EFFICIENT TRANSCODING OF AN MPEG-2 BIT STREAM TO AN H.264 BIT STREAM

by

ROCHELLE PEREIRA

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2005

ACKNOWLEDGEMENTS

The successful completion of my thesis is the consequence of teamwork and guidance of a number of people. I have had the opportunity to interact with very experienced professionals and learn from their invaluable experience.

I wish to express sincere thanks to Dr. K. R. Rao, IEEE Fellow, Multimedia Processing Laboratory, and U.T.A., who has been my mentor and guide during the course of my thesis. His innovative ideas have been a constant source of inspiration throughout my graduate studies.

I am profoundly grateful to the people I worked with at Broadcom Corporation, MA, for the experience and resources which have fostered my interest in this field.

I would also like to express my appreciation to Dr. S. Oraintara and Dr. Z. Wang, members of my thesis committee, for being a part of the culmination of my thesis.

I am grateful to the University of Texas at Arlington for the environment it has provided to hone my skills at a personal and professional level.

Last but not the least, I would like to thank my parents, my roommates and all other well wishers who have helped and supported me in my work.

November 14, 2005

ABSTRACT

EFFICIENT TRANSCODING OF AN MPEG-2 BIT STREAM TO AN H.264 BIT STREAM

Publication No. _____

Rochelle Pereira, M.S.E.E.

The University of Texas at Arlington, 2005

Supervising Professor: Dr. K. R. Rao

MPEG-2 has been a widely accepted video coding standard for various applications ranging from DVD to digital TV broadcast. The new H.264 AVC standard has an even broader perspective to support high and low bit rate multimedia applications on existing and future networks. The advantage in terms of better quality at a lower bit rate is why H.264 is fast replacing MPEG-2. However, the user end hardware had previously been adapted for MPEG-2 streams. This gives rise to a need for portability between MPEG-2 and H.264. The objective of the proposed heterogeneous transcoding process is to achieve standards transcoding from MPEG-2

main profile to H.264 main profile by reusing the data made available in the MPEG-2 bit stream and to perform a comparison with other proposed transcoding architectures.

Algorithm: The proposed research extracts the motion vectors and the transform coefficients from the incoming MPEG-2 bit stream. In the case of I frames, a standard deviation based fast mode decision process is applied to the transform coefficients. The pattern of the standard deviation of the 8x8 blocks within the each macroblock is used to determine the activity of the macroblock and then compute mode decisions compatible with the H.264 specification. In the case of P and B frames, the extracted motion vectors are refined over a one pixel window up to quarter pixel resolution and reused. This allows accommodating quarter pixel motion vectors as compared to half pixel motion vectors from MPEG-2 and it also allows reducing the effect of any errors due to the lossy quantization processes used. Inter frames also use a monotonic mode decision hierarchy, which does not force the MPEG-2 mode decisions yet reduces the computational complexity as compared to the full search motion estimation and mode decision process by 50%. The transcoder proposed reuses information from the MPEG-2 bit stream and also accommodates the advancements of the H.264 standard like sub macroblock partitioned motion search, direct modes in B frames etc. Thus the transcoder achieves low complexity, comparable quality and reduced bit rate in the transcoding process.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Multimedia compression is required to cater to the needs of a variety of conditions like transmission environments, varying bandwidth networks etc. For instance, it may be required that a single compressed bit stream be re-used in different environments. Each environment has its own characteristics. Hence there is a need to adapt the existing bit stream for reuse in each environment. Transcoding [2][4], trans-rating [28] and scalable video coding [29] are some of the present day solutions to this problem.

Scalable video coding [29] is currently being incorporated into the H.264/AVC [25] standard. It involves transmitting the basic compressed bit stream in the lowest layer called the base layer. Additional information that would be required to change the characteristics of the bit stream like the frame rate, spatial resolution and quality is provided in additional layers called enhancement layers. The base layer information and the enhancement layer information are packed into a single bit stream and transmitted. Every decoder will be able to decode the base layer, however only decoders which are capable of handling the enhancement layers can take advantage of it. This method of scalable video coding is preferred in scenarios such as broadcast where the user end requirements vary greatly. Also, it comes at the cost of added complexity in the

decoders and increased overhead in the compressed bit stream. A less expensive option is the use of transcoding. The main goals of a good transcoder are: the quality of the output stream should be comparable to that of the input stream, the bit rate of the output stream should not excessively exceed the bit rate of the input stream, the transcoding process should have low execution time and the information from the input stream should be reused as much as possible. Trans-rating is a special case of bit rate transcoding where the bit rate of the input bit stream has to be re-rated to fit the output requirements. The other characteristics of the stream are retained.

1.2 Outline of the work

This thesis is based on the transcoding of a MPEG-2 [6] bit stream into an H.264 [25] bit stream using a fast, low complexity approach. It essentially, reuses information from the input MPEG-2 bit stream to a large extent, to generate the transcoded H.264 bit stream. The thesis is organized as follows: Chapter 2 describes the MPEG-2 and H.264/AVC video coding standards in a nutshell; Chapter 3 describes the need for transcoding, the basic transcoding architectures and the pros and cons of each architecture; Chapter 4 describes how the proposed transcoding method is applied to intra (I) frames ; Chapter 5 describes the transcoding process applied to predicted (P) frames; Chapter 6 describes the transcoding process applied to bi-predicted (B) frames; Chapter 7 concludes with a comparisons of the obtained results.

CHAPTER 2

VIDEO BIT STREAMS

2.1 MPEG-2 encoder

MPEG-2 [14] is widely used in state-of-the-art video systems, including DVD, DBS and HDTV. Its popularity can be attributed to its efficient compression of both interlaced and progressive contents at bit rates ranging from 4 Mbit/s (DVD) to 19 Mbits/s(HDTV).

The typical MPEG-2 video encoder structure is shown in the Fig.2.1. The important and complex blocks are as follows:

DCT: The MPEG-2 encoder uses 8x8 2-D DCT. In the case of intra frames, it is applied to the 8x8 blocks of pels and in the case of inter frames it is applied to 8x8 blocks of the residual (motion compensated prediction errors). Since DCT is more efficient in compressing correlated sources, intra pictures compress more efficiently than inter pictures [14].

Quantization: The header information relevant to the quantizer scale factors is the quantizer scale code in the slice and macroblock headers and the quantizer scale type in the picture coding header [11]. Each quantization matrix has a default matrix (Fig. 5.6) [14]. User defined matrices may be downloaded and can occur in the sequence header or in the quantization matrix extension header.

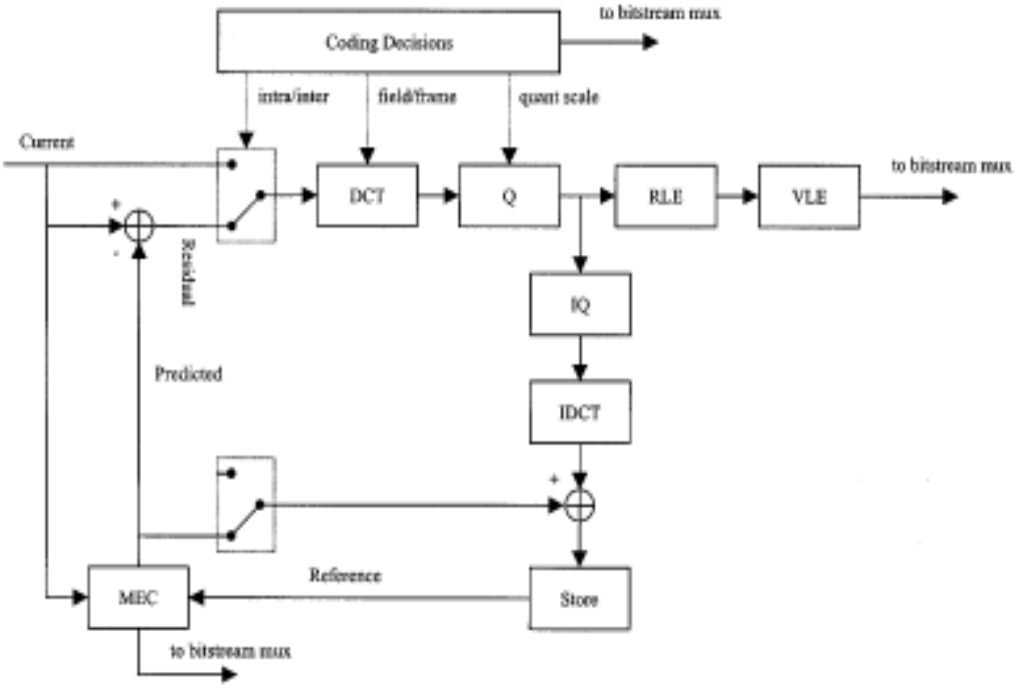


Fig. 2.1 Structure of the MPEG-2 encoder [31].

Motion Estimation and Compensation: In the motion estimation process, motion vectors per macroblock are estimated. These are coded differentially with respect to previously decoded motion vectors and transmitted. P frame macroblocks have one forward motion vector per macroblock. B frames have one forward motion vector and one backward motion vector per macroblock.

Coding Decisions: Coding modes in MPEG-2 are chosen based on whether the encoder encodes a frame picture as a frame or two fields or in the case of interlaced pictures , it can chose to encode it as two fields or use 16x8 motion compensation.

2.2 Structure of MPEG-2 coded video data

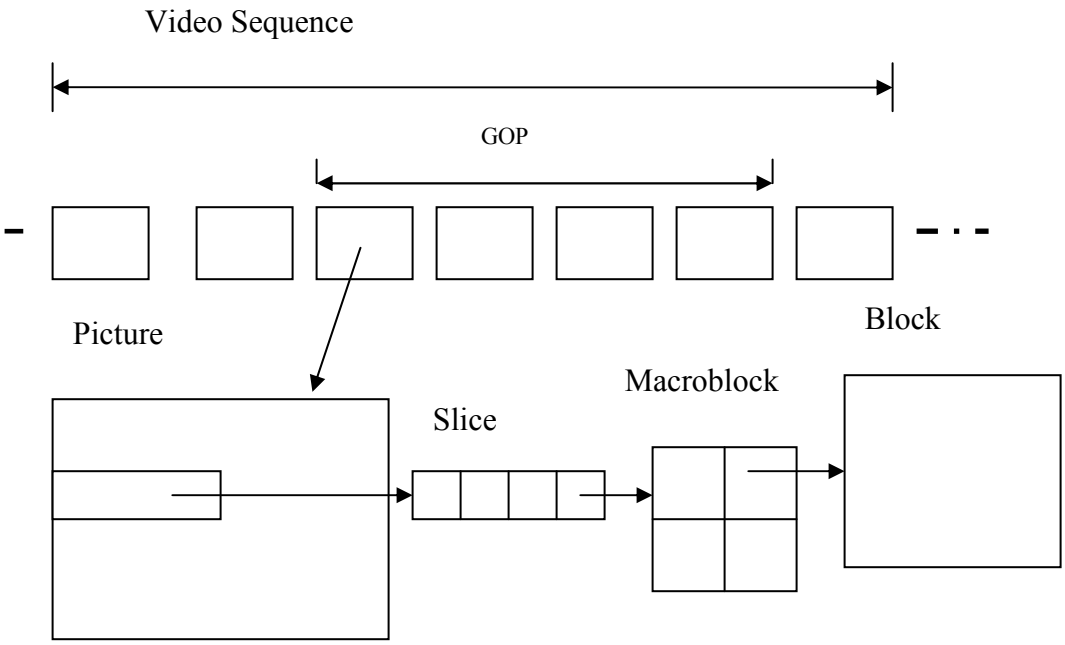
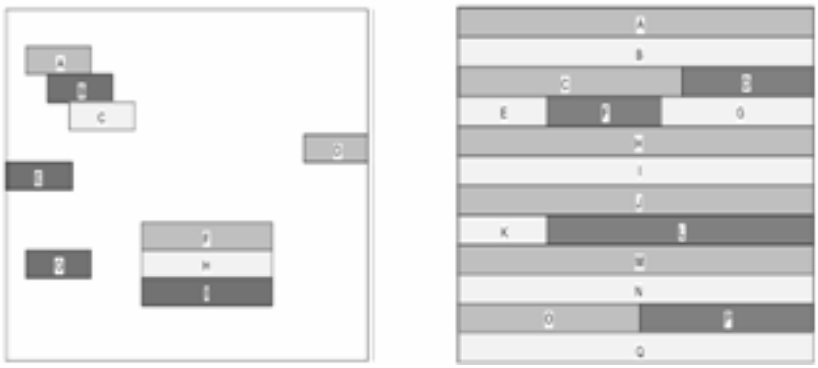


Fig. 2.2 Structure of a video sequence and its sub parts in MPEG-2

2.2.1 Video sequence

The video sequence (Fig. 2.2) is the highest element of the coded video bit stream. A video sequence commences with a sequence header (Fig. 2.6). The sequence header is followed by a sequence extension. This may be optionally followed by a group of picture headers and one or more coded frames. The GOP header is an optional header that can be used immediately before a coded I frame to indicate to the decoder if the first consecutive B frame immediately following the coded I frame can be reconstructed properly in the case of a random access [6]. If the preceding reference frame is not available those B frames, if any, cannot be reconstructed properly unless

they use backward prediction or intra coding. A group of picture header also contains time code information that is not used by the decoding process. The order of the coded frames is the order in which the decoder processes them but that is not necessarily the correct order for display. The video sequence is terminated by a sequence end code.



(a) (b)

Fig. 2.3 a. Slice structure [6] [14]. b. Restricted slice structure [6] [14]

A slice is a series of an arbitrary number of consecutive macroblocks. The first and last macroblocks of a slice (Fig. 2.3a) shall not be skipped macroblocks. Every slice must have at least one macroblock. Slices must not overlap [6]. In a general case, it is not necessary for slices to cover the entire picture. Those areas that are not enclosed in a slice are not encoded. In certain defined levels of defined profiles a restricted slice structure (Fig. 2.3b) is used. In this case, every macroblock in the picture must be enclosed in a slice.

The term macroblock refers to source and decoded data or to the corresponding data elements. There are three chrominance formats for a macroblock, namely, 4:2:0,

4:2:2 and 4:4:4 formats. The structure of the luminance and chrominance components for each format is depicted in Fig. 2.4.

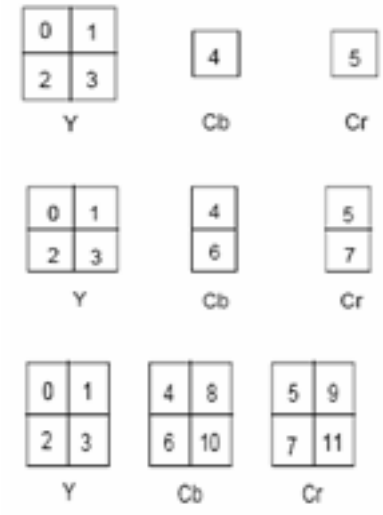


Fig. 2.4 Layout of the luma and chroma blocks in each macroblock for the 4:2:0, 4:2:2 and the 4:4:4 formats.

2.2.2 Video bit stream start codes

Start codes (Fig. 2.5) are used to identify the different elements in an MPEG-2 encoded bit stream. Every start code has a prefix and a value. The start code prefix [6] is '0000 0000 0000 0000 0000 0001'. The start code value is an 8-bit integer which identifies the type of the start code.

Name	Start code value (hexadecimal)
picture_start_code	00
slice_start_code	01 through AF
reserved	B0
reserved	B1
user_data_start_code	B2
sequence_header_code	B3
sequence_error_code	B4
extension_start_code	B5
reserved	B6
sequence_end_code	B7
group_start_code	B8
system start codes (Note)	B9 through FF

Fig. 2.5 Start codes in the MPEG-2 bit stream [6] [7] [14].

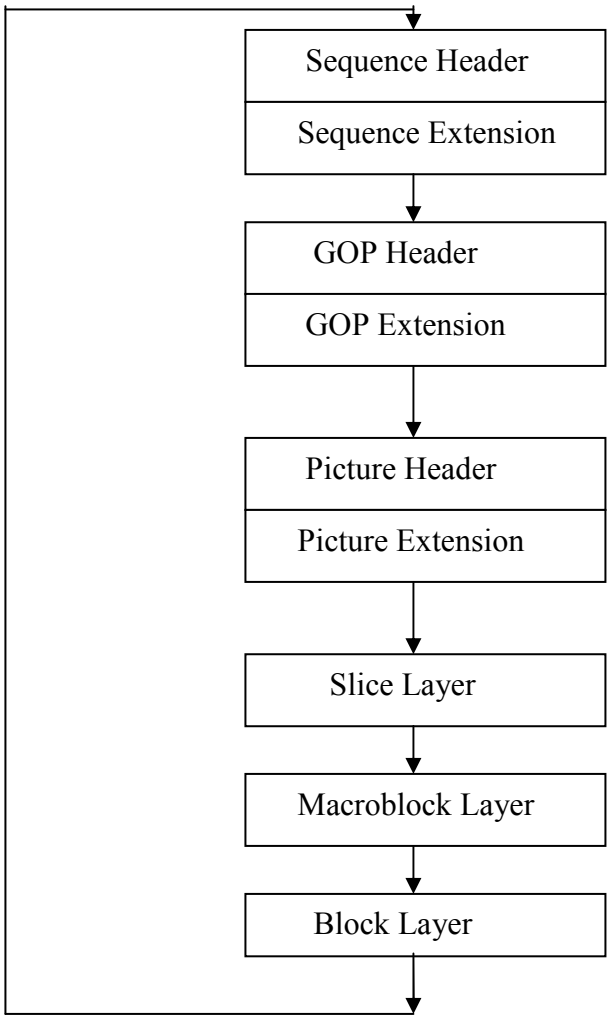


Fig. 2.6 MPEG-2 video hierarchy and extended functions [14].

2.3 MPEG-2 decoder

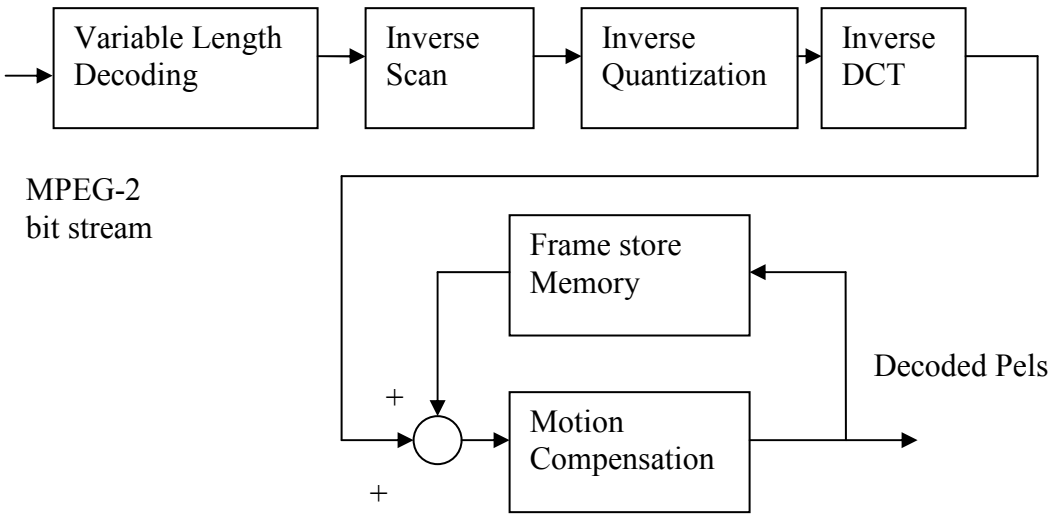


Fig. 2.7 Structure of the MPEG-2 decoder [6].

The main blocks of the MPEG-2 decoder (Fig. 2.7) are as follows:

Variable length decoding: This process has a table defined for decoding of intra DC coefficients and three tables for non intra DC coefficients, intra AC coefficients and non intra AC coefficients. The decoded values basically infer one of three courses of action: end of block, normal coefficients and escape coding [6].

Inverse scan: The data from the output of the variable length decoding process is one dimensional and of length 64. Inverse scan converts this one dimensional data

into a two dimensional array of coefficients according to a predefined scan matrix (Fig. 4.2).

Inverse Quantization: The two-dimensional array of coefficients is inverse quantized to produce the reconstructed DCT coefficients. This process is essentially multiplication by the quantizer step size. The quantizer step size is modified either by a weighting matrix or a scale factor. After inverse quantization, saturation and mismatch control are performed. Saturation ensures that the coefficients after inverse quantization lie in the range [-2048:+2047]. Mismatch control [6] is used to ensure that there are no large differences between the state of the encoder and the decoder. Sometimes small non zero inputs to the IDCT may result in zero outputs. If this occurs then mismatch may occur in some pictures. An encoder should avoid this by checking the output of its own IDCT.

Inverse Discrete Cosine Transform: Once the DCT coefficients are reconstructed, a 2-D 8x8 IDCT is applied to the inverse transformed values. In a macroblock, if the pattern code for block in the macroblock is one then the corresponding coefficient data is included in the bit stream. If the pattern code is zero or if the macroblock is skipped, then that block contains no coefficient data.

Motion Compensation: The motion compensation process forms predictions from previously decoded reference pictures which are combined with the coefficient data in order to recover the final decoded samples.

2.4 H.264 encoder

The primary goal of H.264 is to achieve high coding efficiency [11]. Unlike MPEG-2, it does not support at present layered scalable video coding (SVC). However, SVC is being actively developed as extensions to H.264. Further unlike MPEG-4 part 2 it does not support object based video or object based scalable coding.

Overview of profiles: The H.264 standard consists of tools designed to address efficient coding over a wide variety of video material. However, not all these tools are required for each application. If every decoder was forced to implement these tools, then there would be an unnecessary increase in the complexity. Therefore the standard defines subsets of tools intended for different classes of applications called profiles. The profile structure for H.264 is as shown in Fig. 2.8.

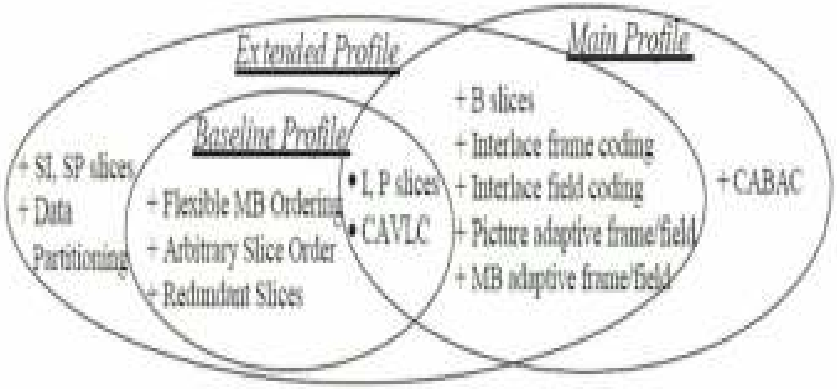


Fig. 2.8 Profile structure in H.264 [25].

The H.264 encoder structure is shown in Fig.2.9. The main blocks in the encoder are:

4x4 Integer transform: The 4x4 integer transform coefficients are explicitly specified in AVC and allows it to be perfectly invertible [11]. In AVC, the transform coding always uses predictions to construct the residuals, even in the case of intra macroblocks.

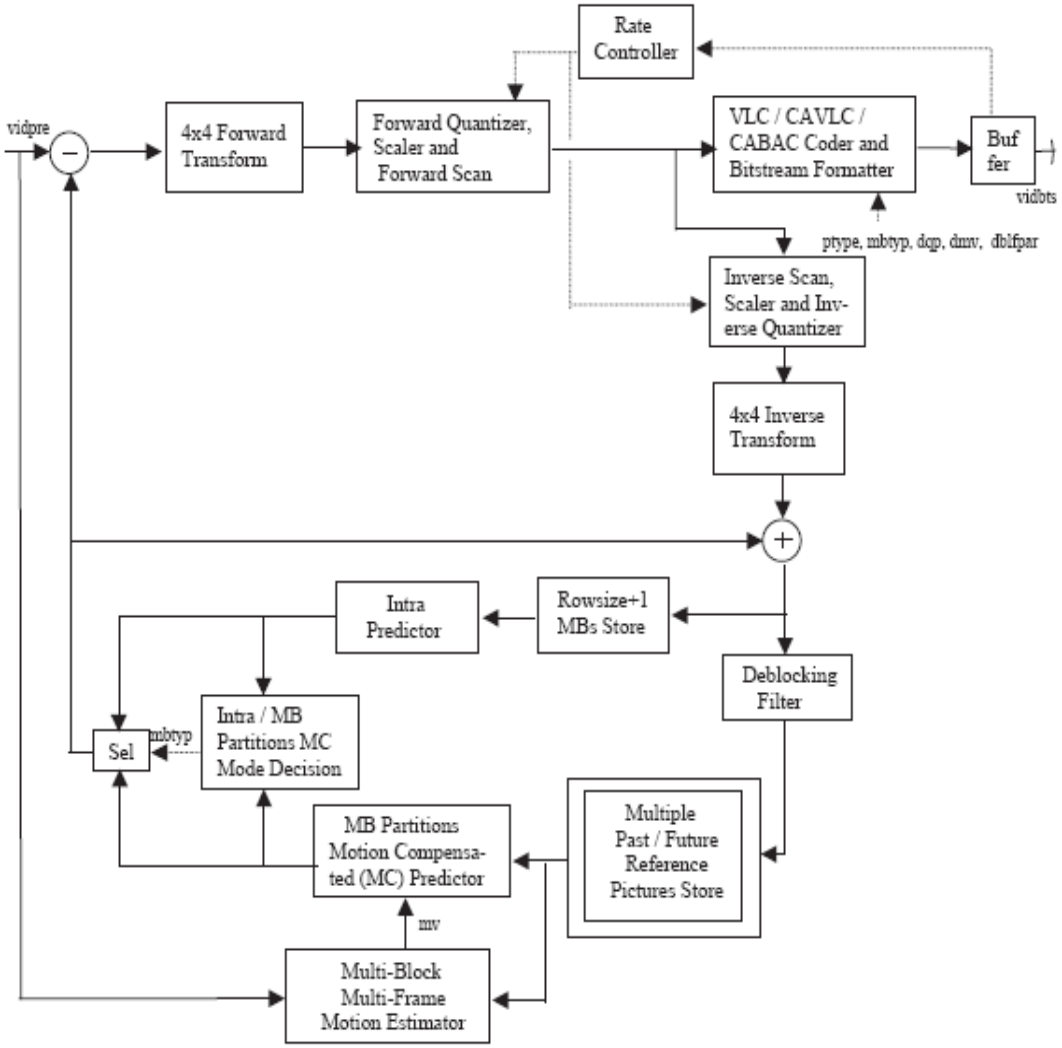


Fig. 2.9 H.264/AVC encoder block diagram [11].

Quantization and scan: The standard specifies the mathematical formulae of the quantization process [11]. The scale factor for each element in each sub block varies as a function of the quantization parameter associated with the macroblock, and as a function of the position of the element within the sub block. The rate control algorithm controls the value of quantization parameter. Two scan patterns for 4x4 blocks are used in this standard – one for frame coded macroblocks and one for field coded macroblocks.

CAVLC and CABAC entropy coders: VLC encoding of syntax elements for the compressed stream is performed using Exp-Golomb codes. For transform coefficient coding AVC includes two different methods CABAC and CAVLC. The entropy coding method can change as often as every picture.

Loop filter: The AVC loop filter, also called the deblocking filter [3], operates on a MB after motion compensation and residual coding , or on a macroblock after intra prediction and residual coding, depending upon whether the macroblock is inter coded or intra coded. The result of loop filtering is stored as a reference picture except for pictures that are not used as a reference picture. Loop filtering operates on the edges (Fig. 2.10) of both macroblocks and 4x4 sub-blocks.

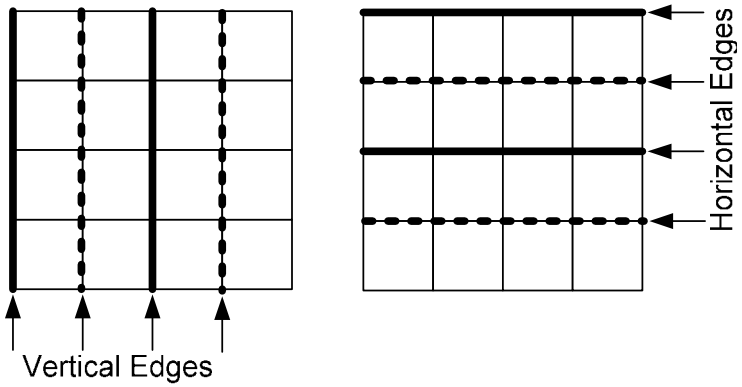


Fig. 2.10 Boundaries in a macroblock to be filtered [16][25].

Mode decision: This block decides the coding mode for each macroblock. Mode decision to achieve high efficiency may use rate distortion optimization. The outcome of mode decision is the best-selected coding mode for a macroblock.

Intra prediction: It forms predictions of pixel values by linear interpolation of pixel values from the neighboring macroblocks or blocks which have been previously decoded. The interpolations are directional in nature, with multiple modes. For luma pixels with 4x4 mode (Fig. 4.4), nine directional modes (Fig. 4.4) are defined. Four directional modes are defined when 16x16 mode is used.

Inter prediction: This block includes motion estimation (ME) and motion compensation (MC). It generates a predicted array of pixels from a previously decoded reference picture. In AVC the rectangular arrays of pixels that are predicted using MC can have the following sizes: 4x4, 4x8, 8x4, 8x8, 16x8, 8x16 and 16x16 pixels (Fig.

5.11) [16] [3]. The motion vectors used in the prediction process have quarter pixel precision. Chroma vectors at 1/8 pixel resolution are derived from the transmitted luma motion vectors of ¼ pixel resolution.

Multiple reference picture prediction: In the many previous standards, for prediction of blocks of a P-picture being coded, only immediately previous I or P picture is used as a reference [11]. However, the H.264 standard allows the current picture to predict from a wider set of previously decoded reference frames. This multi frame referencing capability is applied to both P pictures and B pictures.

2.5 Structure of H.264 coded video data

Coding of video is performed picture by picture. Each picture to be coded is first partitioned into a number of slices. Slices are individual coding units in this standard as compared to earlier standards as, each slice is coded independently [18]. A slice is a sequence of macroblocks, or, when macroblock adaptive frame field coding is used, a sequence of macroblock pairs. When macroblock adaptive frame field coding is in use, the picture is partitioned into slices containing an integer number of macroblock pairs. Each macroblock pair consists of two macroblocks (Fig. 2.11).

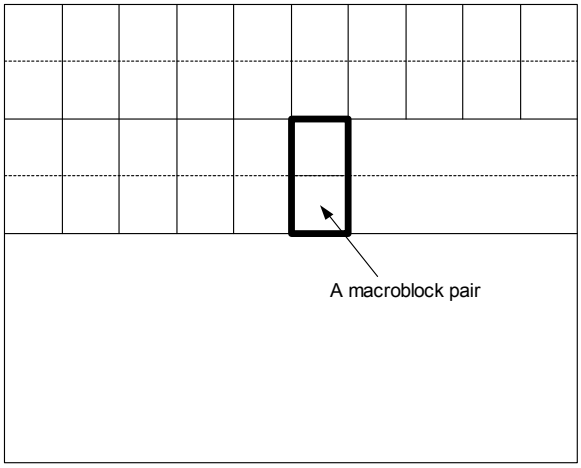


Fig. 2.11 Macroblock pairs in the case of MBAFF [16][18].

There are three basic slice types: I-intra, P-predictive and B-bi predictive slices. In H.264, I slice macroblocks are compressed without using any motion prediction from the slices in other pictures. A special picture called instantaneous data refresh (IDR) is defined which contains only I slices and all following frames cannot reference from any frame prior to the IDR picture. P slices consist of macroblocks that can be compressed by using motion prediction, but they can also have intra macroblocks. Macroblocks of a P slice when using prediction can use one prediction only. Unlike previous standards, the pixels used as reference for motion compensation can either be in past or in future in the display order [11]. B slices also consists of macroblocks that can be compressed by using motion prediction and like P slice they can also have intra macroblocks. Like earlier standards, one of the motion predictions can be past and the other in future in the display order, but unlike earlier standards, it is also possible to have both motion predictions from the past (Fig. 5.7) or both motion predictions (Fig. 6.1) from the

future. Also unlike earlier standards, B slices can be used as reference for motion prediction by other slices in the future or past. Besides I, P and B slices, there are two derived slice types called SI and SP slices. They allow switching between multiple coded streams.

CHAPTER 3

TRANSCODING ARCHITECTURES

MPEG-2 [6][14][15][16] has been a widely accepted video coding standard for various applications ranging from DVD to digital TV broadcast [8]. A large variety of products based on the MPEG-2 standard are available in the market. The most important goal of MPEG-2 was to make the storage and transmission of digital AV material more efficient. The new H.264 AVC standard [3] has an even broader perspective to support high and low bit rate multimedia applications on existing and future networks. The advantage in terms of better quality at a lower bit rate is why H.264 is fast replacing MPEG-2. However, the user end hardware had previously been adapted for MPEG-2 bit streams. This gives rise to a need for portability between MPEG-2 and H.264.

Video transcoding is the operation of converting video from one format to another [1]. A format is defined by characteristics such as bit rate, spatial resolution etc. One of the earliest applications of transcoding is to adapt the bit rate of a compressed stream to the channel bandwidth for universal multimedia access in various kinds of channels like wireless networks, Internet, dial-up networks etc. Changes in the characteristics of an encoded stream like bit rate, spatial resolution, quality etc can also be achieved by scalable video coding [2]. However, in cases where the available network bandwidth is insufficient or if it fluctuates with time, it may be difficult to set

the base layer bit rate. In addition, scalable video coding demands additional complexities at both the encoder and the decoder.

Transcoding [2] can be of different types depending upon the need for which it is designed. For instance, bit rate reduction transcoding is designed to accommodate compressed streams within network bandwidth constraints, spatial resolution reduction transcoding is used to accommodate output display spatial resolution constraints, temporal resolution reduction is used to accommodate frame rate variations and heterogeneous transcoding is used to make two different standards compatible. Transcoding can in general be implemented in the spatial domain or in the transform domain or in a combination of the two domains.

3.1 Transcoding architectures

The basic architecture for converting an MPEG-2 elementary bit stream into an H.264 elementary bit stream arises from complete decoding of the MPEG stream and then re-encoding into an H.264 stream. However, this involves significant computational complexity [4]. Hence there also is a need to transcode at low complexity.

The common transcoding architectures [2] are:

3.1.1. Open loop transform domain transcoding

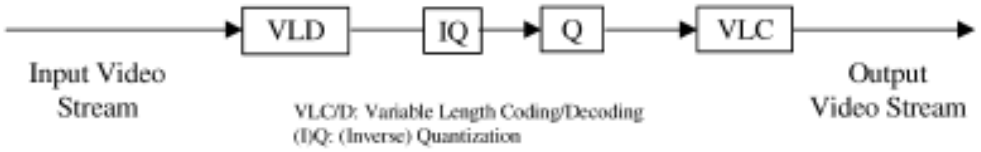


Fig. 3.1 Open loop transform domain transcoder architecture [2].

Open loop transcoders are computationally efficient. They operate in the transform (DCT) domain. However they are subject to drift error. Drift error occurs due to rounding, quantization loss and clipping functions.

3.1.2 Cascaded Pixel Domain Architecture (CPDT)

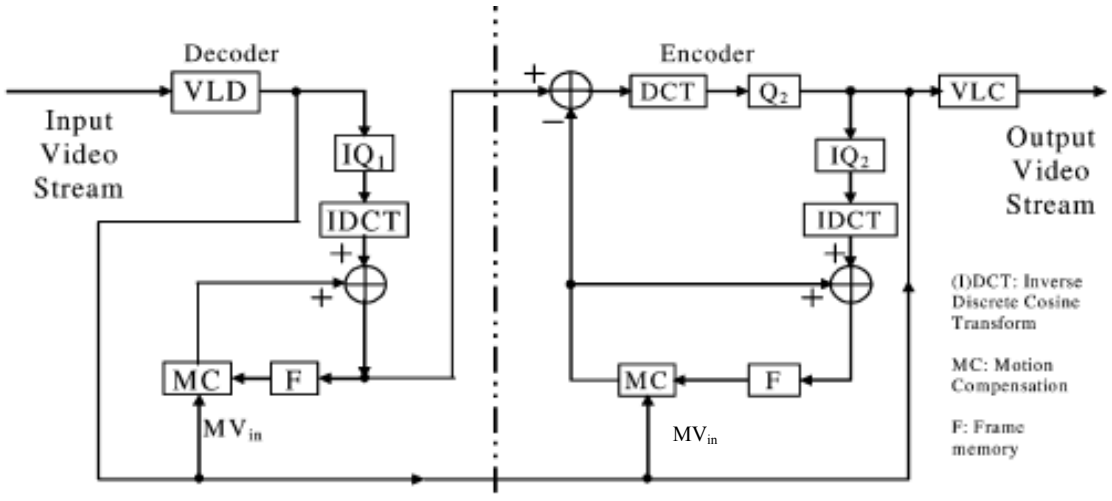


Fig. 3.2 Cascaded pixel domain transcoder architecture [2].

This is the most basic transcoding architecture. The motion vectors from the incoming bit stream are extracted and reused. Thus the complexity of the motion estimation block is eliminated which accounts for 60% of the encoder computation [4]. As compared to the open loop transcoding architecture, CPDT is drift free. Hence, even though it is slightly more complex, it is suitable for heterogeneous transcoding between different standards where the basic parameters like mode decisions etc are to be re-derived.

3.1.3 Simplified DCT Domain transcoders (SDDT)

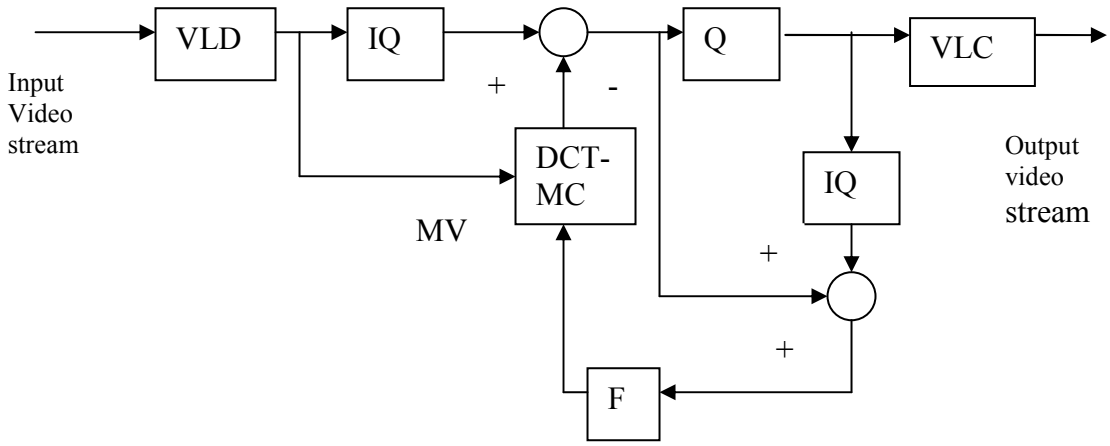


Fig. 3.3 Simplified transform domain transcoder architecture [2].

The transcoder (Fig. 3.3) is based on the assumption that DCT, IDCT and motion compensation are linear processes. This architecture requires that motion compensation be performed in the DCT domain, which is a major computationally intensive operation [4]. For instance, as shown in Fig. 3.4, the goal is trying to compute

the DCT coefficients of the target block B from the four overlapping blocks B1, B2, B3 and B4. It must be noted that these blocks B1, B2, B3 and B4 are all in the transform domain and hence to compute the transform coefficients of the block B block splitting and merging algorithms have to be used as discussed in [17]. These algorithms use intensive matrix multiplications and additions, also the number of matrices required will depend upon the number of maximum possible search positions within the macroblock. Moreover since H.264 supports several sub macroblock modes (Fig. 5.11) and quarter pixel motion search, the number of matrices required for the block splitting and merging processes is very large. Also, clipping functions and rounding operations performed for interpolation in fractional pixel motion compensation lead to a drift in the transcoded video [4].

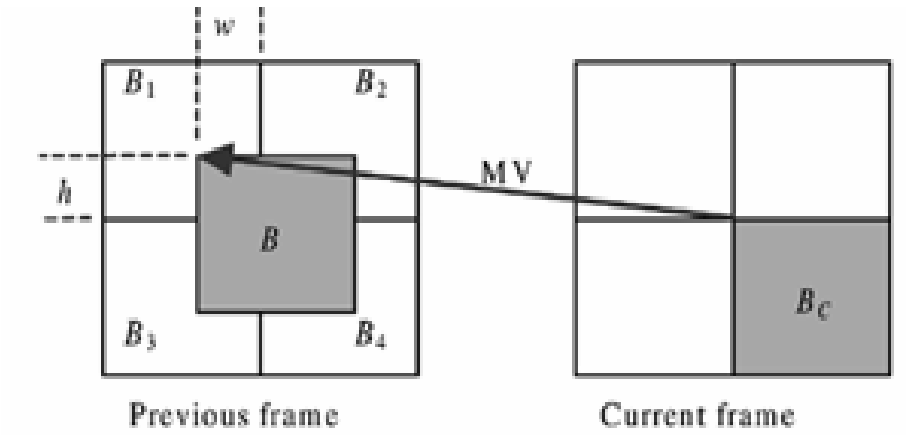


Fig. 3.4 Transform domain motion compensation illustration [2].

3.1.4 Cascaded DCT Domain transcoders (CDDT)

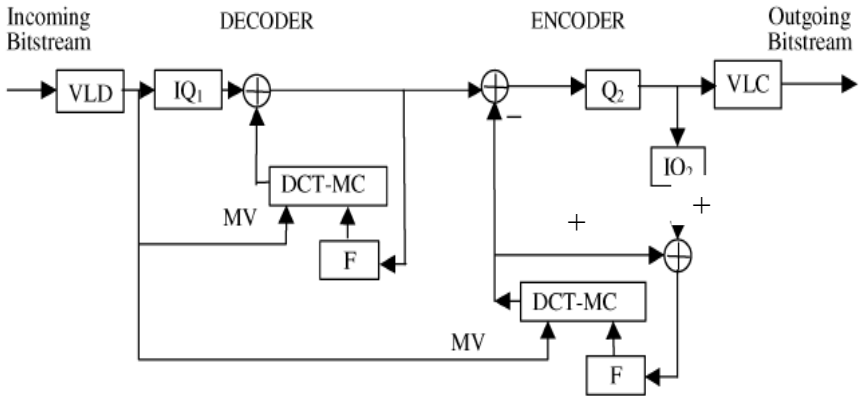


Fig. 3.5 Cascaded transform domain transcoder architecture [2].

CDDT is used for spatial/temporal resolution downscaling and other coding parameter changes. As compared with SDDT, greater flexibility is achieved by introducing another transform domain motion compensation block; however it is far more computationally intensive and requires more memory [1]. It is also, sensitive to drift like in SDDT. It is often applied to downscaling applications where the encoder end memory will not cost much due to downscaled resolution.

3.2 Choice of basic transcoder architecture

DCT domain transcoders have the main drawback that motion compensation in transform domain is very computationally intensive. DCT domain transcoders are also,

less flexible as compared to pixel domain transcoders, for instance, the SDDT architecture can preferably be used for bit rate reduction transcoding. It assumes that the spatial and temporal resolutions stay the same and that the output video uses the same frame types, mode decisions and motion vectors as the input video.

For heterogeneous transcoding from MPEG-2 to H.264, it is required to implement several changes in order to accommodate the sophistication of H.264 as compared to MPEG-2. For instance, MPEG-2 supports 16x16 and 16x8 macroblock partitions, but it is also required to refine the motion vectors to accommodate 8x16, 8x8 and sub 8x8 modes. Hence, the use of DCT domain transcoders is not very practical.

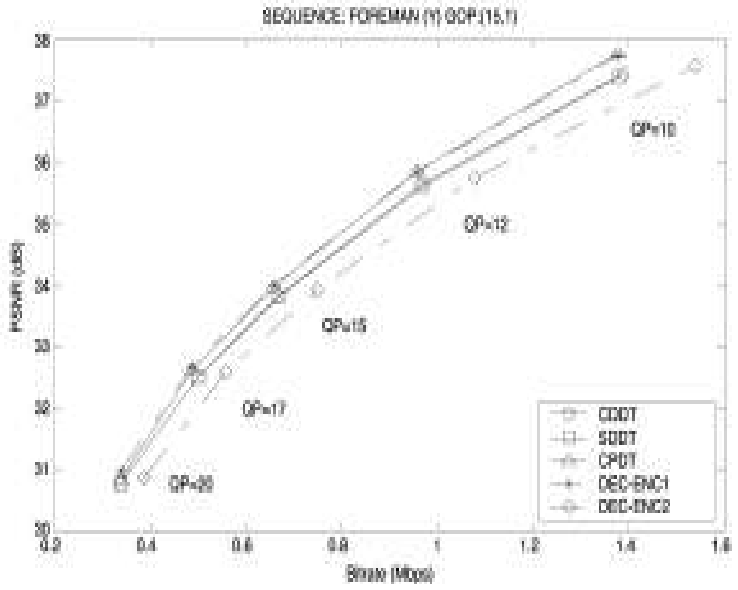


Fig. 3.6 PSNR vs. Bit rate graph for the Foreman sequence encoded at QP=7 and transcoded with different QP values and a GOP size 15 ,using different transcoding architectures as described in Fig. 3.1, 3.2, 3.3 and fig. 3.5. DEC-ENC1 is CPDT using full scale full search motion estimation. DEC-ENC2 is CPDT using three step fast search motion estimation [2].

From Fig. 3.6, it can be inferred that, the cascaded pixel domain architecture outperforms the DCT domain architecture. Also for larger GOP sizes, the drift in DCT domain transcoders becomes more significant, as it progressively builds up till the next I frame is coded (Fig. 3.7). These large GOP sizes are practically used especially in implementations like networked video streaming and wireless video where high coding efficiency is desired.

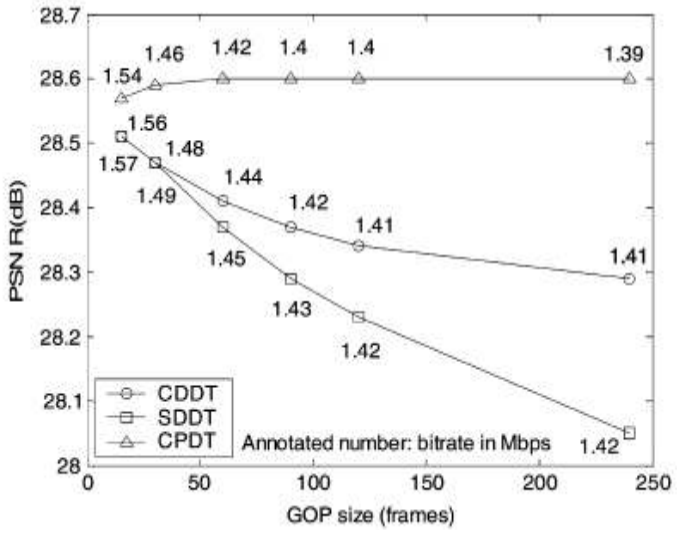


Fig. 3.7 Performance comparison of average PSNR for CPDT, SDDT and CDDT for different GOP sizes, using the test clip mobile-calendar encoded at QP=5 and transcoded at QP=11 [2].

Based on the comparison of the various transcoding techniques, it can be inferred that from the point of view of low complexity and fast execution, pixel domain transcoding with motion vector reuse offers the best performance in terms of PSNR, execution time and resistance to drift.

CHAPTER 4

INTRA FRAME TRANSCODING

Before discussing transcoding from MPEG-2 [6] to H.264, it is important to study the intra frame coding process in both of these.

4.1 Intra Frame coding in MPEG-2

In an MPEG-2 bit stream, if the picture header indicates the three bit code 001 for the picture coding type, it is an intra frame. MPEG-2 standard performs coding of intra frames by dividing the given image into non-overlapping 8x8 blocks. Then the 8x8 2D DCT [14] (eq. 4.1) is applied to each block. Thus each block of 8x8 pels results in an 8x8 block of DCT coefficients. Since DCT is more efficient for highly correlated sources, the I frames can be more efficiently compressed than the P frames or B frames.

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \tag{4.1}$$

where,

f(x,y) represents the input pixel domain values.

F(u,v) represents the output transformed coefficients.

u, v, x, y =0,…, N-1

x, y are spatial co-ordinates in the sample domain.

u, v are co-ordinates in the transform domain.

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u\, v = 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.2}$$

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Fig. 4.1 Default quantization matrix for intra DCT coefficients [7][14].

After the 8x8 2-D DCT, each macroblock has six blocks (Fig. 2.4) associated with it; four 8x8 luma blocks and two 8x8 chroma blocks for 4:2:0 format. It is required to transcode this MPEG-2 I frame data into H.264 I frame data with compatible mode decisions made. In the MPEG-2 encoder, the DCT coefficients are then quantized using a default matrix (Fig. 4.1) or modified matrix [14]. The quantizer step size is derived from the quantization matrix and the quantizer scale. In I pictures, intra mode (intra-d) and intra with modified quantizer mode (intra-q) are used. If macroblock type is intra-q (Table 4.1), then the macroblock header contains a 5-bit integer that defines the

quantizer scale. The scale can be changed from 1 to 41, both inclusive. If macroblock type is intra-d (Table 4.1), then no quantizer scale is transmitted and the decoder uses the previously set value. The quantized DCT coefficients are coded losslessly by a DPCM technique. The DC predictors used in this technique are reset to 128, 256, 512, or 1024 according to the DC precision required, 8, 9, 10, 11 bits, respectively. The a.c. coefficients are coded using run/level coding technique. These quantized coefficients are then scanned and converted to a one dimensional array and they are variable length coded. There are two methods of scanning available, zigzag scan which is typical for progressive mode processing and alternate scan which is more efficient for interlaced format video.

Table 4.1 Macroblock type VLC for I pictures [7][14].

Macroblock type	VLC Code	MB-quant	MB-intra
Intra-d	1	0	1
Intra-q	01	1	1

v \ u	0	1	2	3	4	5	6	7
0	0	1	5	6	14	15	27	28
1	2	4	7	13	16	26	29	42
2	3	8	12	17	25	30	41	43
3	9	11	18	24	31	40	44	53
4	10	19	23	32	39	45	52	54
5	20	22	33	38	46	51	55	60
6	21	34	37	47	50	56	59	61
7	35	36	48	49	57	58	62	63

(a)

v \ u	0	1	2	3	4	5	6	7
0	0	4	6	20	22	36	38	52
1	1	5	7	21	23	37	39	53
2	2	8	19	24	34	40	50	54
3	3	9	18	25	35	41	51	55
4	10	17	26	30	42	46	56	60
5	11	16	27	31	43	47	57	61
6	12	15	28	32	44	48	58	62
7	13	14	29	33	45	49	59	63

(b)

Fig 4.2 Scan matrices in MPEG-2 [14], (a) Zig-zag scan (b) Alternate scan.

4.2 Intra Frame coding in H.264

H.264 performs adaptive spatial prediction to exploit the spatial redundancy in the I frame. It supports nine prediction modes for a 4x4 sub-block and four prediction modes for a 16x16 luma macroblock. For chroma 4 different modes are defined which are similar to the 4 modes for 16x16 intra luma prediction. Both the chroma blocks Cb and Cr use the same prediction mode. If a sub-block or a macroblock is to be coded in the intra mode, a prediction block is formed based on the neighboring samples of the previously coded blocks that are to the left and/or immediately above the block to be coded. The prediction block is then subtracted from the original macroblock to obtain the error residual.

Intra prediction modes:

Spatial prediction is performed on either 4x4 blocks or a 16x16 luma macroblock. The use of 4x4 blocks definitely offers better accuracy in prediction, however it requires greater overhead for conveying the sub-block mode decisions etc in the compressed bit stream and hence the 16x16 macroblock mode may offer better overall bit rate.

4x4 Prediction of Luma:

Each macroblock is divided into 4x4 sub-blocks. Each sub-block can choose to predict in DC mode or use directional prediction as shown in Fig.4.3. The 9 possible prediction modes are listed in table 4.2.

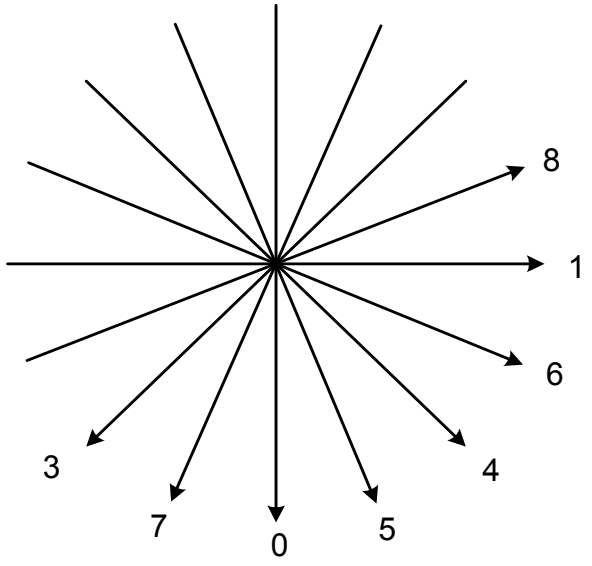


Fig. 4.3 Directional prediction modes in a 4x4 sub-block [6].

Table 4.2 Intra prediction modes for a 4x4 sub-block [6]

Intra4x4PredMode[luma4x4BlkIdx]	**Name of Intra4x4PredMode[luma4x4BlkIdx]**
0	Intra_4x4_Vertical (prediction mode)
1	Intra_4x4_Horizontal (prediction mode)
2	Intra_4x4_DC (prediction mode)
3	Intra_4x4_Diagonal_Down_Left (prediction mode)
4	Intra_4x4_Diagonal_Down_Right (prediction mode)
5	Intra_4x4_Vertical_Right (prediction mode)
6	Intra_4x4_Horizontal_Down (prediction mode)
7	Intra_4x4_Vertical_Left (prediction mode)
8	Intra_4x4_Horizontal_Up (prediction mode)

For an illustration consider the 4x4 block shown in Fig 4.4., in which the pixels of the current macroblock are a, b,….,p and A,B,…,L represent the previously decoded pixels . In the case of vertical prediction, the pixel A is used to predict pixels a, e, i and m, similarly the pixel B is used to predict the pixels b, f, j and n, and so on.

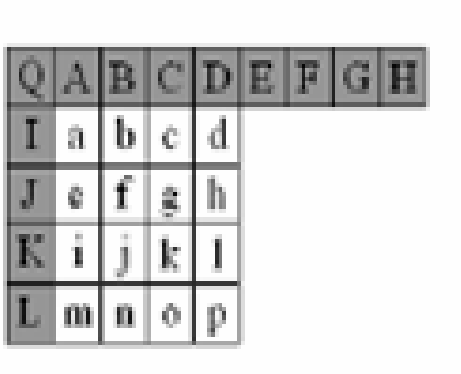


Fig. 4.4 Pixel illustration of a 4x4 block and the surrounding pixels [6].

Table 4.3 Intra prediction modes for a 16x16 macroblock [6].

Intra16x16PredMode	**Name of Intra16x16PredMode**
0	Intra_16x16_Vertical (prediction mode)
1	Intra_16x16_Horizontal (prediction mode)
2	Intra_16x16_DC (prediction mode)
3	Intra_16x16_Plane (prediction mode)

Table 4.3 lists the different prediction modes for the 16x16 macroblock. For an illustration, consider a 16x16 block, in the case of vertical prediction each of the 16 columns of the current macroblock is predicted from only 1 past decoded pixel, in the corresponding column.

4.3 Coding mode decisions in the H.264 encoder

Before implementing any part of transcoding, it is important to study the behavior of the H.264 encoder. Only after observing the behavior of the system, one will be able to emulate the way mode decisions are made without having to go through an exhaustive search.

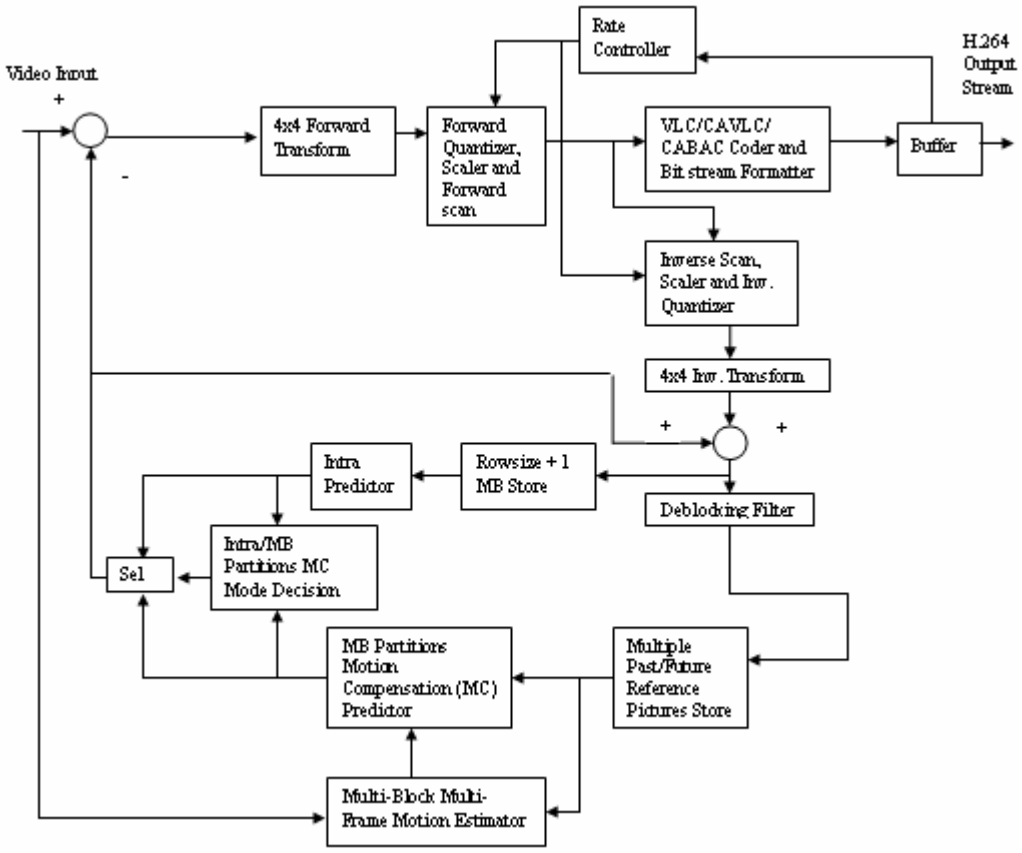


Fig. 4.5 H.264 /MPEG-4 AVC encoder block diagram [11]

Motion estimation and mode decision are the most computationally intensive processes in the encoder. They at times take up to 80 % of the total encoding time . The next relatively intensive process is applying the 4x4 forward transform and the 4x4 inverse transform.

In the case of intra prediction specifically, the two main modes possible, as discussed earlier, are 4x4 and 16x16 modes. It is clear that the decision between the two modes relates to a tradeoff between compression and coding efficiency. When the macroblock contains intricate detailed information, it will be necessary to encode with maximum coding efficiency, otherwise it is necessary to exploit spatial redundancy and achieve maximum compression. A good measure of the information content of the macroblock is its statistical properties.

To compare the way the encoder chooses between these two intra modes and whether it has any relation with the statistical properties of the DCT coefficients like standard deviation , one must run several tests and plot the modes chosen versus the standard deviation of each macroblock in the I frame. The results obtained are shown in Figs.4.6, 4.7 and 4.8. These are 352x288 progressive sequences which were encoded at a constant bit rate of 1 Mbps without rate-distortion optimization. As it can be seen, the macroblocks with a lower standard deviation tend to choose the 16x16 mode. After the first threshold, they tend to switch between the 16x16 mode and the 4x4 mode and, after the next higher threshold they very distinctly choose the 4x4 mode. There are aberrations to this relationship, as can be observed. However they are negligibly small.

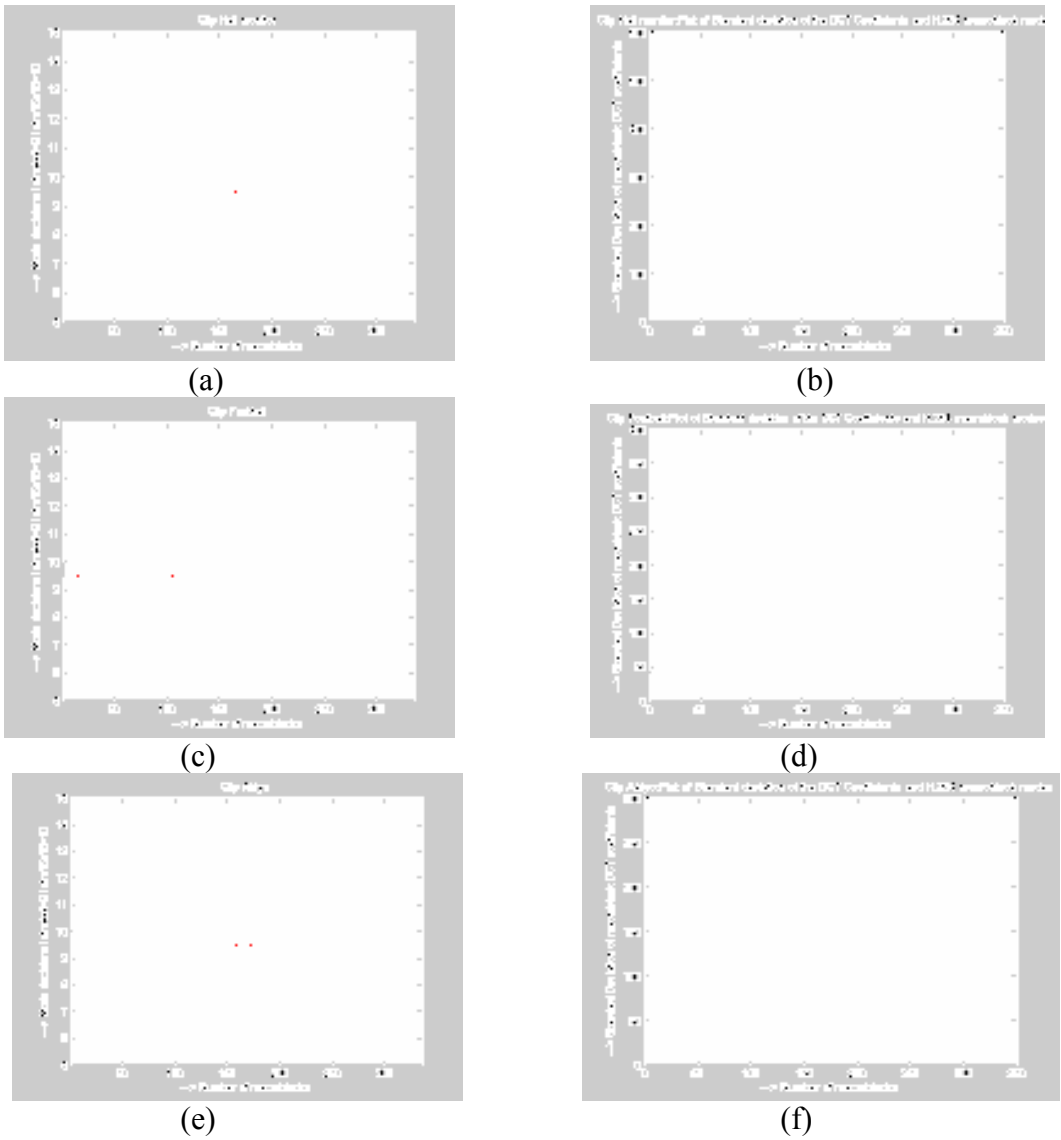


Fig. 4.7 a. The mode decisions (intra 4x4 or 16x16 computed for a I frame in Clip Hall monitor are plotted vs. the number of macroblocks. b. The corresponding values of the standard deviation of the macroblocks vs. plotted versus the number of macroblocks. c. The mode decisions (intra 4x4 or 16x16 computed for an I frame in Clip Football are plotted vs. the number of macroblocks. d. The corresponding values of the standard deviation of the macroblocks vs. plotted versus the number of macroblocks. e. The mode decisions (intra 4x4 or 16x16 computed for an I frame in Clip Akiyo are plotted vs. the number of macroblocks. f. The corresponding values of the standard deviation of the macroblocks vs. plotted versus the number of macroblocks.

If this relation can be exploited during transcoding, since the DCT coefficients of the error residual are already available, the complexity of the intra-predictor block and mode decision can be completely avoided.

4.4 Transcoding of an I frame

The general block diagram of the transcoder which would implement standard deviation based I frame mode decisions is shown in Fig.4.7. This proposed system essentially affects the H.264 encoder side.

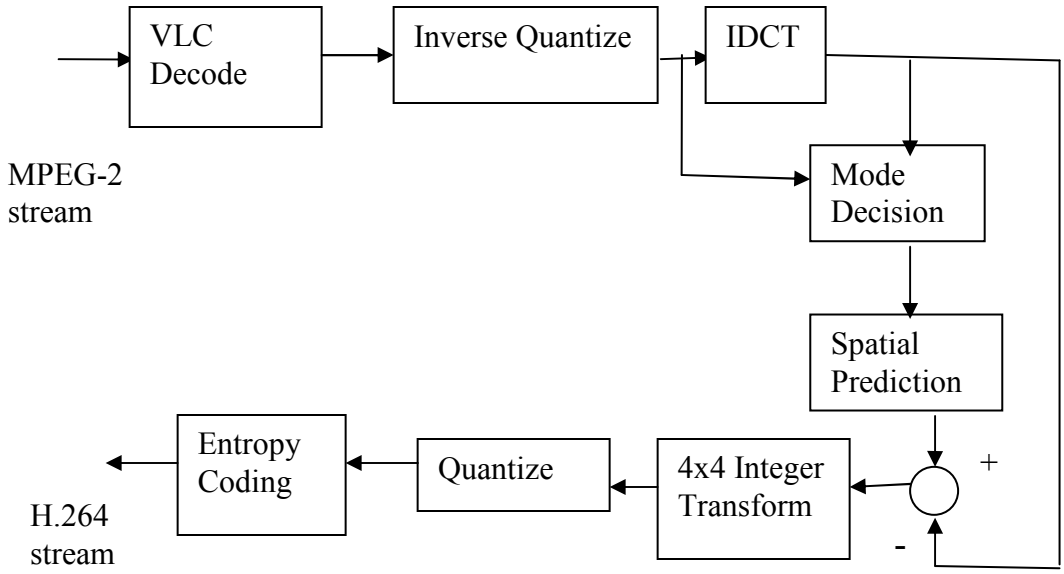


Fig. 4.7 General block diagram of the transcoder processing for an I frame.

The detailed flowchart of how a simple intra macroblock will be handled in software is shown in Fig. 4.8. The output from the decoder is the prediction residual after inverse discrete cosine transform. The transform coefficients are used to make intra mode decisions. After adding up the residual, to obtain the reconstructed intra frame, it is directly fed into the encoder side, along with the mode decisions indicating whether the 16x16 or the 4x4 mode is to be used. Each macroblock is then predicted

using these mode decisions. Sub macroblock directional modes are chosen based on macroblock availability and sum of absolute difference value. For instance, suppose the 16x16 mode was selected. Then the availability of the surrounding macroblocks is determined. If only the macroblock to the left of the current macroblock is available then the 16x16 horizontal mode is selected as a valid mode for prediction. The sum of absolute difference (cost) for that mode is computed. The predicted macroblock is subtracted from the input macroblock and then the difference is transformed, quantized and entropy coded. If there is more than one macroblock available and there is more than one valid mode, then the best mode selected is the one with the minimum cost.

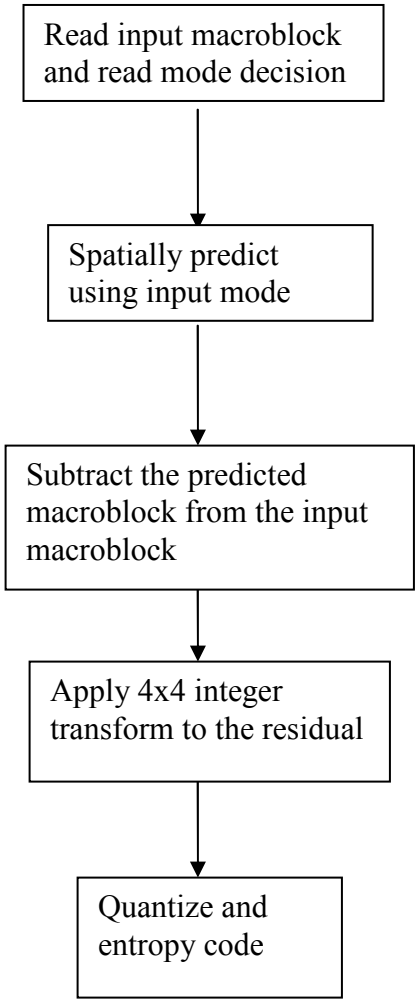


Fig. 4.8 Intra coding using previously computed variance based mode decisions.

4.5 Mode decision algorithm

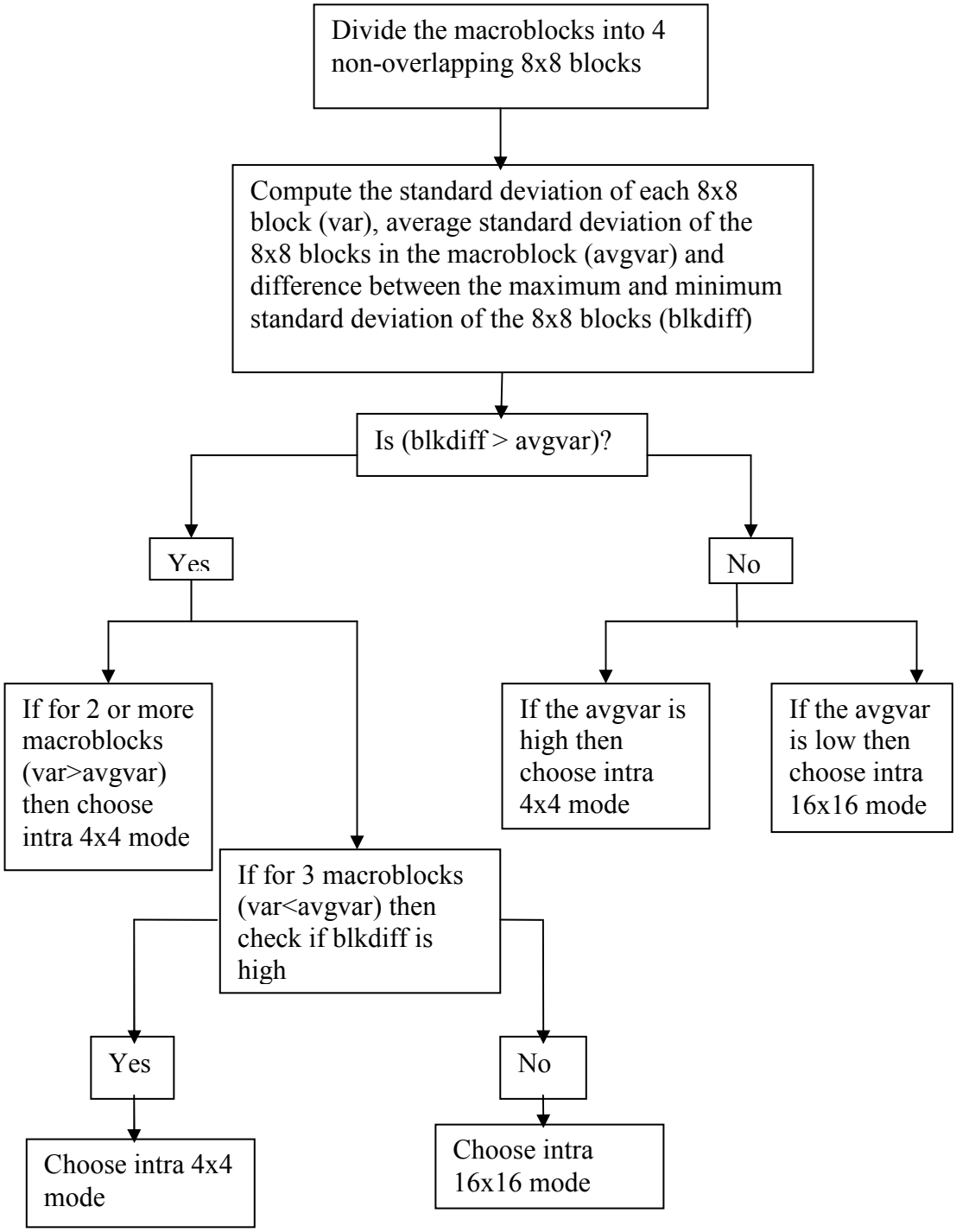


Fig. 4.9 Mode decision algorithm for I frame transcoding.

4.6 Results

As shown in Fig. 4.9, the mode decision algorithm essentially captures the need for choosing coding efficiency over reduction in bit rate .It uses the standard deviation as an indication of the amount of information contained in the macroblock. It studies the pattern of the variations in the standard deviation within the blocks of a macroblock and determines the need to use either intra 16x16 mode or intra 4x4 mode.

Table 4.4 Results obtained for the first I frame when the test clips were transcoded with variance based mode decision algorithm.

	PSNR (dB)		Bits used (bits/pic)		Execution Time (ms)	
Test clip	Exhaustive SAD based mode selection	Standard deviation based mode selection	Exhaustive SAD based mode selection	Standard deviation based mode selection	Exhaustive SAD based mode selection	Standard deviation based mode selection
Akiyo	43.221	43.086	56480	67720	300	240
Coastguard	39.798	39.81	119880	126024	420	331
Football	39.316	39.368	139576	142520	420	391
Foreman	41.448	41.4	75008	92520	301	241
Crawfish	42.2	42.153	68864	77592	301	240
Flower Garden	39.232	39.303	215200	217376	381	311
Hall Monitor	42.131	42.083	76712	91224	310	240

Table 4.4 list the results obtained for an I frame when the mentioned test clips were transcoded at 1 Mbps from an MPEG-2 bit stream to an H.264 bit stream. The

mode decisions for the H.264 stream were made using the algorithm mentioned above. The PSNR obtained by the proposed new low complexity transcoding scheme is comparable to that obtained by complete decoding and re-encoding of the bit stream. The bits used however, tend to be slightly higher than those in the case of full decoding and re-encoding. The full decoding and re-encoding process, performs an exhaustive search to find the best mode with the minimum cost. Hence the prediction residual is also very small and it requires less number of bits to be transmitted. However, in the proposed scheme, achieving low complexity is also of significant importance and hence a small increase in the number of bits used is acceptable with negligible loss in PSNR.

Figs. 4.12a-4.15c show an I frame in the MPEG-2 stream, after transcoding and with complete decoding and re-encoding. As can be observed, the subjective quality of these frames is very close and comparable.

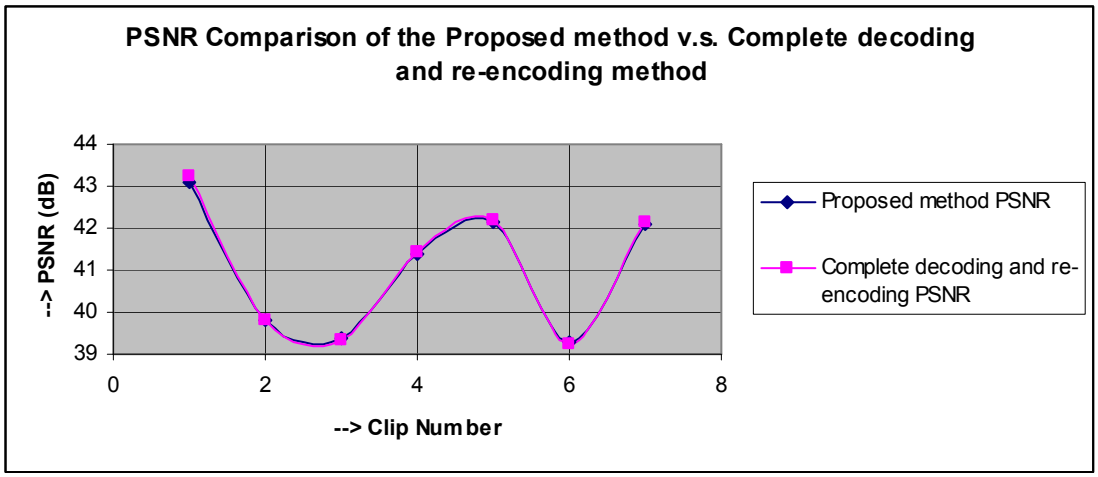


Fig. 4.10 Comparison of the PSNR of the proposed method and complete decoding and re-encoding method.

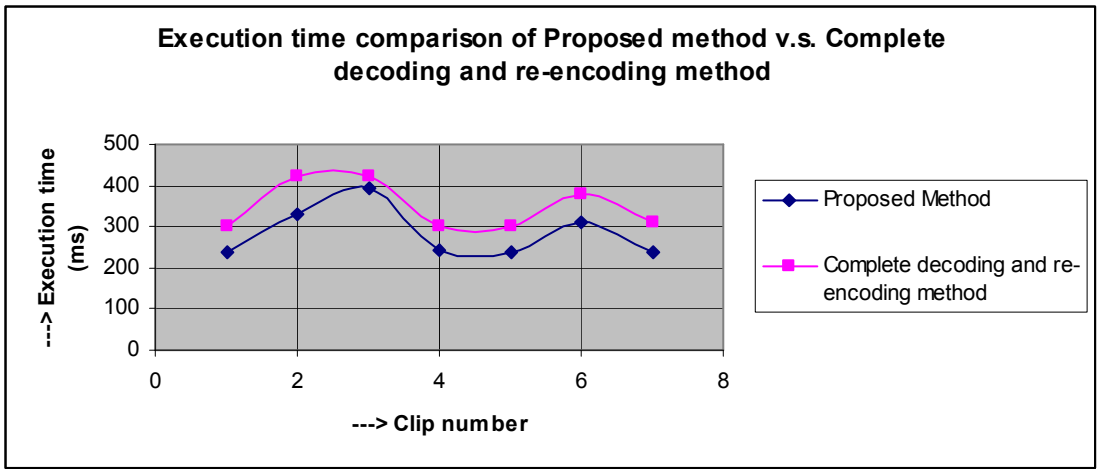


Fig. 4.11 Comparison of the execution time of the proposed method and complete decoding and re-encoding method.

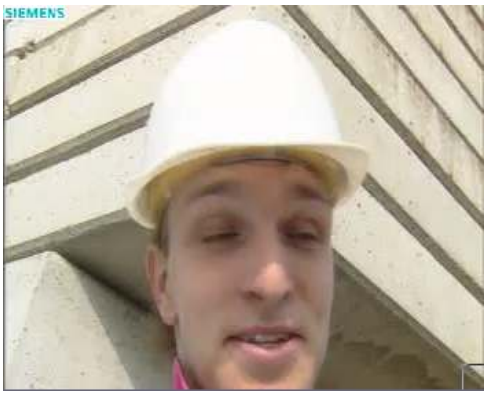
(a)

(b)

(c)

Fig 4.12 a. Subjective quality of an I frame of the clip Akiyo in an MPEG-2 compressed stream. B.Subjective quality of an I frame of the clip Akiyo in the H.264 transcoded compressed stream. c. Subjective quality of an I frame of the clip Akiyo in the H.264 re-encoded compressed stream.

(a)



(b)

(c)

Fig. 4.13 a. Subjective quality of an I frame of the clip Foreman in an MPEG-2 compressed stream. b.Subjective quality of an I frame of the clip Foreman in the H.264 transcoded compressed stream. c. Subjective quality of an I frame of the clip Foreman in the H.264 re-encoded compressed stream.

(a)

(b)

(c)

Fig. 4.14 a. Subjective quality of an I frame of the clip Flower garden in an MPEG-2 compressed stream. b.Subjective quality of an I frame of the clip Flower garden in the H.264 transcoded compressed stream.c. Subjective quality of an I frame of the clip Flower garden in the H.264 re-encoded compressed stream.

(a)

(b)

(c)

Fig. 4.15 a. Subjective quality of an I frame of the clip Coast guard in an MPEG-2 compressed stream.b.Subjective quality of an I frame of the clip Coast guard in the H.264 transcoded compressed stream. c. Subjective quality of an I frame of the clip Coast guard in the H.264 re-encoded compressed stream.

CHAPTER 5

P FRAME TRANSCODING

5.1 P frame coding in MPEG-2

In the MPEG-2 standard [6], macroblocks in P frames can be coded using inter modes or the intra modes. Macroblocks that are inter-coded seek to exploit temporal correlation among frames and thus achieve compression. In P pictures, some modes available are MC/no MC, coded/not coded, intra/inter and quantizer modification or not. The standard itself does not specify how to make these decisions.

The MC/ no MC decision is by the encoder, whether to transmit motion vectors or not. If the motion vector is zero due to MC decision then some bits can be saved by not transmitting it. The intra/non intra coding decision is made based on variance. The coded/not coded decision is a result of quantization. If all the quantized DCT coefficients are zero then the block need not be coded. The quant/no quant decision is made as to whether the quantizer scale is to be changed or not. This is usually based on the frame content and on the buffer fullness assessment of the decoder. The decision process for each macroblock can be described as shown in Fig.5.1.

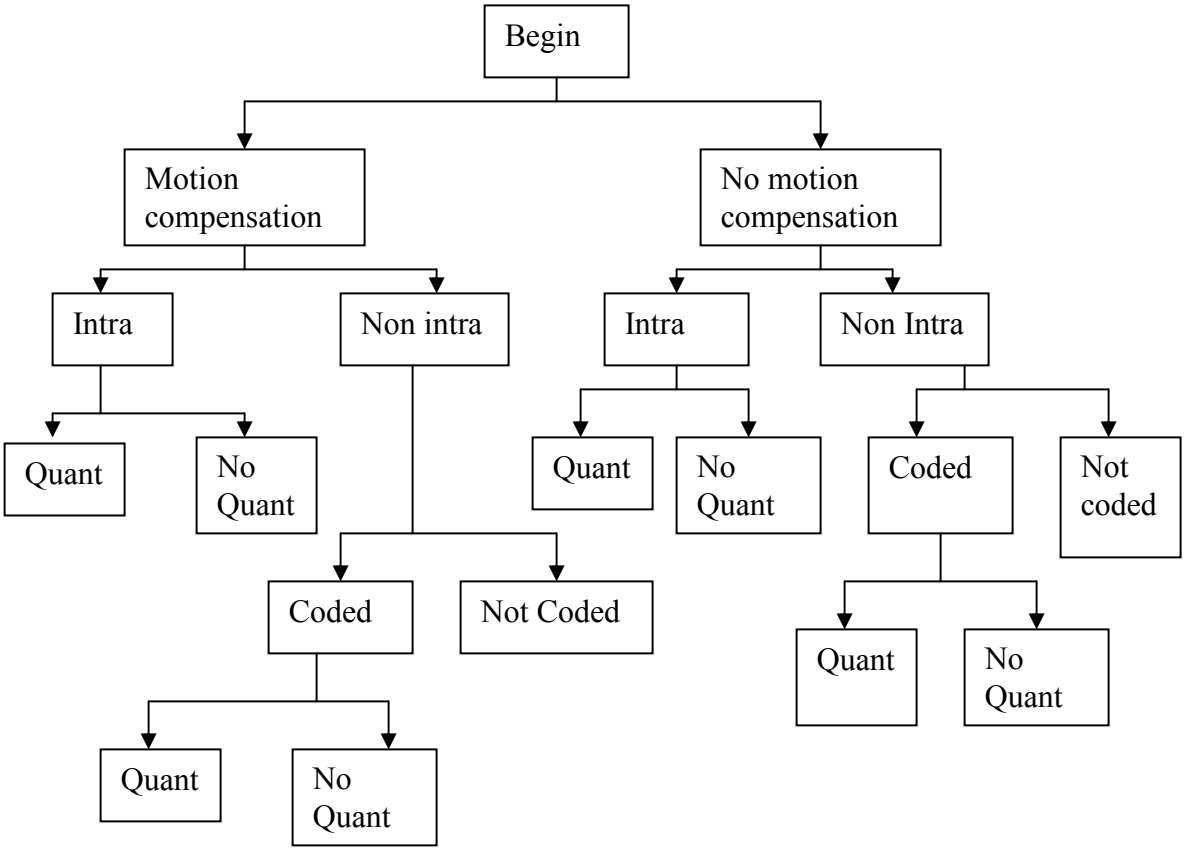


Fig. 5.1 Macro block mode selection process for P frames in MPEG-2.

5.1.1 Motion Compensated Prediction

In the motion estimation process, the motion vectors for predicted and interpolated macroblocks are coded differentially with respect to previously decoded motion vectors in order to reduce the number of bits required to represent them. The prediction MV is set to zero at the start of a slice or if the last macroblock was intra coded. The motion compensation process forms predictions from the previously decoded frames, using motion vectors that are of integer pel or half pel resolution. In

the case of an intra coded macroblock, no prediction is formed. They may however carry concealment motion vectors which are used to aid the decoding process when bit stream errors have occurred.

The differential motion vectors are coded using VLC. The motion vectors for the color components are obtained from the motion vectors of the luminance component as shown in eqs. 5.1-5.4.

For 4:2:0,

MV (horizontal chroma) = MV (horizontal luma) / 2 (5.1)

MV (vertical chroma) = MV (vertical luma) / 2 (5.2)

For 4:2:2,

MV (horizontal chroma) = MV (horizontal luma) / 2 (5.3)

MV (horizontal chroma) = MV (horizontal luma) (5.4)

There are four prediction modes in MPEG-2:

1. Field prediction: A frame is composed of two fields, the top field has parity zero and the bottom field has parity one. Field predictions can be made between the same parity fields or opposite parity field. Predictions are made independently for each field. In P-pictures the two reference fields from which predictions shall be made are the most recently decoded reference top field and the most recently decoded reference bottom field. The simplest case is shown in Fig. 5.2 . It is used when predicting the

first picture of a coded frame or when using field prediction within a frame picture. In this case the two reference fields are part of the same reconstructed frame.

The case when predicting the second field picture of a coded frame is more complicated because the two most recently decoded reference fields shall be used. In this case the most recent reference field was obtained from decoding the first field picture of the coded frame. Fig. 5.3 illustrates this situation when the second picture is the bottom field. Fig. 5.3 illustrates the situation when the second picture is the top field.

2. Frame prediction: In P pictures, predictions are made from the most recently reconstructed reference pictures as illustrated in Fig. 5.5.

3. (16x8) motion compensation: In this case, two motion vectors are used per macroblock. The first motion vector is used for the upper 16x8 region and the second motion vector is used for the lower 16x8 region. This method can be used by field pictures only.

4. Dual Prime prediction: This is present in P pictures only. Only one motion vector is encoded in the bit stream together with a small differential motion vector. In the case of field pictures two motion vectors are derived from this information. They are used to form predictions from two reference fields (one top field, one bottom field), which are averaged to form the final prediction.

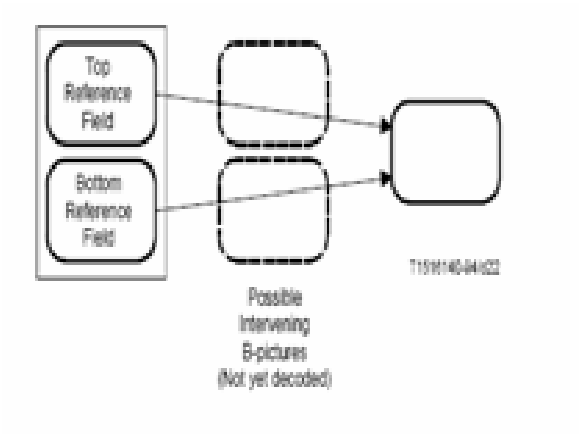


Fig. 5.2 Prediction of the first field or field prediction in a frame picture [6].

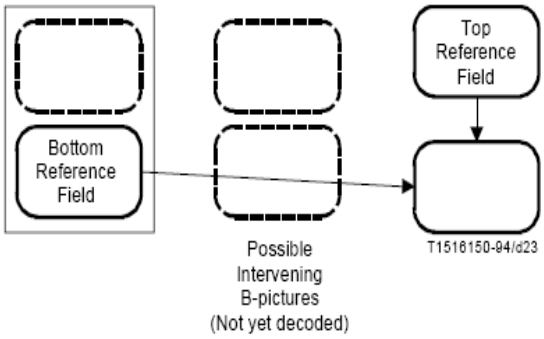


Fig. 5.3 Prediction of the second field picture when it is the bottom field [6].

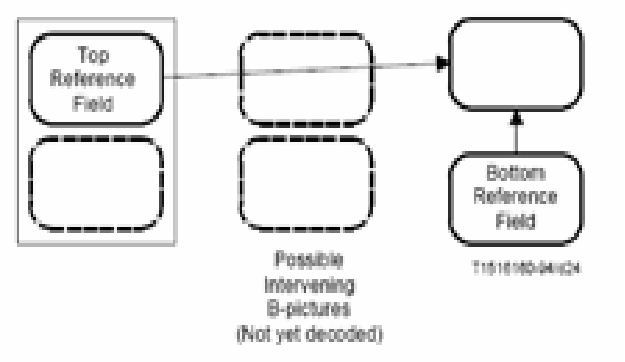


Fig. 5.4 Prediction of the second field picture when it is the top field [6].

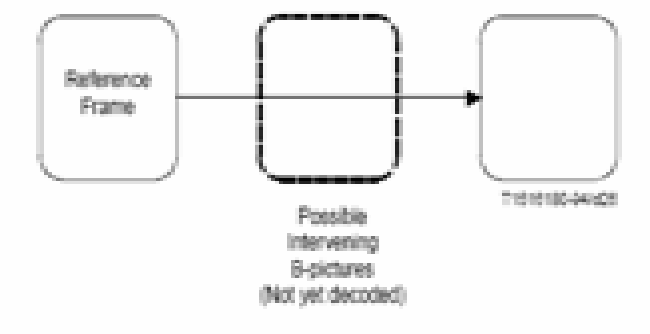


Fig. 5.5 Frame prediction of a P picture [6].

5.1.2 DCT coding

Only the DCT coefficients of the basic 8x8 blocks indicated by the coded block pattern (CBP) in a macroblock are coded. The DCT coefficients of the remaining blocks are quantized as zeros. After applying DCT, the coefficients are then quantized using the default quantization matrix (Fig. 5.6) for non-intra macro blocks or a user defined matrix. The transformed and quantized coefficients are then scanned and VLC coded.

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

Fig. 5.6 Default quantization matrix for 8x8 DCT coefficients in non-intra macroblocks [14].

5.2 P frame coding in H.264

P frame coding in H.264 [19] utilizes inter prediction similar to the MPEG-2 standard. Inter prediction includes both motion estimation (ME) [3] and motion compensation (MC) [3]. It generates a predicted version of a rectangular array of pixels, by choosing another similarly sized rectangular array of pixels from a previously decoded reference picture and translating the reference array to the position of the current rectangular array. In AVC, the rectangular array of pixels that are predicted using MC can have the following sizes: 4x4, 4x8, 8x4, 8x8, 16x8, 8x16 and 16x16 pixels (Fig. 5.11). Hence it can be noted that AVC allows sub 16x16 and sub 8x8 ME and MC unlike the MPEG-2 standard. Also, AVC allows ME of up to quarter pixel accuracy. It allows the use of more than one recently decoded reference frames from the reference frame buffer. It also supports the use of B frames as reference frames, however this feature is optional. Such multi reference prediction, quarter pixel accuracy ME and use of B frames as reference frames is not supported by the MPEG-2 standard.

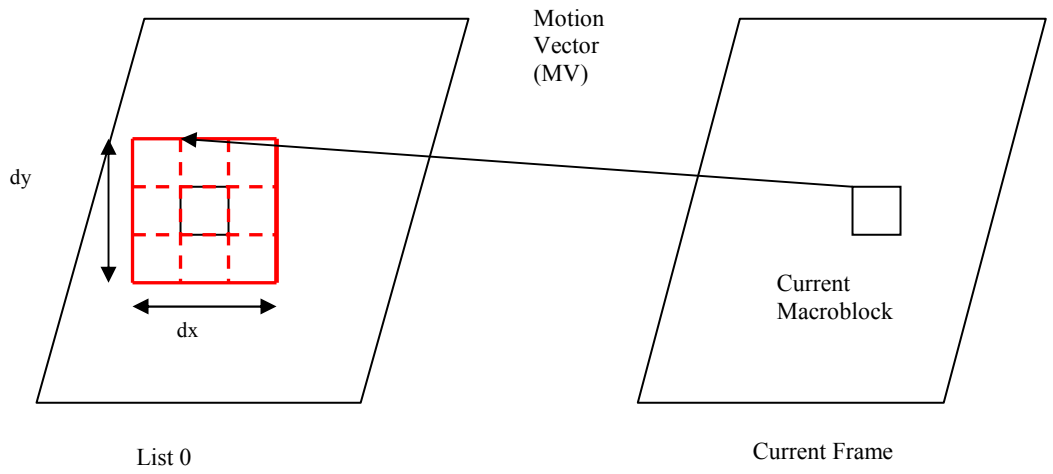


Fig. 5.7 Process of P frame prediction for the current macroblock over a search window dx x dy in a reference frame of List0.

Width of the search window, dx = 2*swx +1

Height of the search window, dy = 2*swy+1

AVC maintains two lists of reference frames, list0 and list1. P frames predict only from past decoded reference frames in List0 i.e., it uses forward prediction. For each pixel of each macroblock , it refers to each reference frame in list0, as shown in Fig. 5.7, and it computes the sum of absolute difference (SAD) value at each pixel within a user specified search window(swx – x axis search window size, swy – y axis search window size) in the reference frame. It selects the pixel in the reference frame which has the minimum SAD value as a search center. It then uses this search center to perform half pixel resolution ME. Again the pixel or sub pixel with the least SAD value

is selected as the search center and quarter pixel ME is performed. This process is repeated for each type of macroblock and sub macroblock partition, and each reference frame in List0 and then the best match with the minimum SAD value is selected. Depending upon the best mode or sub partition selected, the macroblock can have one motion vector or up to 16 motion vectors. It can be noted from the execution time in the test results table 4.1, that this full search method will yield the best results but it is very computationally intensive.

For instance, for motion vector prediction of a 16x16 pixel macroblock with no sub partitions, when estimated over a 1 pixel window, the size of the search window will be 3x3 with 9 pixels. At each of the 9 pixels, computing the SAD for the 16x16 macroblock requires 256 subtractions and 255 additions. Therefore a total of 256x9 subtractions and 255x9 additions before the motion vector with the lowest SAD value at full pel resolution can be derived. Then this is further extended to half pel resolution and quarter pel resolution over search windows of the same size or smaller sizes.

5.3 Transcoding

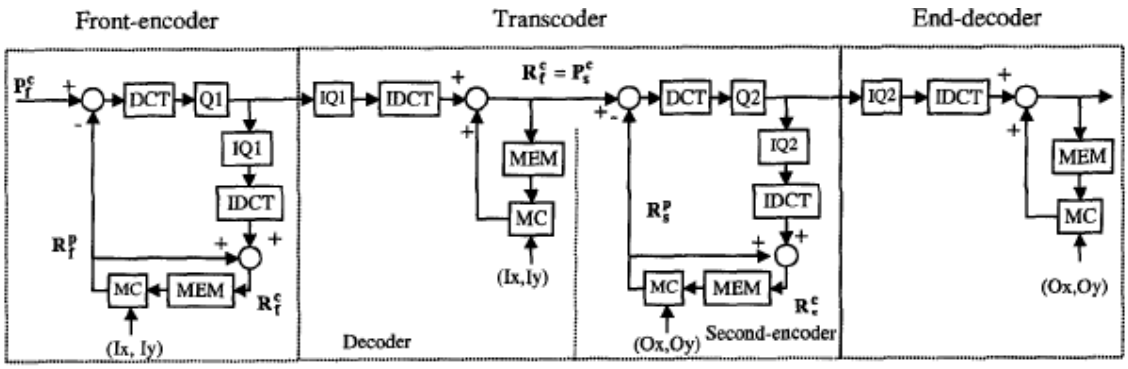


Fig. 5.8 Structure of the cascaded transcoder [4].

In most video coding standards, motion vector selection is performed based on the sum of absolute difference (SAD) value. In order to select a motion vector, the one corresponding to the best matching macroblock with the minimum SAD value over a search area 's' is selected such that

$$(Ox,Oy) = \arg \min_{(m,n)\in S} SAD_s(m,n) \tag{5.5}$$

where,

$$SAD_s(m,n) = \sum\sum \left| P_s^c(i,j) - R_s^p(i+m, j+n) \right|, \tag{5.6}$$

m, n are horizontal and vertical components of the motion vector.

The superscripts 'c' and 'p' denote current and previous frame respectively, as shown in Fig. 5.8. The subscript's' and 'f' indicate second encoder and front encoder respectively as shown in Fig. 5.8. Since the reconstructed picture in the front encoder is the same as the input picture to the second encoder,

$$SAD_s(m,n) = \sum_i \sum_j \left| P_f^c(i,j) - R_f^p(i+m, j+n) + \Delta_f^c(i,j) - \Delta_s^p(i+m, j+n) \right|. \quad (5.7)$$

where,

$$\Delta_f^c(i,j) = R_f^c(i,j) - P_f^c(i,j)$$
$$\Delta_s^p(i,j) = R_s^p(i,j) - P_s^p(i,j). \quad (5.8)$$

In the equations (5.7) (5.8), $\Delta_f^c(i,j)$ represents the quantization error in the front encoder and $\Delta_s^p(i,j)$ represents the quantization error of the previous frame in the second encoder. As can be seen, simple reuse of the MPEG-2 motion vector does not take into consideration the quantization error and hence it will produce non optimal results. The quality degradation will be especially significant when the difference between the quantization parameters of the first encoder and the second encoder is large.

5.3.1 Motion Vector Refinement

In transcoding, simple reuse of the motion vectors obtained from the MPEG-2 bit stream produces non optimal results due to quantization errors, leading to severe quality degradation. Motion vector refinement improves the accuracy of the motion vectors and allows for reuse, without significantly increasing the complexity. It also, helps to account for modes in MPEG-2 which do not have motion vectors coded. In the case of transcoding from MPEG-2 to H.264, MPEG-2 allows for up to half pel motion search where as H.264 allows for up to quarter pel motion search. Motion vector refinement allows taking advantage of quarter pel motion search and thus even improving the accuracy of motion search.

P frames can also contain Intra macroblocks. These are usually decided in MPEG-2 based on the macroblock content. However, since MPEG-2 does not evaluate sub macroblock partitions, there is a possibility that these macroblocks may be preferably coded as inter than intra. Using motion vector refinement with a (0, 0) search center, helps refine this decision.

The MPEG-2 standard allows for field coding or frame coding of frame pictures. If the motion vectors of the MPEG-2 stream are extracted from a frame macroblock which was field coded then one pixel compensation has to be applied to the value of the motion vectors. For instance, as shown in Fig. 5.9, when the macroblock was field predicted with the current macroblock in the top field and the referenced macroblock in the bottom field, one must be added to the motion vector (y component).

When the macroblock was field predicted with the current macroblock in the bottom field and the referenced macroblock in the top field, one must be subtracted from the motion vector (y component). This motion vector adjustment is performed prior to motion vector refinement.

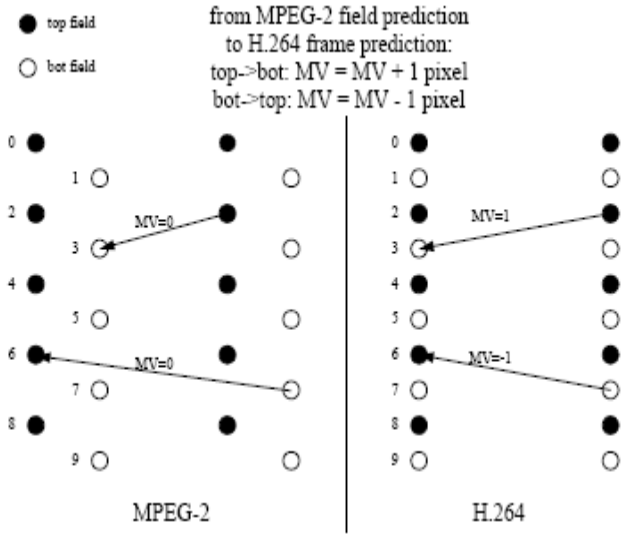


Fig. 5.9 Motion vector adjustment from field to frame prediction [2][4].

5.3.2 Algorithm

The refinement scheme (Fig. 5.10) applied to the motion vectors involves refining them over a one pixel window with the MPEG-2 motion vectors as the search centers. The motion vectors obtained by full pel search are then used as search centers for the half pel search. Similarly again the motion vectors from the half pel search are further refined by a quarter pel search.

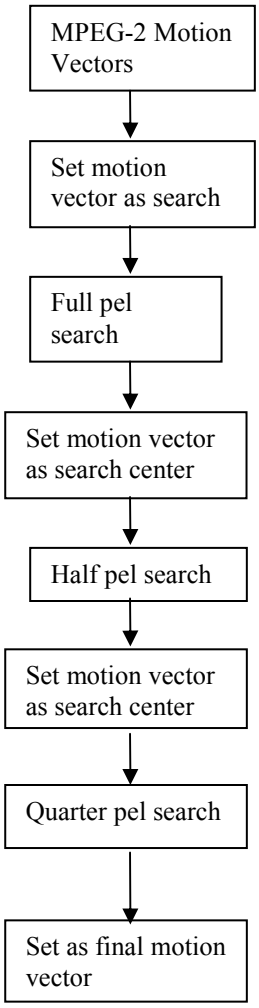


Fig. 5.10 Motion Vector Refinement Algorithm for P frames.

5.3.3 Coding mode decision

Coding mode decisions are made by comparing sum of absolute difference value i.e. motion cost for the refined motion vectors and the motion vectors predicted from the surrounding macroblocks in the same frame.

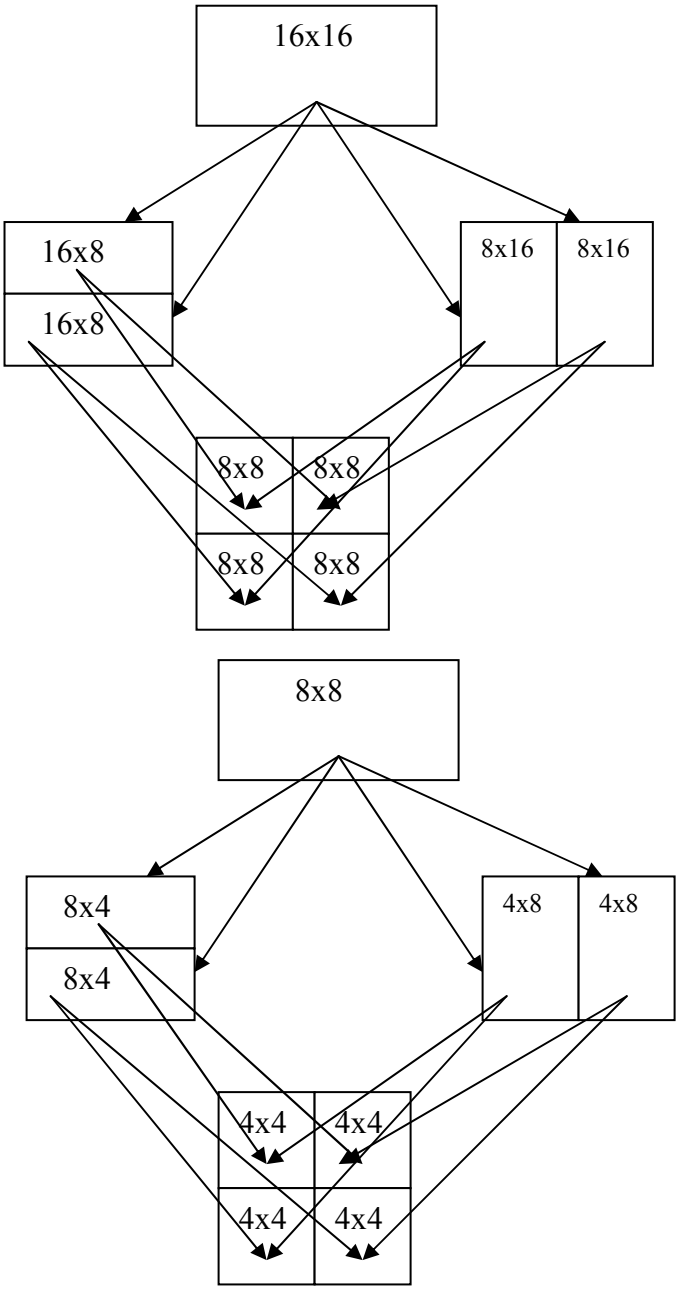


Fig. 5.11 Top down block splitting approach used to minimize the computational complexity of the coding mode decision process.

The motion cost for each pixel in a one pixel window is determined and the best one is selected based on the lowest cost as the best motion vector. To decide the best block size mode, top down splitting procedure is used as shown in Fig.5.11. Using this

top-down block splitting approach, initially the motion costs for the 16x16 block, the 8x16 blocks and the 16x8 blocks are determined. If the total motion cost for the 8x16 blocks or the 16x8 blocks exceeds the motion cost for the 16x16 block, then it implies that further partition may not significantly improve the performance. Hence the 8x8 and sub 8x8 block partitions are skipped from the motion estimation and mode decision process. Thus in general, if the costs for the next level of smaller sub blocks exceed that of the current level, further partitioning is stopped without checking the smaller blocks which come after the next level, to save computations. As shown in the Fig. 5.11, the same top-down approach is used for 8x8 blocks and sub 8x8 block partitions also. This helps to reduce the computational complexity of the coding mode decision process greatly.

5.4 Results and overview of P-frame transcoding

The overall scheme of P frame transcoding can be largely, divided into two parts: Extracting the frame data and motion vectors from the MPEG-2 bit stream, and adjusting the motion vectors and refine them over a small window and reuse them in sub pel motion estimation and mode decision.

The video clips used for this experiment are standard test clips [17]. The objective of the proposed scheme for P frame transcoding is to reduce the complexity of the transcoding process without significantly changing the PSNR .It can be observed in Table 5.1, that the PSNR obtained by complete decoding and re-encoding of the MPEG-2 bit stream is comparable that obtained by the proposed scheme. The PSNR also depends on other factors such as the motion search window size, bit rate etc. For these experiments the, bit rate was maintained constant at 1 Mbps and the motion search window for the full re-encoding process was maintained at -1pel to +1pel and -24 pels to +24 pels. The savings in terms of execution time are also quite significant.

Note that the Figs. 5.14a-5.25c indicate the motion vectors and the mode decisions for each test clip in all three scenarios; the original MPEG2 bit stream, the transcoded H.264 bit stream and the H.264 bit stream obtained by complete decoding and re-encoding of the input. The forward motion vectors are marked in red. In the case of B frames, the backward motion vectors are marked in green. The mode decisions are indicated by the black grid structure. In the case of bit stream MPEG-2, since I frames only support 16x16 block motion compensated transform coding, the grid indicates the

16x16 block squares. In the case of the H.264 bit stream, the cells of the grid are sub divided to indicate the kind of sub macroblock partition, if any.

Table 5.1 The results obtained by transcoding a P frame from MPEG-2 to H.264 at 1 Mbps, compared with the complete MPEG-2 decoding and H.264 re-encoding of the MPEG-2 bit stream.

		P frame with motion vector reuse and hierarchical mode decisions			P frame with full motion search , complete decoding and re-encoding		
Test Clip No.	Test Clip	PSNR (dB)	Number of Bits used	Motion estimation time (MET) (ms)	PSNR (dB)	Number of Bits used	Motion estimation Time (MET) (ms)
1	Crawfish	40.249	60080	631	37.349	50392	4196
2	Akiyo	41.952	13536	401	42.137	10464	3976
3	Coast guard	37.531	115544	751	37.93	94688	4136
4	Football	37.888	83440	691	37.89	90112	4407
5	Foreman	39.835	38280	581	40.09	32620	4016
6	Flower garden	36.882	212120	671	37.349	134880	4196
7	Hall Monitor	40.46	32264	350	40.517	29936	3916

It can be observed that the results obtained by the proposed method are very close to those obtained by the full re-encoding process. For the test clip Akiyo, the motion vectors are shown in Fig. 5.14a-5.14c and the mode decisions can be observed in Fig. 5.15a-5.15c. The mode decisions and the motion vectors computed are similar. The MPEG-2 motion vectors are coded for a 16x16 macroblock. Sub macroblock partitions are not part of the standard hence fewer motion vectors can be observed. The

refinement scheme helps to take advantage of sub macroblock modes and up to 50 percent more new motion vectors are determined. In this test clip, the motion is concentrated around the news readers face, hence the detailed motion vectors and sub macroblock decisions can be observed around these areas.

The same results can also be observed on test clips Flower garden (Fig.5.20a-5.21c), Foreman (Fig.5.16a-5.17c), Football (Fig. 5.22a-5.23c) and Coast (Fig. 5.18a-5.19c). It can be noted that the percentage of macroblocks choosing the sub 8x8 modes is far less in the proposed scheme than in the full re-encoding method, due to hierarchical mode decisions. Also, between MPEG-2 and H.264, the change in the decision between intra mode and inter mode can be observed, particularly in the test clip Foreman (Fig. 5.16a-5.17c), where the macroblocks in the face tend to select intra mode over the inter mode after they are transcoded, hence the motion vectors in those areas are zero.

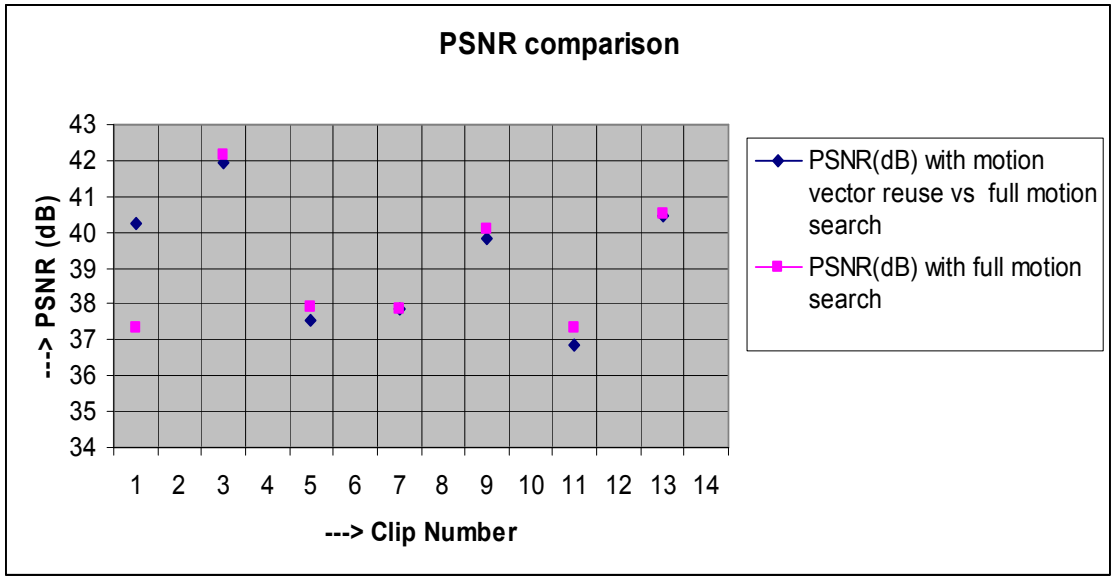


Fig. 5.12 Comparison of PSNR obtained by the proposed method of motion vector reuse vs. full motion search for P frames in different test clips.

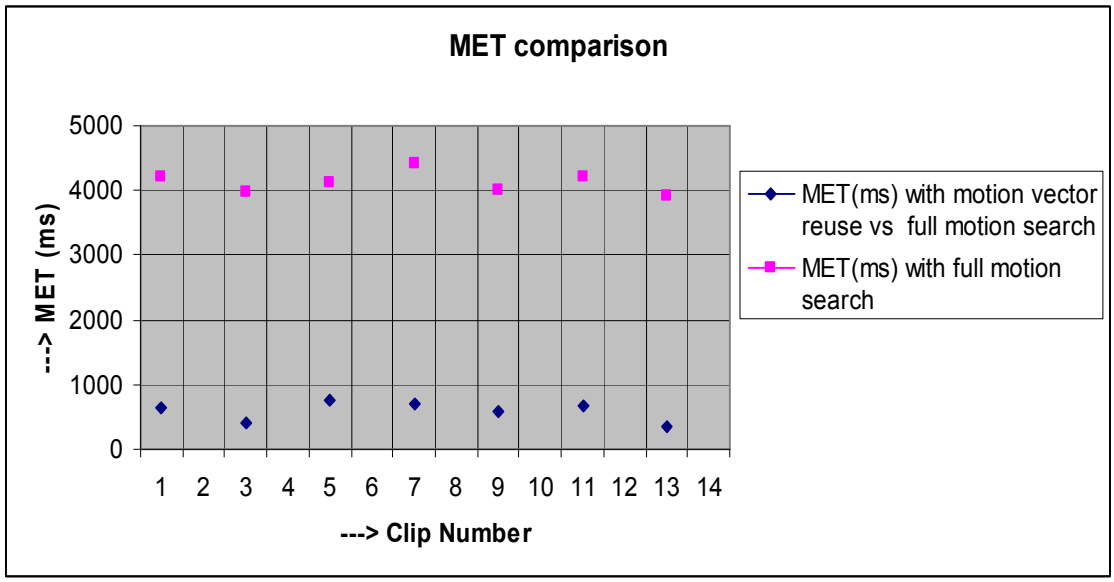


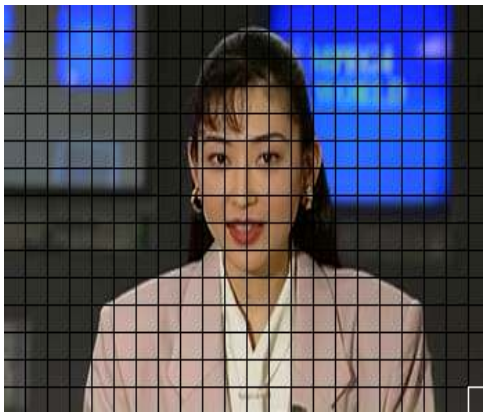
Fig. 5.13 Comparison of MET obtained by comparing the proposed method of motion vector reuse vs. full motion search for P frames in different test clips.

(a)

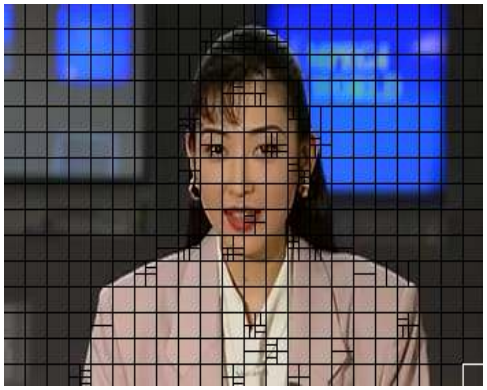
(b)

(c)

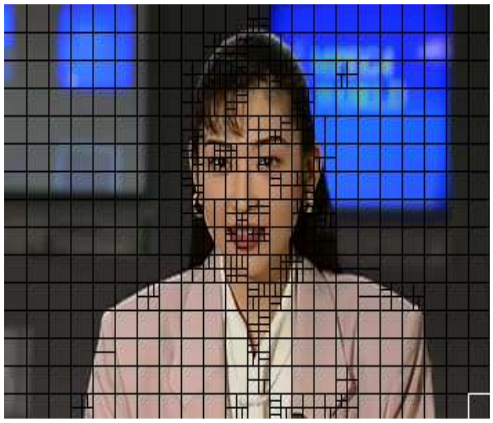
Fig. 5.14 a.Shows a P frame of the 1Mbps compressed MPEG-2 bit stream Akiyo with the motion vectors marked.b. Shows a P frame of the 1Mbps compressed transcoded H.264 bit stream Akiyo with the motion vectors marked.c. Shows a P frame of the 1Mbps compressed H.264 bit stream Akiyo obtained by complete decoding and re-encoding with the motion vectors marked.



(a)



(b)



(c)

Fig. 5.15 a. Shows a P frame of the 1Mbps compressed MPEG-2 bit stream Akiyo with the mode decisions marked.b. Shows a P frame of the 1Mbps compressed transcoded H.264 bit stream Akiyo with the mode decisions marked.c. Shows a P frame of the 1Mbps compressed H.264 bit stream Akiyo obtained by complete decoding and re-encoding with the mode decisions marked

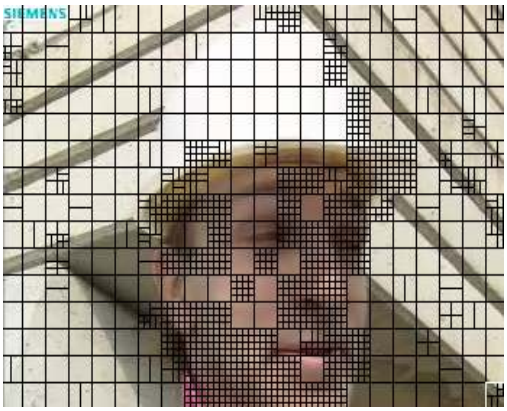
(a)

(b)

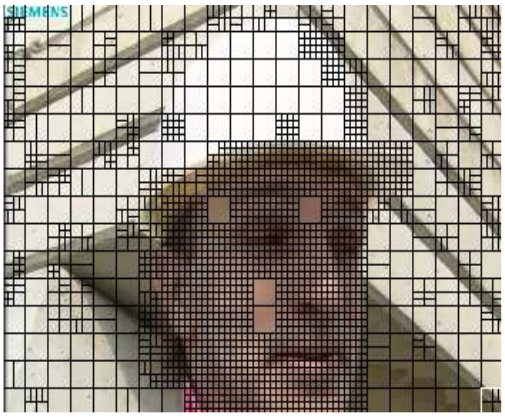
(c)

Fig. 5.16 a.Shows a P frame of the 1Mbps compressed MPEG-2 bit stream Foreman with the motion vectors marked.b. Shows a P frame of the 1Mbps compressed transcoded H.264 bit stream Foreman with the motion vectors marked.c. Shows a P frame of the 1Mbps compressed H.264 bit stream Foreman obtained by complete decoding and re-encoding with the motion vectors marked.

(a)



(b)



(c)

Fig. 5.17 a. Shows a P frame of the 1Mbps compressed MPEG-2 bit stream Foreman with the mode decisions marked.b. Shows a P frame of the 1Mbps compressed transcoded H.264 bit stream Foreman with the mode decisions marked. c. Shows a P frame of the 1Mbps compressed H.264 bit stream Foreman obtained by complete decoding and re-encoding with the mode decisions marked

CHAPTER 6

B FRAME TRANSCODING

The major difference between P frames and B frames is that there can be three types of prediction in B frames: forward, backward and interpolated prediction. There are two estimated motion vectors forward and backward [14]. The motion vectors are set to zero at the start of each slice and at each intra macroblock.

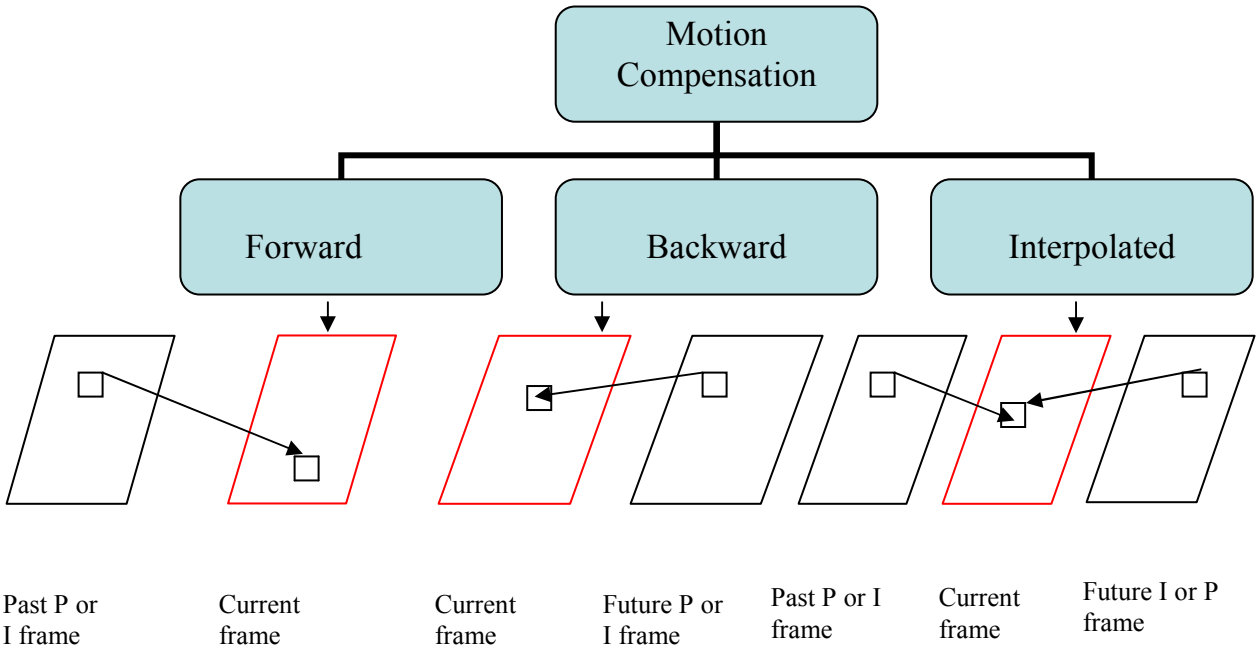


Fig. 6.1 Different coding modes for each macroblock in a B frame.

6.1 B frame encoding in MPEG-2

In the case of MPEG-2 [14], the encoder does not store B frames in the memory because they are not used for motion estimation and compensation.

There are three ways of coding the motion vectors. If only a forward motion vector is present, then the motion compensated macroblock is predicted from the previous I or P picture. This is the same as the P frame where motion compensation is based on forward motion vectors only. If only a backward motion vector is present, the motion compensated macroblock is predicted from a future I or P picture. If both forward and backward motion vectors are present then the motion compensated macroblocks are constructed from both the previous I or P frame and the next I or P frame and the result of the two is averaged to form the interpolated motion compensated macroblock.. As illustrated in Fig. 6.2, besides the intra mode and the predicted modes, the skip mode also exists for the B frames. However there exists one basic difference in the coding of skip macroblock motion vectors for B frames. Skipped macroblocks in P frames have motion vectors equal to zero; however, skipped macroblocks in B frames have the motion vector differential equal to zero, which implies that the motion vectors of the current macroblock are the same as that of the previous macroblock. The skipped macroblock in a B frame has no VLC. The sequential mode decision procedures in Fig. 6.2 lead to macroblock type selection. The encoder first determines the motion compensation mode i.e. forward, backward or interpolative. The MC/no MC decision is not necessary for B frame. The macroblocks are either regarded as motion compensated or intra. The second decision is intra or non intra coding. The intra macroblock is coded

similarly as in the I frame. For the non intra case, the prediction error is checked to see if it is large enough to be coded using the DCT. The last step is to decide whether the quantizer scale is satisfactory for the quantization of the DCT coefficients.

Blocks of 8x8 pels are transformed into an array of 8x8 transform coefficients using the 2-D DCT that is the same as that for I and P pictures. Quantization and coding of the DCT coefficients in B pictures is the same as in P pictures.

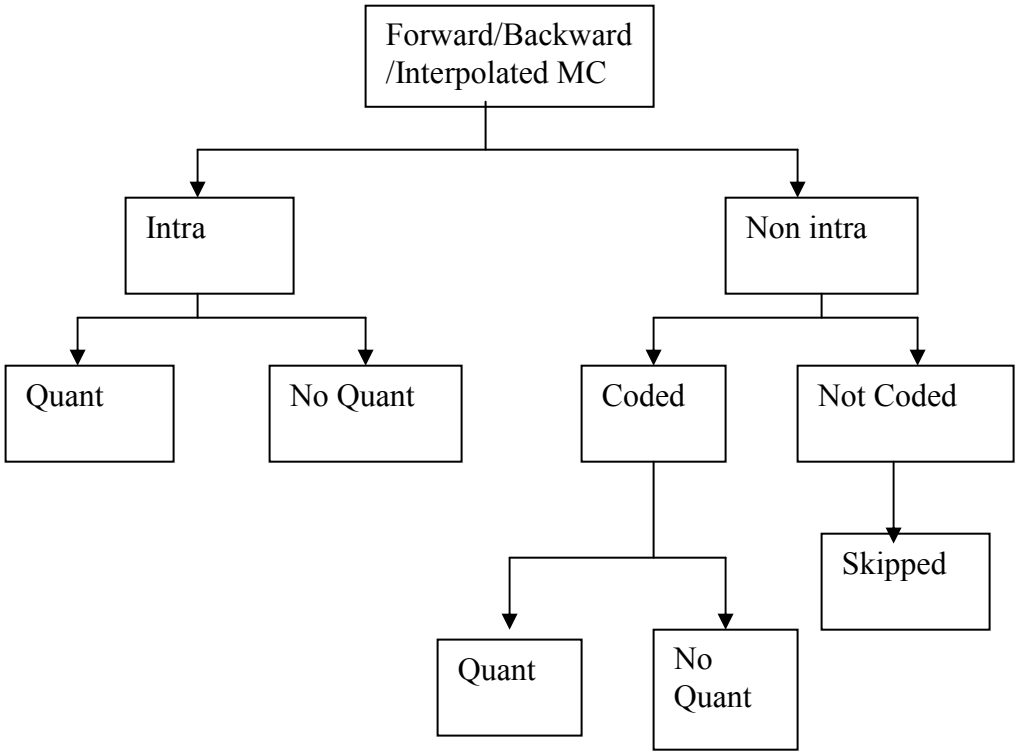


Fig. 6.2 Mode decision tree structure for macroblocks in B frames

6.2 B frame encoding in H.264

B frame encoding in H.264/AVC is similar to that of MPEG-2; however there are certain additional capabilities in it that surpass MPEG-2. H.264 supports motion estimation and compensation for sub macroblock partitions. For B frame macroblocks, each sub partition can have up to two motion vectors allowed for temporal prediction [3]. They can be from any picture in the future or the past in display order. Hence H.264 supports multi frame referencing which MPEG-2 does not support. There is a constraint on the maximum number of reference frames that can be used for prediction based on the profile and level being used [3]. Also, H.264 allows the use of B frames as reference frames for temporal prediction.

As compared to MPEG-2, B frames also have a special mode in H.264 called the direct mode. In this mode the motion vectors are not explicitly derived. The receiver obtains the motion vectors by scaling the motion vectors of a collocated macroblock in another reference picture. In this case, the reference picture for the current macroblock is the same as that of the collocated macroblock.

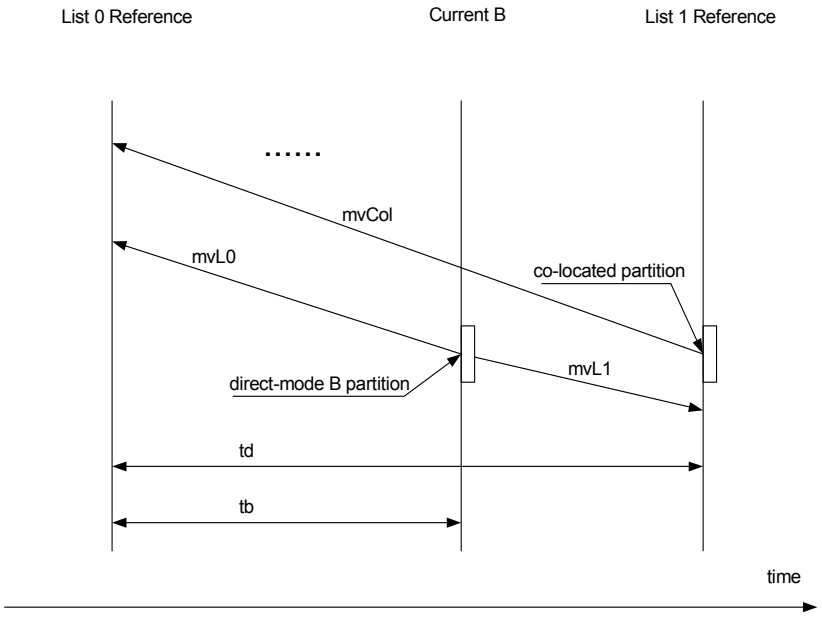


Fig. 6.3 Computation of motion vectors from the collocated macroblock for the direct mode in B frames [3][25].

The weighted prediction concept [11] is further extended in the case of B frames. Weighted prediction can be used to enable the encoder adjustment of the weighting used in the weighted average between the two predictions that apply to bi-prediction. This can be especially effective for implementing "cross-fades" between to different video scenes, as bi-prediction allows flexible blending of content from these two scenes. The rest of the processing of B frames in H.264 remains the same as, in the case of P frames.

6.3 Transcoding of B frames and results

Transcoding of B frames also requires motion vectors reuse and refinement [1][4][5], as in the case of P frames. The proposed method refines them over a -1 to +1 pixel window around the current pixel pointed to by the motion vectors. This process is

repeated for the forward motion vector, as well as, the backward motion vector. The search window size around the current pixel pointed to as defines the accuracy of the refinement. Several tests were executed using different search window sizes and the results are as shown in Fig. 6.4.

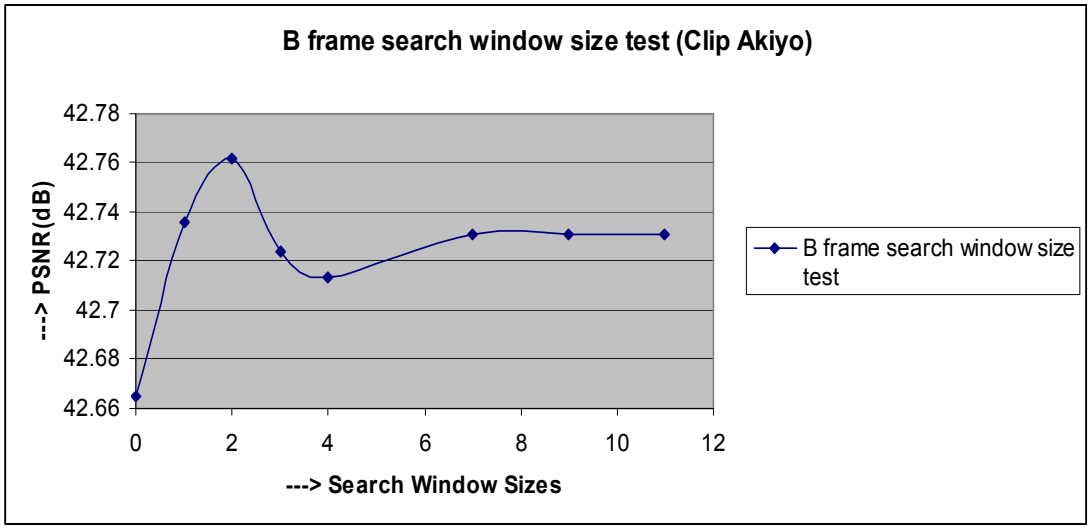


Fig. 6.4 Effect of the choice of search window sizes on PSNR, when tested on the Clip Akiyo transcoded from a 1Mbps MPEG-2 stream to a 699Kbps H.264 stream.

It can be observed that as the search window size is increased, initially the PSNR increases, however the increase tends to saturate to a steady state value above a certain search window size. It can also be noted that the one pixel window search provides a value close to the steady state PSNR value. Also, it involves the use of nine search centers which gives a fairly good tradeoff between computational complexity and PSNR. Similar results were obtained, when other test sequences were also tested. Hence a -1 to +1 pixel window was selected as a reasonable tradeoff.

Further, as in the case of P frames the use of hierarchical mode decisions was also tested for different test sequences and the results for test sequence Akiyo are as tabulated in Table 6.1, Table 6.2, Fig.6.5 and Fig.6.6. It can be observed that the PSNR

in the two cases is almost the same however; the execution time for the proposed method reduces by approximately 50% with the use of hierarchical mode decisions.

Table 6.1 Comparison of the PSNR, Bit rate and Execution time for the test sequence Akiyo when transcoded from a 1 Mbps MPEG-2 stream to an H.264 stream using the proposed method.

Test clip Akiyo		
PSNR(dB)	Bit rate(kbps)	Execution time(ms)
32	137.46	2422
35.34	224.23	2815
38.58	379.86	2752
42.58	695.34	1070
45.31	1044.62	1071
49.11	1685.9	1062
52.93	2534.16	942

Table 6.2 compares the PSNR, Bit rate and Execution time for the test sequence Akiyo when transcoded from a 1 Mbps MPEG-2 stream to an H.264 stream using the proposed method without hierarchical mode decision.

Test clip Akiyo w/o hierarchical mode decision		
PSNR(dB)	Bit rate(kbps)	Execution time(ms)
32	137.46	4674
35.36	224.94	5426
38.59	380.37	5750
42.6	699.18	2434
45.33	1050	2463
49.14	1705.81	2353
52.95	2538.77	2352

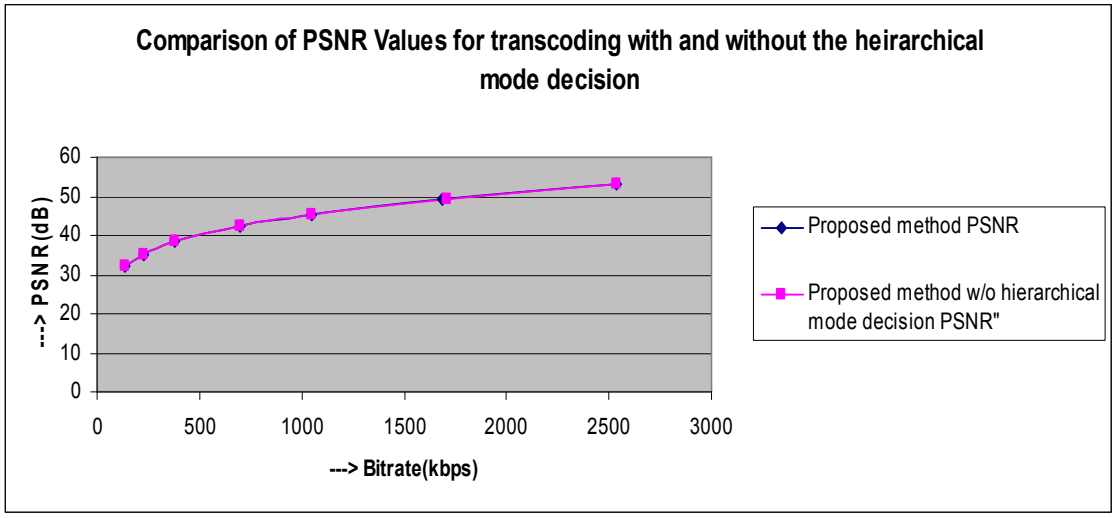


Fig. 6.5 Comparison of the PSNR in dB for the proposed reuse method with and without hierarchical mode decision.

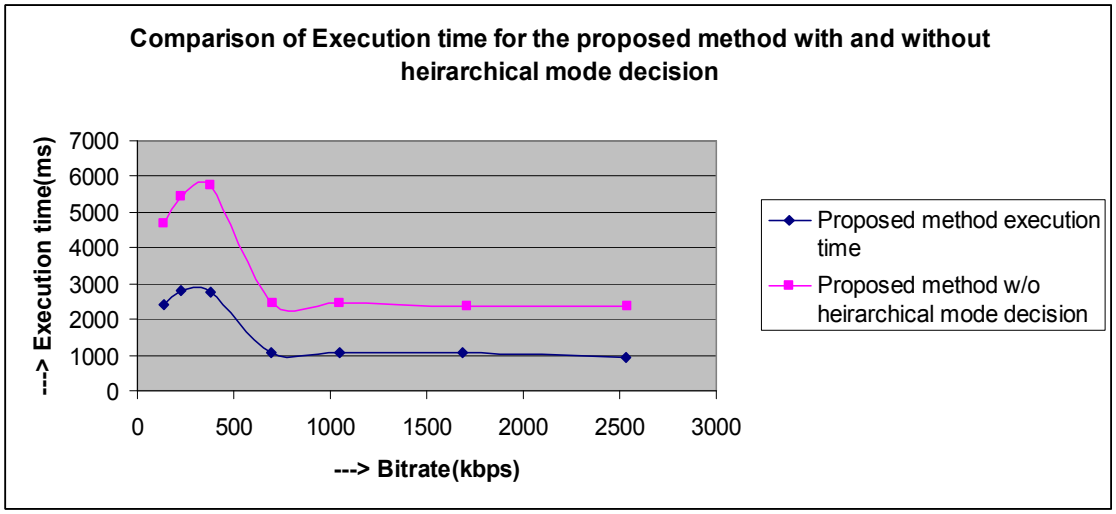


Fig. 6.6 Comparison of the execution time in ms for the proposed reuse method with and without hierarchical mode decision.

The overall results while transcoding a B frame for different test clips are tabulated in Table 6.3.

Table 6.3 Comparison of the results obtained by transcoding different test sequences from a 1Mbps MPEG-2 stream to an H.264 stream at a lower bit rate with those obtained by complete decoding and re-encoding of the same MPEG-2 stream.

		B frame with motion vector reuse and hierarchical mode decisions			B frame with full motion search , complete decoding and re-encoding		
Test Clip Number	Test Clip	PSNR (dB)	Number of Bits used	Motion estimation time(MET) (ms)	PSNR (dB)	Number of Bits used	Motion estimation Time(MET) (ms)
1	Crawfish	40.466	55032	881	40.830	43248	8112
2	Akiyo	42.712	5416	660	42.769	4456	7961
3	Coastguard	37.534	95432	872	38.035	66912	7801
4	Football	37.922	74696	801	37.905	71776	7993
5	Foreman	40.369	27768	881	40.545	21752	7800
6	Flower garden	36.854	179328	771	37.202	105664	8013
7	Hall Monitor	41.175	15528	660	41.210	13368	7482

(a)

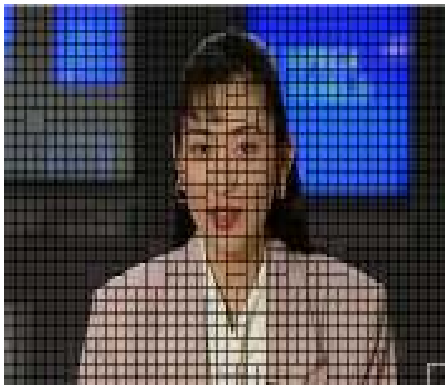
(b)

(c)

Fig. 6.7 a.Shows a B frame of the 1Mbps compressed MPEG-2 bit stream Akiyo with the forward and backward motion vectors marked. b. Shows a B frame of the 1Mbps compressed transcoded H.264 bit stream Akiyo with the forward and backward motion vectors marked.c. Shows a B frame of the 1Mbps compressed H.264 bit stream Akiyo obtained by complete decoding and re-encoding with the forward and backward motion vectors marked

(a)

(b)

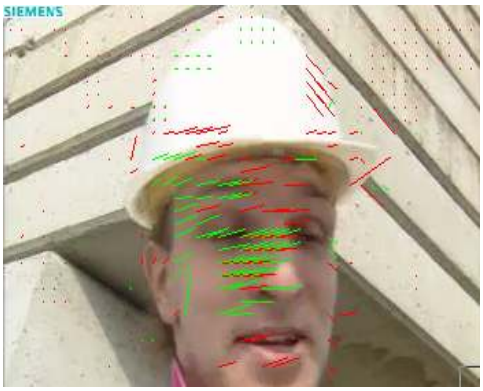


(c)

Fig. 6.8 a. Shows a B frame of the 1Mbps compressed MPEG-2 bit stream Akiyo with the mode decisions marked..b. Shows a B frame of the 1Mbps compressed transcoded H.264 bit stream Akiyo with the mode decisions marked.c. Shows a B frame of the 1Mbps compressed H.264 bit stream Akiyo obtained by complete decoding and re-encoding with the mode decisions marked

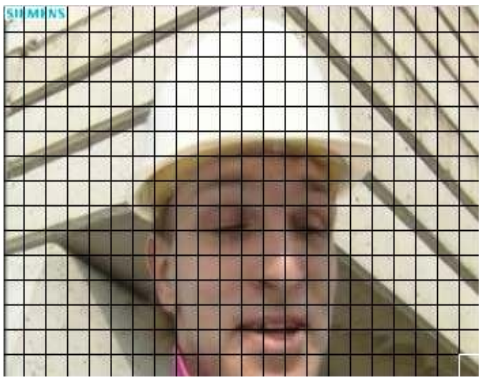
(a)

(b)

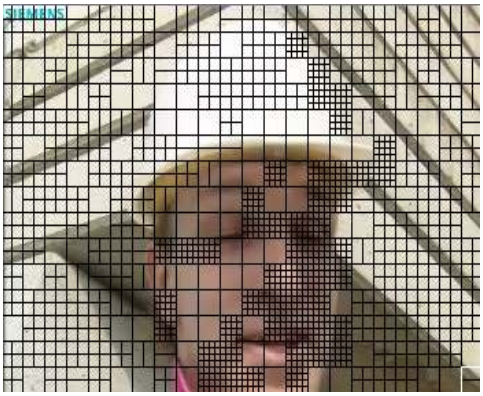


(c)

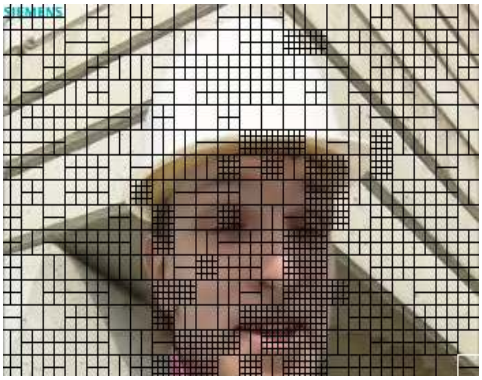
Fig. 6.9 a.Shows a B frame of the 1Mbps compressed MPEG-2 bit stream Foreman with the forward and backward motion vectors marked. b. Shows a B frame of the 1Mbps compressed transcoded H.264 bit stream Foreman with the forward and backward motion vectors marked.c. Shows a B frame of the 1Mbps compressed H.264 bit stream Foreman obtained by complete decoding and re-encoding with the forward and backward motion vectors marked.



(a)



(b)



(c)

Fig. 6.10 a. Shows a B frame of the 1Mbps compressed MPEG-2 bit stream Foreman with the mode decisions marked. b. Shows a B frame of the 1Mbps compressed transcoded H.264 bit stream Foreman with the mode decisions marked. c. Shows a B frame of the 1Mbps compressed H.264 bit stream Foreman obtained by complete decoding and re-encoding with the mode decisions marked.

CHAPTER 7

RESULTS AND CONCLUSIONS

The results presented so far compared the proposed transcoding method with complete decoding and re-encoding of the input MPEG-2 bit stream. As has been observed, the two methods are very close and comparable in PSNR. However, a significant reduction in the execution time is achieved by the proposed method. The results presented here are a comparison between the properties of the input MPEG-2 bit stream and the output H.264 bit stream.

$$\text{Cost} = \Lambda * \text{Rate} + \text{Distortion} \quad (7.1)$$

where,

Λ = Lagrange multiplier.

Rate = the total number of bits used to transmit the transform coefficients and header information.

Distortion = the difference between the original and the reconstructed macroblock.

The bit rate of the output H.264 stream is maintained the same as the input MPEG-2 bit stream by controlling the quantization parameter of the H.264 bit stream. The rate control engine of JM9.4 depends largely on the initial value of the quantization parameter (QP) at the beginning of each GOP and the past values of QP in the QP buffer. By varying the initial QP of each GOP, the bit rate of the output stream, as well

as, the quality of the frames in that GOP can be influenced. This rate control cannot be directly affected by the mode decision module because the QP selection process has greater influence on the output bit rate. The mode decision process, however, computes the sum of absolute difference value which has a direct correlation with the distortion computed. The results (Table 7.1) are without rate–distortion optimization (RDO). They can be further improved by using RDO. When RDO is used, the cost of each mode decision and motion estimate is computed using the eq. (7.1).To compute the values of the 'Rate' and the 'Distortion' term the block or macroblock has to be completely reconstructed for each subpartition. Hence the complexity associated with using RDO is much higher.

Rate control is responsible for maintaining consistent quality while satisfying bandwidth, delay and memory constraints. In the case of transcoding it is best to retain the GOP structure of the input stream to take advantage of the bit distribution between frames. Transcoding does not typically change the frame types to keep the complexity low. One solution is to scale the bits in the incoming bit stream according to the rate conversion ratio. The rate conversion ratio is the ratio of the bit rate of the transcoded output bit stream to the bit rate of the input bit stream. Also, selective scaling can be applied to assign more bits to the I frames to obtain greater accuracy in intra coding. When an initial delay is incorporated in the transcoding process, the statistics of the incoming bit stream can be used to improve the rate control [27].

7.1 Results

The test clips used are standard (352x240) CIF resolution test clips. They are encoded into MPEG-2 streams at a bit rate of 1Mbps and with a GOP size of 12 and the IBBPBBP…GOP structure. These streams are transcoded to 1Mbps H.264 streams with the same GOP structure of IBBPBBP... and a GOP size of 12. The PSNR values presented are the average PSNR values for all the frames in the test sequence used. The PSNR is computed with reference to the original source clip before the MPEG-2 encoding process.

Table 7.1 Comparison of the PSNR of the input MPEG-2 bit stream and the H.264 transcoded bit stream measured with reference to the original source test clip.

Test Clip	PSNR of the MPEG-2 bit stream (dB)	PSNR of the H.264 transcoded bit stream (dB)
Flower	30.15	27.11
Crawfish	36.95	33.1
Football	28.12	26.11
Foreman	37.46	34.13
Akiyo	43.68	41.04
Coast	32.2	28.63
Hall Monitor	36.93	34.01

As can be observed in Table 7.1, transcoding results in a reduction of about 2-4 dB in the PSNR of the input bit stream. However, perceptually this reduction is not significantly visible as can be observed in the Figs. 7.3-7.5.

To derive conclusions about the performance of the proposed method it is very important to compare it with other proposed methods for transcoding. Figure 7.2 compares the proposed method (PM) with a DCT domain transcoder (DDT) [23] and with complete decoding and re-encoding (CDRE) of the MPEG-2 bit stream. The test is run on 30 frames of the test sequence Foreman at a bit rate of 1 Mbps and with a GOP structure of IBBPBBP… The DCT domain transcoder used is shown in Fig. 7.1.

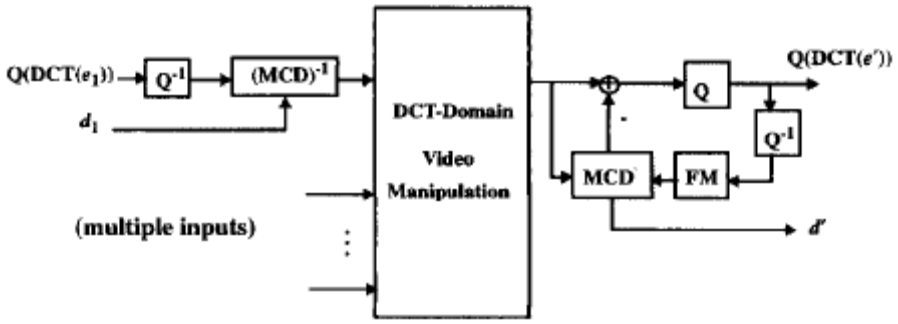


Fig. 7.1 DCT domain transcoder proposed by Chang and Messerschmitt [23].

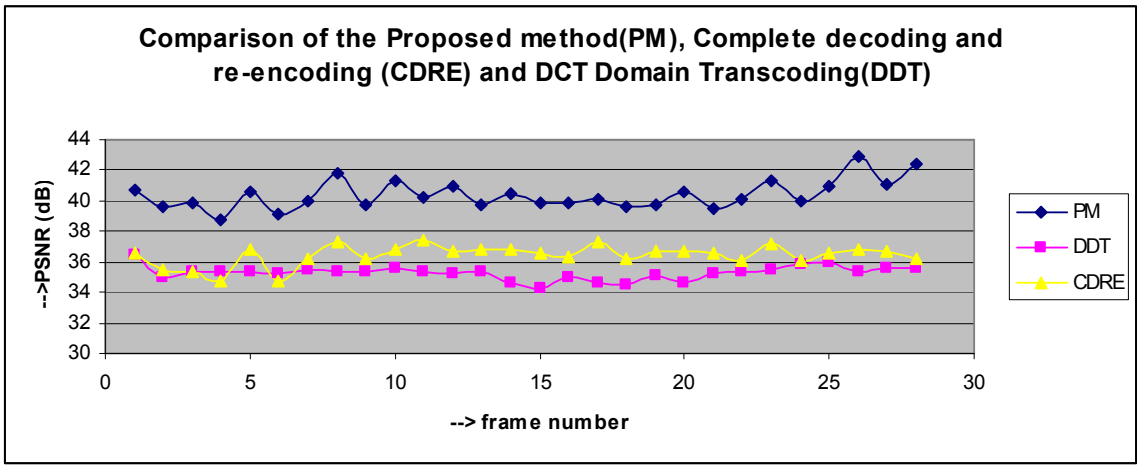


Fig. 7.2 Comparison of the proposed method (PM) with a DCT domain transcoder (DDT) and with complete decoding and re-encoding (CDRE) of the 1 Mbps MPEG-2 bit stream Foreman.

The proposed method clearly performs better than the complete decoding and re-encoding method. Also, both the proposed method and CDRE perform better than transcoding in DCT domain. In terms of complexity, the average encoding time for each frame type is given in Table 7.2 for all the three methods.

Table 7.2 Comparison of the time (ms) to transcode a 1 Mbps input MPEG-2 bit stream Foreman to an H.264 bit stream at the same bit rate with the same GOP structure using PM, CDRE and DDT.

	PM	DDT	CDRE
I frame	681	-NA-	781
P frame	2524	2521	10218
B frame	3152	3987	19505

The proposed method is very efficient in terms of encoding time. It is close and comparable to the DCT domain transcoder. However, CDRE is very computationally intensive as the method re-computes motion vectors, mode decisions etc without taking advantage of the data already present in the input MPEG-2 bit stream.

The subjective quality of the transcoded bit stream can be compared for each frame type in Figs.7.3-7.5. It can be noted that perceptually the frames are very close in quality.

Fig. 7.3 Comparison of the subjective quality of the input I frame (left) and the transcoded output I frame (right) for the test sequence Foreman transcoded at 1 Mbps.

Fig. 7.4 Comparison of the subjective quality of the input P frame (left) and the transcoded output P frame (right) for the test sequence Foreman transcoded at 1 Mbps.

Fig. 7.5 Comparison of the subjective quality of the input B frame (left) and the transcoded output B frame (right) for the test sequence Foreman transcoded at 1 Mbps.

7.2 Conclusions

This thesis is based on transcoding of an MPEG-2 bit stream to an H.264 bit stream. It makes use of both transform domain as well as pixel domain. The proposed algorithm provides a very low complexity and fast transcoding technique with fairly acceptable reduction in the PSNR. It incorporates the benefit of speed like a DCT domain transcoder and good quality like a CDRE pixel domain transcoder. Thus all the expectations of a good transcoder are met.

7.3 Future research

The research presented in this thesis is directed at low complexity, speed and comparable quality. Now that the above targets have been achieved, the transcoder can be optimized for use in specific applications.

For use in wireless environments, the major constraints would be adaptive network bandwidth usage, reduced spatial resolution and reduced frame rate. The proposed transcoder extracts the transform coefficients and auxiliary information and hence can be easily used to incorporate these requirements.

Also, a strong error resilient rate control engine can be developed for the transcoder so that the bit rate of the transcoded stream can be varied as desired. Some scenarios require bit rate reduction transcoding techniques to accommodate changes in network bandwidth or variable bit rate transcoding for applications such as DVD recording.

REFERENCES

[1] J. Youn and M-T. Sun , “Motion Vector Refinement for high-performance transcoding”, in IEEE Int. Conf. Consumer Electronics, Los Angeles, CA, Vol. 1, Issue 1, pp. 30-40, March 1999.

[2] J. Xin, C-W. Lin and M-T. Sun, “Digital Video Transcoding” , Proceedings of the IEEE, Vol. 93, pp. 84-97, Jan. 2005.

[3] T. Wiegand et. al., “Overview of the H.264/AVC Video Coding Standard”, IEEE Trans. CSVT, Vol. 13, pp. 560-576, July 2003.

[4] A. Vetros, C. Christopoulos and H. Sun, “Video transcoding architectures and techniques: an overview”, IEEE Signal Processing magazine, Vol. 20, pp. 18-29,March 2003.

[5] H. Kalva, “Issues in H.264/MPEG-2 Video Transcoding”, IEEE Consumer Communications and Networking Conf., CCNC 2004, pp 657-659, Jan 2004.

[6] Information Technology-Generic coding of moving pictures and associated audio information: Video, ITU-T Rec. H.262 (2000 E).

[7] B. Haskell, A. Puri and A. Netravali, “Digital Video: an introduction to MPEG-2”, N.Y. Chapman and Hall, International Thomson Pub., 1997.

[8] G. Chen et. al., “Efficient block size selection for MPEG-2 to H.264 transcoding”, Proceedings of the 12th annual ACM International Conference on Multimedia, pp. 300-303, Oct. 2004

[9] MPEG-2 software (version 12) from MPEG software simulation group, http://www.mpeg.org/MPEG/MSSG/#source

[10] H.264 Software (JM9.5) from http://iphome.hhi.de/suehring/tml/download/jm94.zip

[11] A. Puri, X. Chen and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard", Signal processing: Image communication, Vol. 19, pp. 793-849, Oct. 2004.

[12] B. Shen and I. Sethi, "Direct feature extraction from compressed images", SPIE: Vol. 2670 Storage and Retrieval for Image Databases IV, pp. 404-414, 1996.

[13] Commercially available transcoders, PSP Video 9, http://www.pspvideo9.com

[14] K.R. Rao and J. J. Hwang, "Techniques and Standards for Image, Video and Audio coding", Upper Saddle River, N.J.: Prentice Hall, 1996.

[15] M. Ghanbari, "Video Coding: an introduction to standard codecs", London, U.K.: Institution of Electrical Engineers, 1999.

[16] I. E. G. Richardson, "H.264 and MPEG-4 video compression: video coding for next generation multimedia", Chichester: Wiley, 2003.

[17]Test streams obtained from ftp://ftp.tek.com/tv/test/streams/Element/MPEG-Video/525/ and http://www.cipr.rpi.edu/resource/sequences/sif.html

[18] Y-J. Chuang, Y-C. Huang and J-L Wu, “An efficient block algorithm for splitting an 8x8 DCT into four 4x4 modified DCT used in AVC/H.264”, EURASIP 2005, pp. 311-316.

[19] P. Assunco and M. Ghanbari, “Post Processing of MPEG-2 coded video for transmission at lower bit rates”, Proc. IEEE ICASSP, pp. 1998-2001, Atlanta, GA, 1996.

[20] T. Shanableh and M. Ghanbari, “Transcoding Architectures for DCT domain heterogeneous video transcoding”, Proc. IEEE ICIP, Vol. 1, pp. 433-436, Thessaloniki, Greece, Sept. 2001,.

[21] J. Xin, M.T. Sun and K. Chun, “Motion re-estimation for MPEG-2 to MPEG-4 simple profile transcoding”, Proc. Int. Workshop Packet Video, Pittsburgh, PA, Apr. 2002.

[22] D-Y. Chan, S-J. Lin and C-Y. Chang, “A rate control scheme using Kalman filtering for H.263”, Journal of Visual Communication and Image Representation, Vol. 16, pp. 734-748, Dec. 2005.

[23] S. Liu and A. Bovik, “Foveated embedded DCT domain video transcoding”, Journal of Visual Communication and Image Representation, Vol. 16, pp. 643-667, Dec. 2005.

[24] I. E. G. Richardson, “Video codec design: developing image and video compression systems”, Chichester: Wiley, 2002.

[25] G. Sullivan, T. Wiegand and A. Luthra, "Draft of Version 4 of H.264/AVC (ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 part 10) Advanced Video Coding)", JVT Doc., 14th Meeting: Hong Kong, China 18-21 Jan. 2005.

[26] G. F-Escribano et.al., "Computational complexity reduction of intra frame prediction in MPEG2/H.264 video transcoders", ICME, pp. 707-710, July 2005.

[27] I. Ahmad et. al., "Video transcoding: an overview of various techniques and research issues", IEEE Trans. on multimedia, vol. 7, pp. 793-804, Oct. 2005.

[28] S. Benyaminovich, O. Hadar and E. Kaminsky, "Optimal transrating via DCT coefficients modification and dropping", ITRE, pp. 100-104, June 2005.

[29] J-R. Ohm, "Advances in scalable video coding", Proc. IEEE, Vol. 93, pp. 42-56, Jan. 2005.

[30] J. Wang et. al., "An AVS to MPEG-2 transcoding system", Proc. of ISIMP, pp.302-305, Oct. 2004.

[31] J. McVeigh et. al., "A software based real-time MPEG-2 video encoder", IEEE Trans. CSVT, Vol. 10, pp. 1178-1184, Oct. 2000.

[32] J. Yang et. al., "A rate control algorithm for MPEG-2 to H.264 real-time transcoding" , VCIP, Vol. 5960, pp. 1995-2003, July 2005.

[33] Z. He, M. Bystrom and S. Nawab, "Conversion between DCT coefficients of blocks and their sub-blocks", VCIP, Vol. 5960, pp. 1979-1986, July 2005.

[34] H. Chen, C. Chen and P-H Wu, "Transform-domain intra prediction for H.264", ISCAS, Vol. 2, pp. 1497-1500, May 2005.

BIOGRAPHICAL INFORMATION

Rochelle Pereira is currently studying at the University of Texas at Arlington. During the course of her studies she has worked under the invaluable guidance of Dr. K. R. Rao. She has also been an intern at Broadcom Corporation, MA for eight months (Oct. 2004-May. 2005). She graduated with a Bachelor of Engineering degree in Electronics and Telecommunication from the University of Mumbai, India in 2002. She is a member of IEEE and Tau Beta Pi. She received the Master of Science degree in Dec. 2005.