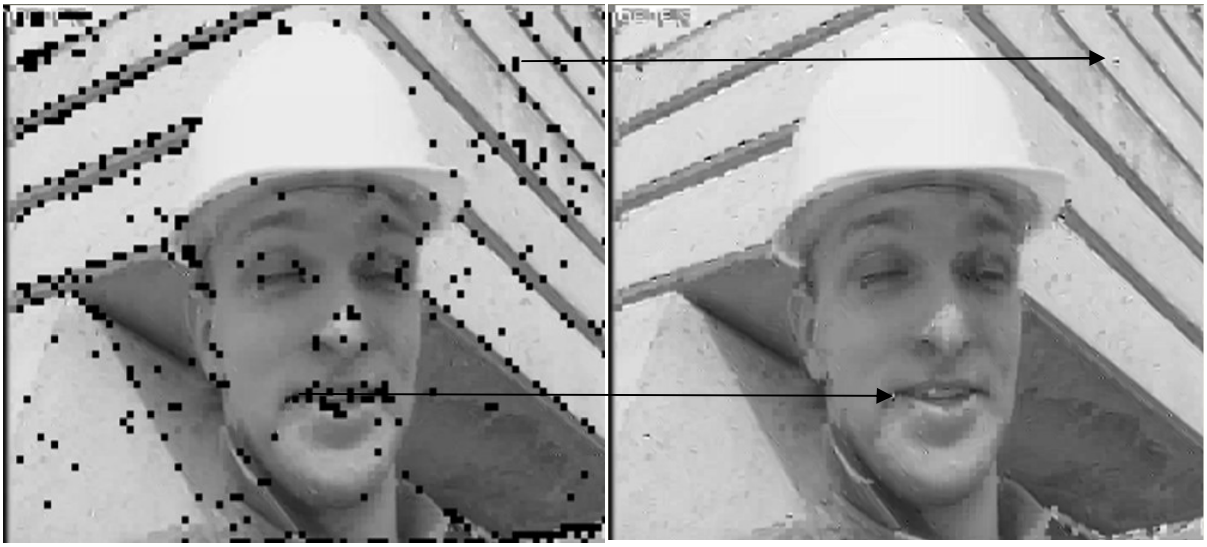


Figure4.16: Modified predicted noisy frames (Left fig), (Right fig)the right one showing localized 4x4 blocks with DC gradient greater than a threshold value.(in this case T=20, three tap filter with two free parameter)

Adding more image information, as in the previous section, aside from improving the prediction, would also decrease the noisy prediction in the frame. More over, doing the least squares optimization by restricting the filter coefficients to sum up to one also prevents DC shift in the prediction and reducing noise in the prediction greatly. However, there will still be unstable filters generated even after the above methods used, especially for two tapped predictor. It was at first necessary to localize the noisy areas in the picture

and observe the behavior of the filters producing these particular areas. One way of getting around this problem was to compute the luminance DC gradient between the newly predicted block with the new method and the same block produced using the standard H.264 intra modes in the encoder.

If | DC_new_block - DC_standard H.264 _block | > T,……...(17)

then the new block shall be discarded as is shown in Figure 4.14 a(right part).
The parameter T is an arbitrary threshold value which is set by observing how much of the noisy areas have been discarded. Below is a plot of sample filter from the print out in these areas which led to noisy prediction.

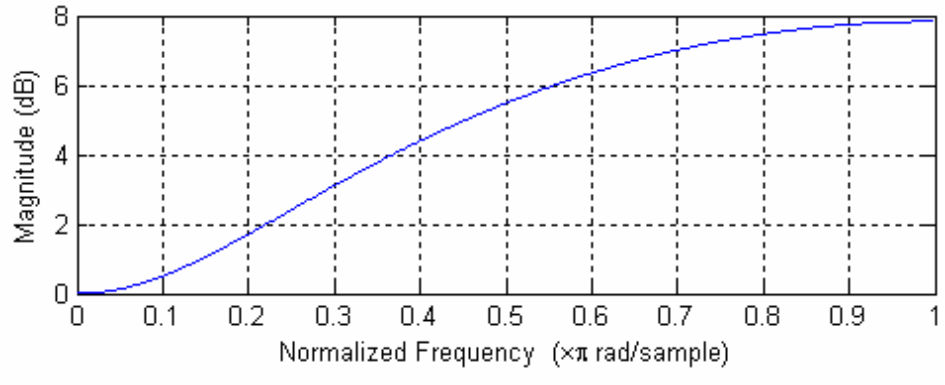


Figure4.17: Frequency response of one of the noisy predictors

As can be seen from the frequency response plot, around edges and sharp area of a picture area, such predictor filters tend to over amplify a slight pixel difference. Therefore, filtering down a block on a row by row basis or across a block column by column, the difference becomes larger and larger altering the characteristics of the entire block greatly. It was therefore necessary to set an optimal restriction for the highest filter coefficient value which was chosen to be +/- 1.2 based on the observation of predicted pixel values.

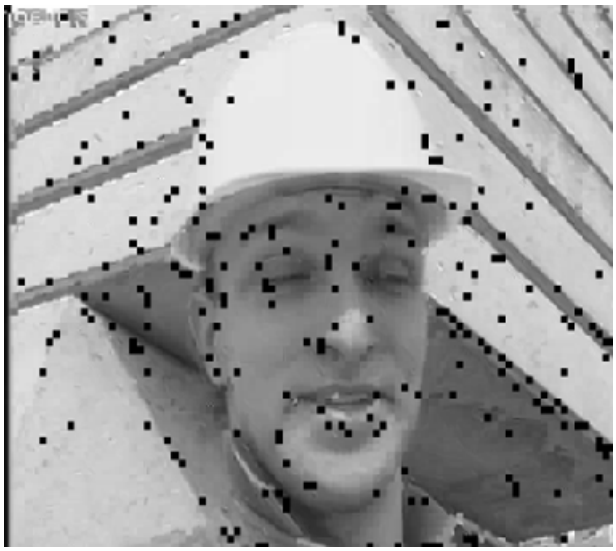


Figure4.18: The null (black) 4x4 blocks indicate blocks with predictors having one of their coefficients exceeding 1.2 (blocks to be replaced), for three tap predictor with two free parameters

Therefore, upon identifying the noisy predictors, by restraining each filter coefficients not to exceed a threshold of 1.2, as in Figure4.17 (the black 4x4 blocks), another training window is switched to generate new set of predictor coefficients which replace the bad ones. The best way at this stage was, instead of totally leaving these blocks unpredicted, it was worth it to change the direction of estimation using least squares prediction by taking in surrounding pixels as shown below in Figure4.18. In another word, instead of doing an analysis on a row by row or column by column basis, a three tap predictor (restricted to sum up to one) could be generated which minimizes the squared difference between pixel D and its approximate version. The target pixels can then be obtained from the generated optimal filter by using the surrounding three pixels for prediction.

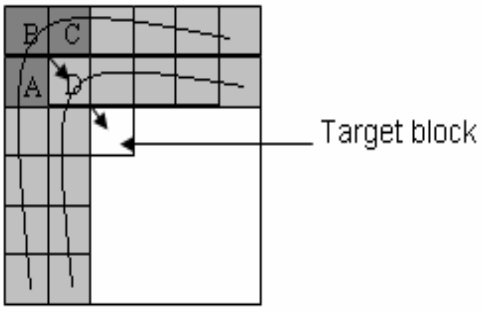


Figure4.19: prediction of pixels based on three neighbouring support pixels

Figure4.20: Noisy predictions replaced (QP 22, PSNR=26.82dB) three tap filter (with two free parameters)

4.5 Evaluation of the new block

From the performances observed in the previous section, it was decided to use three tap predictor with two free parameter (sum of the three coefficients summing up to one, [a, 1-(a+b), b]) for generation of the new block , with a back up filter as a support in case the predicted filter is noisy (Figure 4.18). It should be noted that the length of the filter could be increased for further improvement, which would in turn increase the complexity. At this stage, the new block is still not included in the rate distortion computation; therefore, it was only possible to evaluate its prediction accuracy based on its SSD comparison with that of a block produced by a standard prediction. The larger the resolution of video sequences the better quality the standard prediction has since there is a dense correlation of neighbouring pixels. Like wise, we would expect our prediction filter to perform better for higher resolution sequences and that is also the case. As is seen in section 3.1, the higher the quantization parameter, the smoother the surrounding reconstructed image area is, and thus hiding much of the surrounding structure. The interest of performance of our prediction should therefore be more for lower QPs. The graph below shows how many percent of the new 4x4 blocks produced have a distortion measured in SSD which is equal to or less than that of a standard prediction of the same 4x4 block. The 720p resolution sequence is “Crew” and the CIF sequence is “foreman”

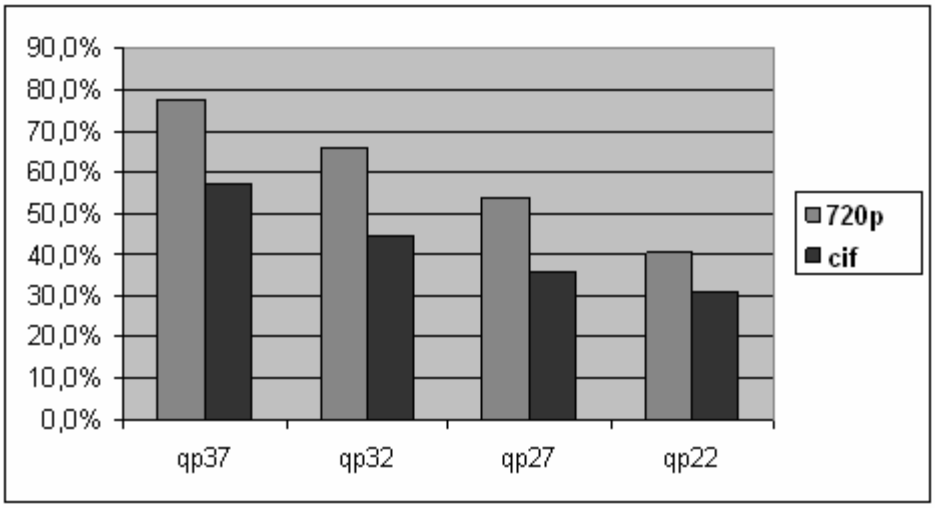


Figure21: Percentage of 4x4 new blocks with equal or less SSD distortion than using standard intra prediction

As can be seen, for higher QPs of both resolutions, the percentage that the new block has a distortion which is less or the same as that of a block predicted by any of the nine standard modes is higher. However, the outcome still deviates from the desire, that is, the block is more accurate in predicting somewhat smoother areas but it was actually desired to have a more accurate prediction in more detailed areas where there are edges. Still, it would require including the new block in the encoder structure in order to see its influence on the prediction of an entire frame. Obviously a better reconstructed block would improve the prediction of the forthcoming blocks.

5. Implementation of modified Coder structure

5.1 RD competence of the new mode for luma inter prediction

The new prediction is intended to be included as an additional 10th mode in addition to the H.264 intra prediction modes. It was important to see how much of the time the rate distortion computation favoured to use the new mode for prediction, compared to the nine standard intra prediction modes, to be able to give it an efficient mode position. Therefore, the block was initially included in the intra 4x4 block structure as “mode 9”. The new prediction would add accuracy to the over all prediction of a frame, and therefore should somewhat influence the distribution of the prediction modes. The desire is to have the new mode picked up by the rate distortion computation as often as possible with dominant percentage. Based on the observation of the distribution of modes with the additional prediction mode for sample sequences, it would be possible to modify the coder structure to try for a better rate distortion performance. Below is shown the percentage distribution of modes for QCIF, CIF and 720p type sequences, with the new mode (mode 9) influencing the distribution.

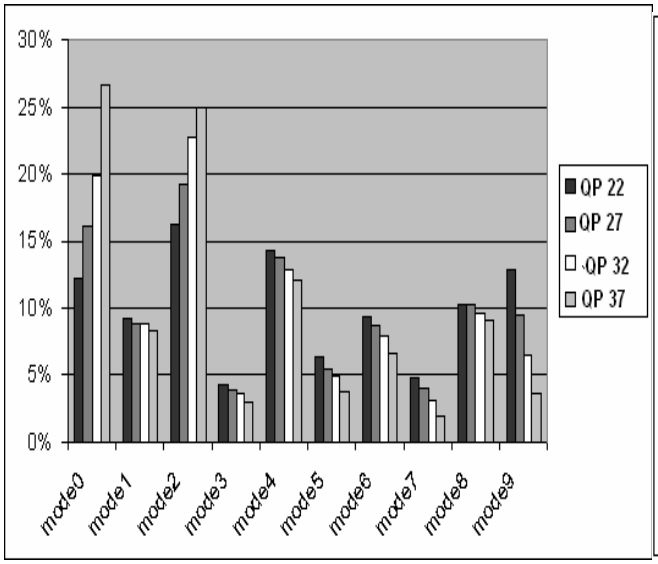


Figure5.1a: Foreman QCIF

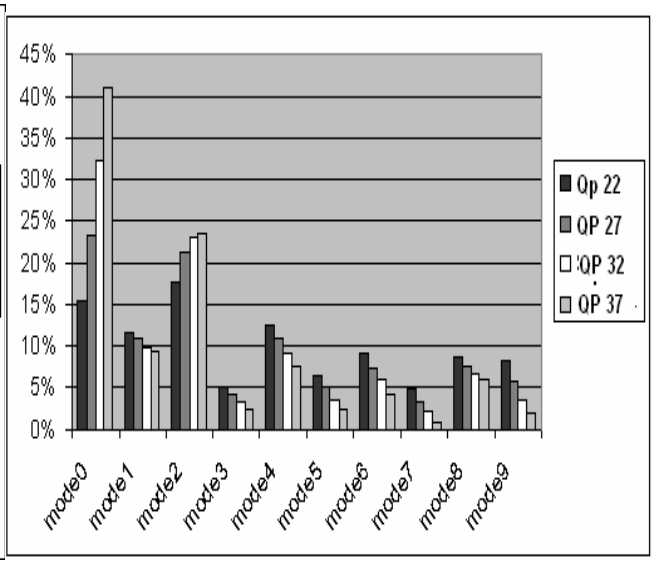


Figure5.1b: Foreman CIF

Figure5.1c: Foreman original frame

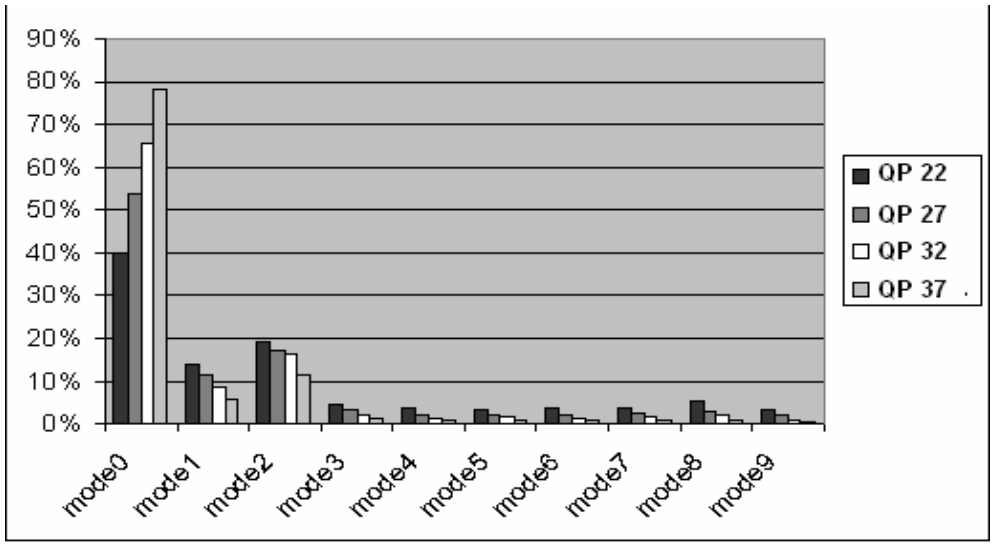


Figure5.2:- Percentage of selection of modes for a 720p sequence crew

Figure5.3: Crew 720p

In QCIF sequences the new prediction (mode 9) was favoured more often than the standard modes are, with the exception of mode 0 and mode 2; for lower quantization parameters. However, for the higher resolution frames, CIF and 720p, especially 720p, the rate distortion computation did not pick up the later modes (modes 3 to 9) including the new mode so often, even less for higher QPs. But looking at the frame above, you would notice it does actually make sense that mode 0 and 2 should dominate the selection because, most of the area in the frame is homogenous surface and therefore, mode 0 , 2 and the new mode would perform comparatively. Since mode 0 comes first, the rate distortion would hold on to it even if it comes across a mode which performs equally. This can be clearly seen in Figure5.4a by modifying the intra prediction mode structure and bringing the new mode to proceed the standard mode 0, i.e., new mode will be mode 0, and standard mode 0 will be mode 1, standard mode 1 to mode 3, and so on …, with the exception of standard mode 2 (DC) holding its original place. As can be seen from figure 5.4a, the

percentage of selection of the new mode has grown considerably high including for QCIF, CIF sequence foreman as is seen in figures 5.4b and 5.4c respectively.

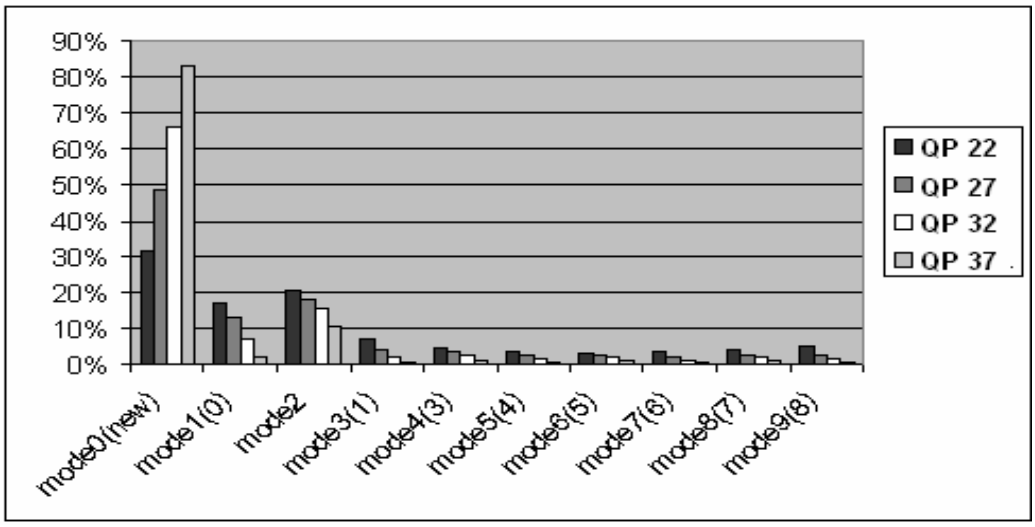


Figure5.4a: Percentage of selection of modes for a 720p sequence crew with the new Mode put on the front as mode 0.

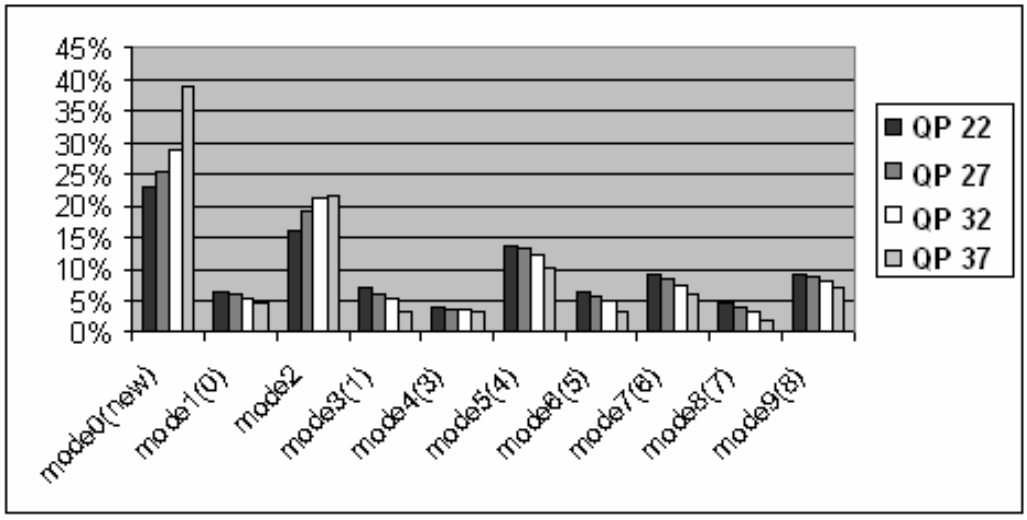


Figure5.4b: Percentage of selection of modes for a QCIF sequence foreman with the new mode put on the front as mode 0.

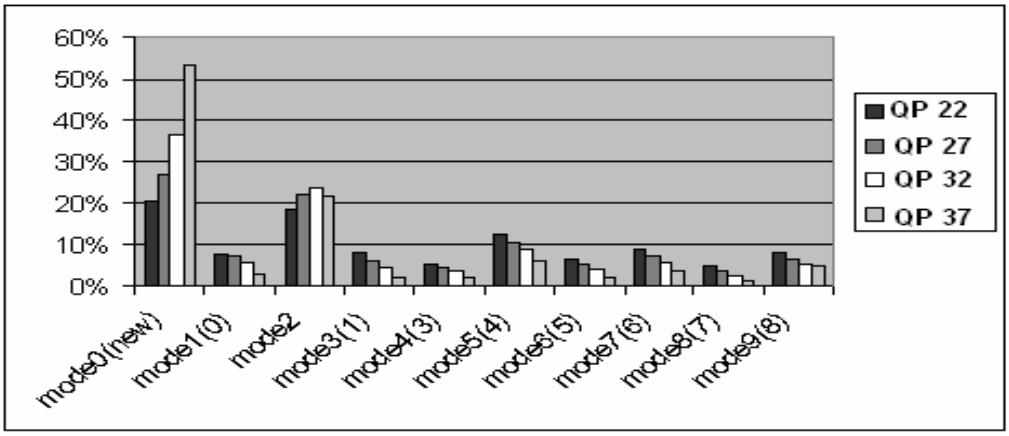


Figure5.4c: Percentage of selection of modes for a CIF sequence foreman with the new mode put on the front as mode 0.

Observe how standard mode 0 , which at the moment was switched to be mode 1 with the modified order, is hugely dominated by the new prediction, even more so for higher QPs. Another HD sequence was chosen with a bit of more varying surface on it in order to for a better evaluation of the new mode with varying structure.

Figure5.5a: A frame from a 720p sequence “City” (original)

Figure5.5b: A Standard 4x4 Intra prediction of the above frame (QP 22, 24.57dB)

Figure5.5c: The above frame predicted using 4x4 blocks produced by the new method, from neighbouring reconstructed blocks (QP 22, 22.98dB).

What is to be seen here is, how well the new blocks can compete with the standard intra prediction blocks when the sequences contain a highly detailed area. This still does not necessarily mean the new blocks performs best. It just shows that it is as good as standard modes and has a higher chance for making a more accurate and different prediction from

the standard modes in these types of sequences. It was therefore worth to see the percentage of choice for these sequences.

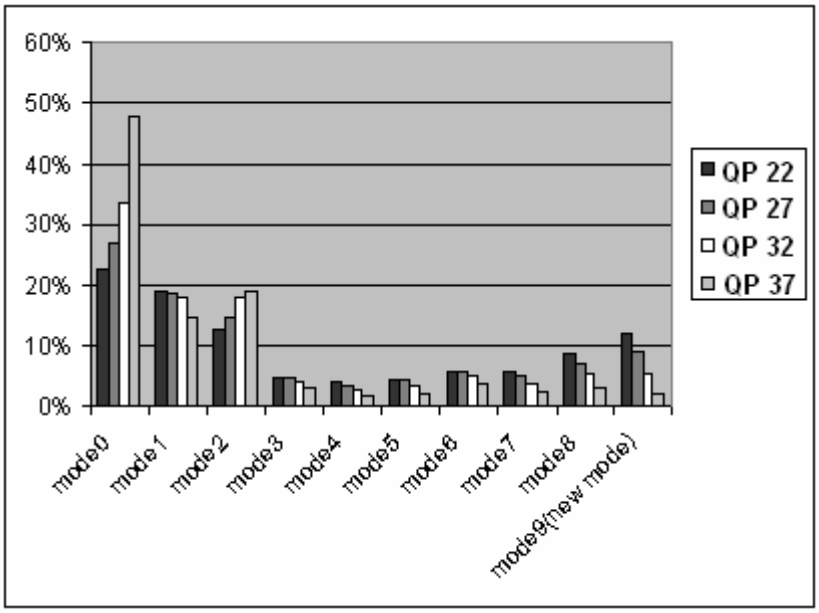


Figure5.6: Percentage of selection of the new mode for 10 frames of the above sequence. (Compare with figure 5.2)

It can be seen that the percentage that RD chooses the new mode has risen by almost 7 % when the picture area does not have much homogeneity.

5.2 Simulations and Results

In sections 5.2.1 and 5.2.2 test sequences were run with the encoder set to Intra 4x4 mode only. In section 5.2.3 both Intra 4x4 and Intra 16x16 modes are enabled. The rate distortion performance of the method studied in this thesis is tabulated as delta bit rate and delta PSNR. It shows the percentage bit savings and the average PSNR improvement. Both are measured using the Bjontegaard measurement method [10], by fitting a curve through four PSNR/rate data points for QP 22, 27, 32, and 37. The over all result includes all standard picture formats QCIF, CIF, and 720P.Test of QCIF sequences was done with every second frame dropped for 300 frames of sequences which imply that the frame rates are 15 frames/s. The tests for the CIF and 720p sequences are with 30 frames/sec and 60 frames/sec respectively. Since all improvements and measurements of interest were on the luma pixels for reasons explained in section 2.1, all PSNR measurements concern luma pixels. The selection of modes uses full rate distortion optimization computation. The results put here are based on RD comparison after modifying the signaling of standard modes to accommodate for an additional mode , which is to mean that the modified intra prediction is evaluated for bit saving and PSNR improvement against the modified signaling of modes for standard intra. As was observed in section 4, it was more beneficial to three tapped predictor with restricted coefficient optimization (coefficients forced to sum up to one) to generate the best modified predicted frame, and hence the predictor used to generate prediction in the new mode.

5.2.1 Adding one mode to standard H.264

Among the two sets of predictions, based on row and column filtering, one is chosen by SSD computation on neighbouring reconstructed blocks, as discussed in section 4. The one giving least SSD is selected. Since this selection is done in both the encoder and the decoder only the mode that indicates that the new method shall be used needs to be sent to the decoder. The adjustment performed in the coding to accommodate one additional mode had its own bit gain and increment for various sequences. Therefore, the performance of the adjustment in the coding without the new mode added is compared to standard H.264 Intra prediction.

Table5.1: RD performance of the mode signalling modification with Standard H.264 using Intra 4x4 only as reference

Sequence	No.of sequences	Resolution	Δ PSNR[dB]	Δ Rate[%]
Container	300	QCIF@15Hz	+0.0526	-0.6393%
Foreman	300	QCIF@15Hz	+0.0428	-0.5855%
Silent	300	QCIF@15Hz	+0.0480	-0.0462%
Paris	300	CIF@30Hz	+0.1734	-0.2592%
Foreman	300	CIF@30Hz	+0.0603	-0.9036%
Mobile	300	CIF@30Hz	-0.0428	+0.4236%
Tempete	300	CIF@30Hz	-0.0317	+0.3731%
BigShips	150	720p@60Hz	+0.0705	-1.9871%
ShuttleStart	150(starting from frame 300)	720p@60Hz	+0.1163	-1.8296%
City	150	720p@60Hz	+0.0907	-0.1491%
Crew	150	720p@60Hz	+0.0214	-1.3814%
Night	150	720p@60Hz	+0.0388	-0.5095%
Average		**QCIF**	**+0.0478**	**-0.4237%**
		CIF	**+0.0398**	**-0.0915%**
		720p	**+0.0675**	**-1.1713%**
		Overall	**+0.0534**	**-0.6245%**

As can be seen from the above result table, the modification that was already done before, at Ericsson, for selection of modes, to fit in one additional mode had an average gain of 0.62% in bits and a small improvement in PSNR 0.05dB, as compared to standard H.264. Therefore, it was proper to evaluate the performance gained due to the new method by comparing it against the modification that was done before as the reference, from now denoted "*modified reference*" in order to distinguish it from standard H.264. Gain in performance with the standard H.264 as a reference was also evaluated.

Therefore, the first experiment was performed by placing an additional new mode as mode 9 behind all other standard intra H.264 modes. As can be seen in Figure5.1, the percentage of selection of the new mode was very low for HD sequences and relatively higher for QCIF and CIF sequences, correspondingly, the result of RD performance as is shown in Table5.2 indicate a bit saving of -0.81% for HD sequences while a higher saving of -1.62% & -1.56% for QCIF and CIF sequences respectively with an over all average bit saving of -1.26% and +0.09dB in PSNR.

Table5.2: Performance with the new prediction as one additional mode, "mode 9"

			Against modified Reference	
Sequence	**No. of frames**	**Resolution**	**Δ PSNR[dB]**	**Δ Rate[%]**
Container	300	QCIF@15Hz	+0.0796	-0.9694%
Foreman	300	QCIF@15Hz	+0.2394	-3.1934%
Silent	300	QCIF@15Hz	+0.0481	-0.6981%
Paris	300	CIF@15Hz	+0.1885	-2.0614%
Foreman	300	CIF@30Hz	+0.1305	-1.9424%
Mobile	300	CIF@30Hz	+0.0692	-0.6873%
Tempete	260	CIF@30Hz	+0.1332	-1.5624%
BigShips	150	720p@60Hz	+0.0639	-1.0311%
ShuttleStart	150(starting from frame 300)	720p@60Hz	+0.0274	-0.2992%
City	150	720p@60Hz	+0.0914	-1.2593%
Crew	150	720p@60Hz	+0.0249	-0.3555%
Night	150	720p@60Hz	+0.086	-1.12%
Average		**QCIF**	**+0.1224**	**-1.6203%**
		CIF	**+0.1303**	**-1.5634%**
		720p	**+0.0587**	**-0.8130%**
		Overall	**+0.0985**	**-1.2650%**

The next experiment was by bringing the new mode to be mode 0 and shifting the entire standard modes one step; it was looked at previously (Figure5.4) that the percentage of selection of the new mode has increased considerably by placing it as mode 0. However, looking at how the percentage of selection of the vertical mode (changed to mode 1 in this case) has reduced, indicating that most of the prediction from the new mode was similar to the standard Intra vertical prediction in accuracy and as is shown in Table5.3. The average bit saving showed that there was not much difference in performance by placing the new prediction scheme as mode 0 from what was gain by placing it in mode 9.

Table5.3: Performance with the new prediction at the front, i.e. mode 0

			Against modified Reference	
Sequence	**No. of frames**	**Resolution**	**Δ PSNR[dB]**	**Δ Rate[%]**
Container	300	QCIF@15Hz	+0.1963	-2.3890%
Foreman	300	QCIF@15Hz	+0.1531	-2.0633%
Silent	300	QCIF@15Hz	+0.0454	-0.6557%
Paris	300	CIF@15Hz	+0.1830	-2.0106%
Foreman	300	CIF@30Hz	+0.0857	-1.2795%
Mobile	300	CIF@30Hz	+0.0609	-0.6038%
Tempete	260	CIF@30Hz	+0.1312	-1.5446%
BigShips	150	720p@60Hz	+0.0572	-0.9111%
ShuttleStart	150(starting from frame 300)	720p@60Hz	+0.1607	-1.8405%
City	150	720p@60Hz	+0.0809	-1.1348%
Crew	150	720p@60Hz	+0.0338	-0.3228%
Night	150	720p@60Hz	+0.0432	-0.5624%
Average		**QCIF**	**+0.1316**	**-1.7027%**
		CIF	**+0.1152**	**-1.3596%**
		720p	**+0.0752**	**-0.9543%**
		Overall	**+0.1026**	**-1.2765%**

Having performed the above experiments, it was still desired to see the potential gain with placing the new block as mode 2 in front of standard DC prediction. It was observed (Figure5.7 below) that unlike the previous cases, the percentage of selection of the new block as mode 2 was almost the same for all QPs, indicating a uniform performance. This has especially benefited HD sequences by improving the RD performance to -2.1% in bit saving from the previously -0.95%. Table5.4 shows the corresponding RD result table for placing the new mode as mode 2 with an average gain of -2.1% and +0.21dB PSNR for all the sequences.

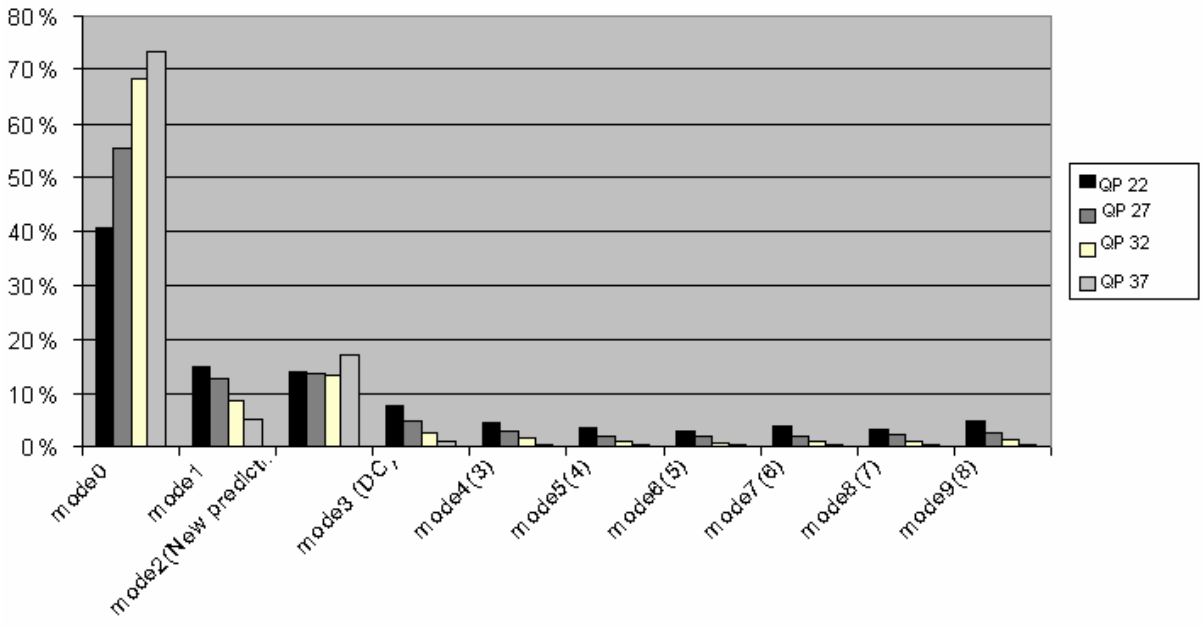


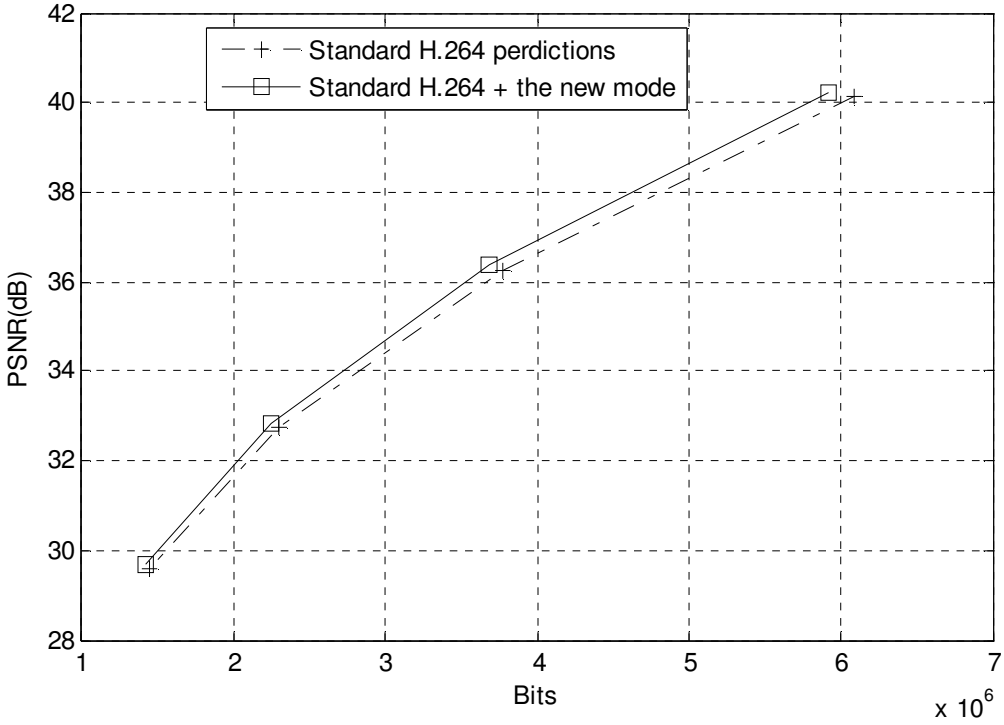
Figure5.7: Percentage of selection of the modes for 10 frames the "Crew" sequence (figure 5.3) with the new mode place in mode 2.

Table5.4: The new block taking mode 2 (DC)'s position

			Against modified Reference		Over all result Reference:- Standard H.264 Intra 4x4	
Sequence	**No. of frames**	**Resolution**	**Δ PSNR[dB]**	**Δ Rate[%]**	**Δ PSNR[dB]**	**Δ Rate[%]**
Container	300	QCIF@15Hz	+0.1836	-2.2284%	+0.2361	-2.8480%
Foreman	300	QCIF@15Hz	+0.2969	-3.9621%	+0.3418	-4.5176%
Silent	300	QCIF@15Hz	+0.0830	-1.2118%	+0.0863	-1.2561%
Paris	300	CIF@15Hz	+0.2162	-2.3635%	+0.2412	-2.6142%
Foreman	300	CIF@30Hz	+0.1652	-2.4576%	+0.2262	-3.3343%
Mobile	300	CIF@30Hz	+0.0725	-0.7168%	+0.0297	-0.2963%
Tempete	260	CIF@30Hz	+0.1646	-1.9313%	+0.1332	-1.5656%
BigShips	150	720p@60Hz	+0.0960	-1.5468%	+0.2150	-3.4768%
ShuttleStart	150(starting from frame 300	720p@60Hz	+0.2850	-3.2732%	+0.4342	-5.0184%
City	150	720p@60Hz	+0.1233	-1.7144%	+0.1346	-1.8590%
Crew	150	720p@60Hz	+0.1684	-2.5612%	+0.2610	-3.8956%
Night	150	720p@60Hz	+0.1082	-1.4104%	+0.1622	-2.1110%
Average		**QCIF**	**+0.1878**	**-2.4674%**	**+0.2214**	**-2.8739%**
		CIF	**+0.1546**	**-1.8673%**	**+0.1576**	**-1.9526%**
		720p	**+0.1562**	**-2.1012%**	**+0.2414**	**-3.2722%**
		Overall	**+0.1636**	**-2.1148%**	**+0.2085**	**-2.7327%**

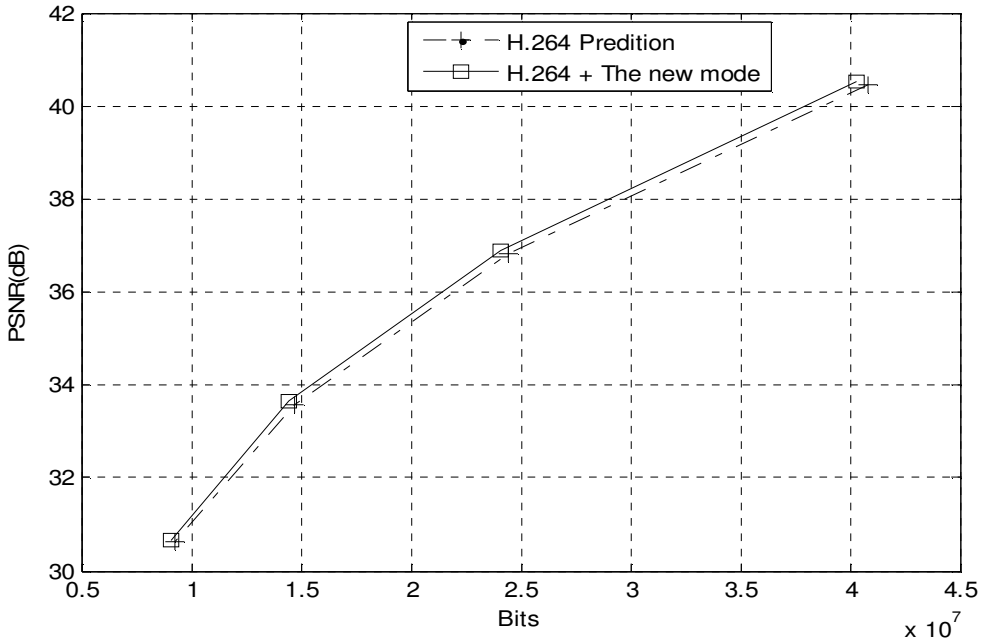
Observation of performance from RD Curve

Below is the RD curve of sample sequences from method C, “putting the new mode as mode 2”



Foreman QCIF

Figure5.8.a: RD curve of the standard H.264 prediction against the improved scheme with the new mode place in mode 2



Foreman (CIF)

Figure5.8.b: RD curve of the standard H.264 prediction against the improved scheme with the new mode place in mode 2

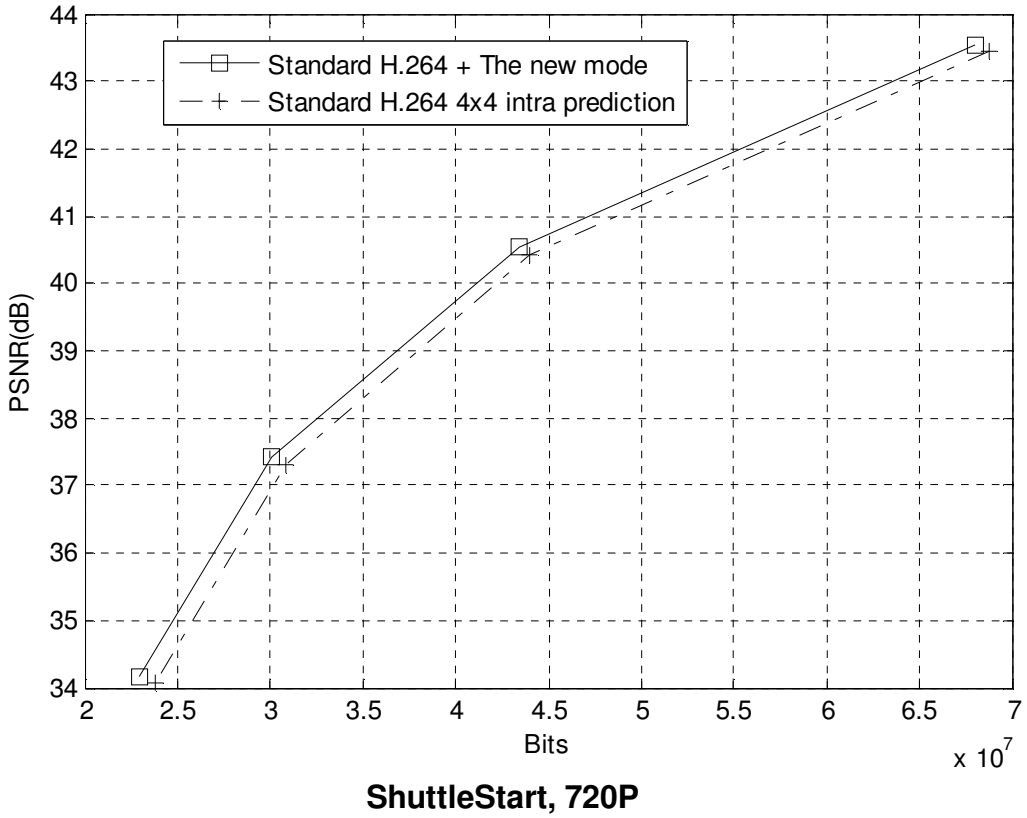


Figure5.8.c: RD curve of the standard H.264 prediction against the improved scheme with the new mode place in mode 2

Much of the gain due to the new mode was for higher bit rates (low QP values), when the neighbouring reconstruction has less smoothening. However there are gains for all QP values.

5.2.2 Adding two modes to standard H.264

In this case, it was desired to include the selection of the new predictions, among the ones produced by the row filter or column filter, in the rate distortion optimization and let the choice to be based on a comparison with the target original block. This would require having information sent to the decoder regarding which one that should be used for the current block. Therefore, two additional modes will be added to the standard intra prediction modes. One mode would take the prediction from the row filters, and another one taking the prediction from the column filters. From the experience on the previous experiment, the new prediction mode worked best when it was put as mode 2. Performance comparison using percentage of choice showed that column filtered blocks had a higher performance in all QPs than row filtered blocks, as could be seen from a sample distribution graph in Figure5.9 below (mode 9 for column filtering, and mode 10 for row filtering). The column filter was therefore given mode 2. The row filter was given mode 4 in order to not push standard Intra DC mode's placement too far from its original position.

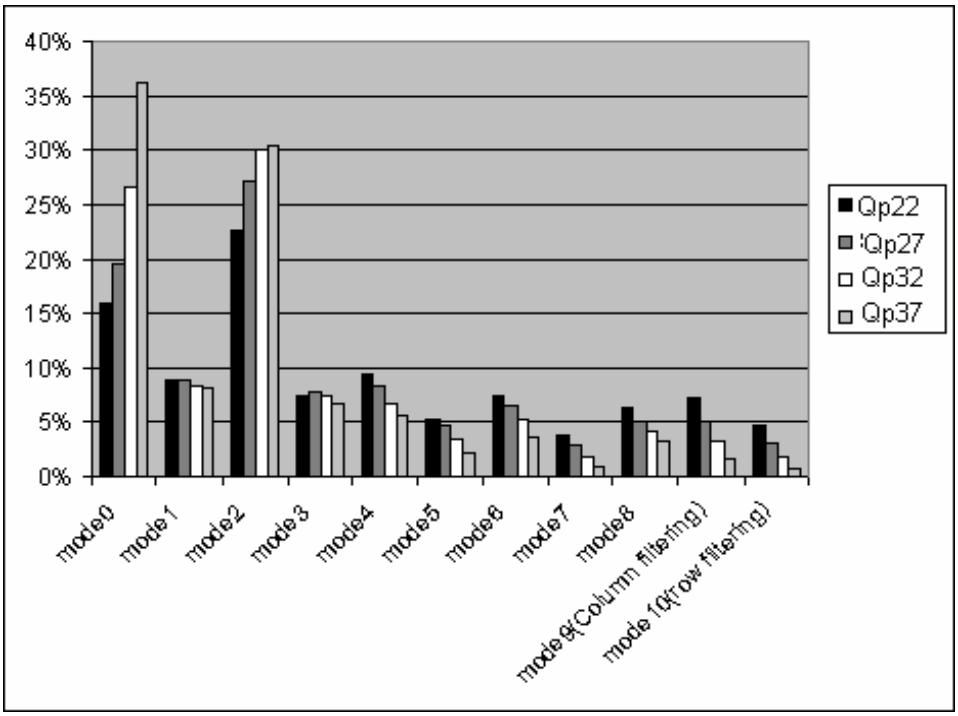


Figure5.9: Percentage of selection of modes for "foreman CIF" sequence, with additional mode 9(column) and 10 (row filtering)

Table5.5: Performance with new modes fitted in mode 2 and mode 4

			Against modified Reference		Over all result Reference:- Standard H.264 Intra 4x4	
Sequence	**No. of frames**	**Resolution**	**Δ PSNR[dB]**	**Δ Rate[%]**	**Δ PSNR[dB]**	**Δ Rate[%]**
Container	300	QCIF@15Hz	+0.1539	-1.8833%	+0.2616	-3.149%
Foreman	300	QCIF@15Hz	+0.3029	-4.0658%	+0.357	-4.7168%
Silent	300	QCIF@15Hz	+0.0733	-1.0728%	+0.0325	0.4902%
Paris	300	CIF@15Hz	+0.2145	-2.3555%	+0.236	-2.5699%
Foreman	300	CIF@30Hz	+0.169	-2.5414%	+0.2571	-3.8089%
Mobile	300	CIF@30Hz	+0.0857	-0.848%	-0.0023	+0.0189%
Tempete	260	CIF@30Hz	+0.1896	-2.2216%	+0.1182	-1.3958%
BigShips	150	720p@60Hz	+0.0956	-1.5599%	+0.2469	-4.0049%
ShuttleStart	150(starting from frame 300)	720p@60Hz	+0.1367	-1.5622%	+0.5232	-6.1106%
City	150	720p@60Hz	+0.1243	-1.7389%	+0.1510	-2.0907%
Crew	150	720p@60Hz	+0.1375	-1.9262%	+0.3413	-5.0804%
Night	150	720p@60Hz	+0.1063	-1.397%	+0.1622	-2.1110%
Average		**QCIF**	**+0.1767**	**-2.3406%**	**+0.2170**	**-2.4585%**
		CIF	**+0.1647**	**-1.9916%**	**+0.1523**	**-1.9389%**
		720p	**+0.1201**	**-1.6368%**	**+ 0.2849**	**-3.8795%**
		Overall	**+0.1491**	**-1.9311%**	**+0.2237**	**-2.8774%**

5.2.3 Enabling Intra 16x16

To complete the evaluation of the new algorithm, intra 16x16 is also enabled in this experiment. Having seen that a relatively higher RD performance could be gained by placing the new prediction as mode 2; that approach was also implemented here. It should be noted that, by including Intra16x16 blocks alone there would be a gain from having to work with standard intra with only 4x4 blocks. Any potential improvement on quality of the neighbouring reconstructed pixels will aid in improving the estimated filter error. Intra 16x16 blocks have the advantage of requiring fewer amounts of bits for mode indication than Intra4x4 blocks.

Table5.6: Performance with new modes fitted as mode 2 and Intra 16x16 blocks enabled

			Against modified Reference with Intra16x16	
Sequence	**No. of frames**	**Resolution**	**Δ PSNR[dB]**	**Δ Rate[%]**
Container	300	QCIF@15Hz	+0.1017	-1.3645%
Foreman	300	QCIF@15Hz	+0.1017	-3.7346%
Silent	300	QCIF@15Hz	+0.0781	-1.1521%
Paris	300	CIF@15Hz	+0.2028	-2.2806%
Foreman	300	CIF@30Hz	+0.1506	-2.4792%
Mobile	300	CIF@30Hz	+0.0606	-0.5938%
Tempete	260	CIF@30Hz	+0.1402	-1.6554%
BigShips	150	720p@60Hz	+0.0606	-1.2005%
ShuttleStart	150(starting from frame 300)	720p@60Hz	+0.0353	-0.7045%
City	150	720p@60Hz	+0.1009	-1.4795%
Crew	150	720p@60Hz	+0.0256	-0.6158%
Night	150	720p@60Hz	+0.0878	-1.2657%
Average		**QCIF**	**+0.0938**	**-2.0837%**
		CIF	**+0.1386**	**-1.7523%**
		720p	**+0.0620**	**-1.0532%**
		Overall	**+0.0955**	**-1.5438%**

As could be seen from the table, there could still be an average gain of -1.5% by including Intra 16x16 blocks compared to the modified reference. It should be noted that it is the relative performance that is being compared here, that is, having Intra 16x16 macro blocks in it improves the rate distortion performance, therefore, the performance should be checked relative to what could be gained using standard H.264 with both Intra 4x4 and Intra 16x16 blocks allowed. The relative improvement with the new mode when having intra 16x16 was not as much as with what was gained by not having Intra 16x16 blocks.

6. Complexity evaluation of algorithm

In this thesis, it was aimed to restrict the algorithm not to be overly complicated with a bit of consideration given to running time, but still never restricting any implementation ideas, i.e., it was more desired to have an improvement in coding efficiency. In the method implemented, a search area of the neighbouring block is specified to determine predictor coefficients, and in the process, for the predictor size chosen above (three tap) and by including more statistics in the search, we would have a neighbouring 4xM block of pixels (in the above scheme M = 8) to do the search from. Since it was decided to restrict the predictor coefficients to sum up to one, the actual parameters that had to be optimized and generated were two. To use the least squares algorithm involves 5 multiplications and 4 additions for computation of auto-correlation and cross correlation per pixel; and a total of 5(M-2) & 4(M-2) multiplications and additions respectively for cross correlation (2x1 vector) and auto-correlation (2x2 matrix) between two row or columns. By including more statistics, a total of the autocorrelation and the cross correlation between the rows and columns in the entire 4xM block is taken in as is seen in section 4, and requiring 15(M-2) multiplications and 12(M-2) additions over all; after which we would have a 2x2 matrix inversion. Still, the above operation is done twice, one for row filter optimization and another of column filter, doubling the above operations for multiplication and addition, for a 4x4 block, the above multiplication and addition operations are performed 30(M-2) and 24(M-2) times to generate two sets of filter coefficients (row and column filters), one of which is chosen to predict the current 4x4 block.

Encoder: All the modes in H.264 Intra predictions implement addition and shift operators for doing extrapolation of neighbouring pixels for prediction of 4x4 luma blocks. Therefore, aside from the complexity that exists in the computation of rate distortion optimization for selection of optimal modes, the actual prediction of a 4x4 block in one of the modes is quite fast. For 720P sequences the running time of the encoder increase by an average of 80% when placed as a second mode and an average of 100% for QCIF sequences.

Decoder: The operation involved in generating prediction using the row and column filters is identical to the operations in the encoder. Therefore, the more the decoder is signalled to use the new mode, the longer the time of execution becomes compared to standard decoder. For 720p sequences the decoding time increases by an average of 86%, and 100% for QCIF sequences. The approach with two additional modes increase the decoding time less since only one of the column or row analysis needs to be performed in this case.

7 Conclusion and possible future works

In this thesis a new Intra 4x4 prediction mode is presented. The new mode is added to existing H.264 intra 4x4 modes. The new mode is based on extrapolation of local structure in previously reconstructed pixels. In difference to H.264, the actual extrapolation performed by the new prediction mode is determined in both the encoder and the decoder. The highest bit rate saving that could be brought was when the new prediction was placed as a second mode which gave an average delta bit rate of -2.1% compared to the modified reference. In general, the rate distortion performance was about the same in all QPs. The larger the resolution of the sequence is, the less detailed the image area would be than its identical version in lower resolution. This means that an image area with an edge in a CIF frame would look smoother in a HD frame. This is the reason why the vertical, horizontal and DC predictions were dominating the structure of the local image area in HD sequences and the new mode was not adding much to the accuracy. Smoothening of the neighbouring reconstructions was a major problem which was hiding details of structures. Although the uncertainty is larger further away from the reference pixels, the sequential filtering (prediction of next row/column from previously predicted row/column) brings about loss of information by smoothening the structure.

Highly detailed frames are usually poorly predicted by standard intra prediction. Although the new block was better in performance for these types of frames and potentially has a higher chance in improving the rate distortion for these sequences, it still was much better in performance for smoother surfaces.

H.264 intra prediction favours to use intra 16x16 blocks in smooth areas as mentioned in section 3, and this fact indicates that the new algorithm for the 4x4 block will have a much less influence when enabling intra 16x16, especially for HD sequences giving only a 1.1% gain in bits compared to the modified reference with Intra 16x16 enabled.

As for adding two modes, the additional bit added for indicating the additional modes decreased the potential gain that could have been brought by referencing the target block, instead of the neighbouring blocks, for choosing between the row and column filtered predictions. However, the number of operations in the decoder could be halved, lessening the complexity in this case, since it will be directed to use one of the two analyses only (row or column).

As is observed, local structure estimation and prediction based on least squares analysis of immediate neighbouring rows and columns would indeed produce an approximation of a local structure, however, the majority of the 4x4 blocks produced were more accurate in predicting somewhat homogenous areas and detailed areas were not often predicted with any more accuracy than those produced by one of the standard intra 4x4 prediction modes in terms of distortion.

Possible future works

- I believe it would be possible to capture the local structure more accurately by doing the estimation in a more locally adaptive manner. That is, instead of doing a row by row or column by column analysis of the neighbouring block and using a fixed predictor for the entire 4x4 block, it might be beneficial to let the prediction of the 4x4 block to go on in a locally adaptive manner which would make the prediction more stable and perform well even when there is a sudden surface change. Currently, the predictor uses fixed coefficients that were generated from its neighbouring pixels in a row and column analysis to predict the whole 16 pixels in a 4x4 block, however, it could be tried to let the analysis to persist by including newly generated pixels along with neighbouring reconstructed pixels in a row or column in the current block for an

analysis to further generate new coefficients. Although this would obviously increase the complexity, it might help in adapting the predictor to newly emerging structure.

- Secondly, implementation of the new block type would potentially give a better gain in CABAC since CABAC (unlike CAVLC) is a lossless method and therefore will never hurt the quality of the neighbouring reconstructed blocks, i.e., it will be very easier to derive neighbouring structural information when the surrounding picture area quality is better, and the effect of smoothening due to high quantization will be less.

Bibliography:

[1] Ian E.G. Richardson**, "**H.*264 and MPEG-4 Video compression*". John Wiley & sons Ltd, 2003.

[2] ITU-T Series H, "*H.264*", Audio visual and multimedia systems.

[3] Gary J. Sullivan, Pankaj Topiwala, and Ajay Luthra, "*The H.264/AVC Advance Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions*", SPIE conference on applications of digital image processing (August 2004).

[4] Thomas Wiegand, Gary J.Sullivan,Gisle Bjontegaard and Ajay Luthra, "*Overview of the H.264 /AVC video coding standard*", IEEE transactions on circuits and systems for video technology, (October 2007).

[5] Johannes Balle & Mathias Wien, "*Extended Texture prediction for H.264 Intra coding*", ITU-T Q.6/SG16 VCEG, VCEG-AE 11, Marrakech, Morocco, (January 2007).

[6] Yan Ye , Marta Karczewicz, "*Improved Intra Coding*", ITU-T Q.6/SG16 VCEG, VCEG- AG 11, San Diego , (October 2007).

[7] Shiodera Taichiro, Akiyuki Tanizawa, Takeshi Chujoh, Tomoo Yamakage "*Improvement of Bidirectional Intra Prediction*", ITU-T Q.6/SG16 VCEG, VCEG-AG 08, Shenzhen, China (October 2007).

[8] Limin Liu, Yuxin(Zoe) Liu and Edward J. Delp, "*Enhanced Intra Prediction Using Context-Adaptive Linear prediction* ", Picture Coding Symposium (PCS 2007), Lisbon, Portugal (November 7-9, 2007).

[9] Yingjia Liu,Sixin Lin,Shan Gao,Lianhuan Xiong, "*Improving Intra DC prediction"* ITU-T Q.6/SG16 VCEG, VCEG-AH 12, Antalya, Turkey ,(January,2008).

[10] Gisle Bjontegaard , "*Calculation of Average PSNR Differences between RD curves*", ITU-T SC 16/Q6, VCEG, VCEG-M33, Austin, Texas, (April 2001).

[11] Gary Sullivan, Thomas Wiegand, "*Rate distortion optimization for video coding",* IEEE signal processing magazine, (November, 1998).

[12] A. Elyousfi, A. Tamtaoui, and E. Bouyakhf,*"A new fast intra prediction mode decision"*, International Journal of Computer Systems Science and Engineering Volume 4 Number 1,page 27 (2008).

[13] TK Tan, Gary Sullivan , Thomas Wedi, "*Recommended Simulation Common Conditions for Coding Efficiency Experiments, Revision 2*", ITU-T Q.6/SG16 VCEG, VCEG- AH 10, Antalya, Turkey, (January,2008).

[14] Topiwala, Pankaj; Tran, Trac; Dai, Wei," *Performance comparison of JPEG2000 and H.264/AVC high profile intra-frame coding on HD video sequences",*SPIE-2006.