TEXAS INSTRUMENTS 75 YEARS OF INNOVATION

Overview of the H.264/AVC

Dinesh Kumar (Senior Design engineer)
Pavan Shastry (Design engineer)
Anirban Basu (Senior Design engineer)
MMCodec TI India

8th Texas Instruments Developer Conference India
30 Nov - 1 Dec 2005, Bangalore

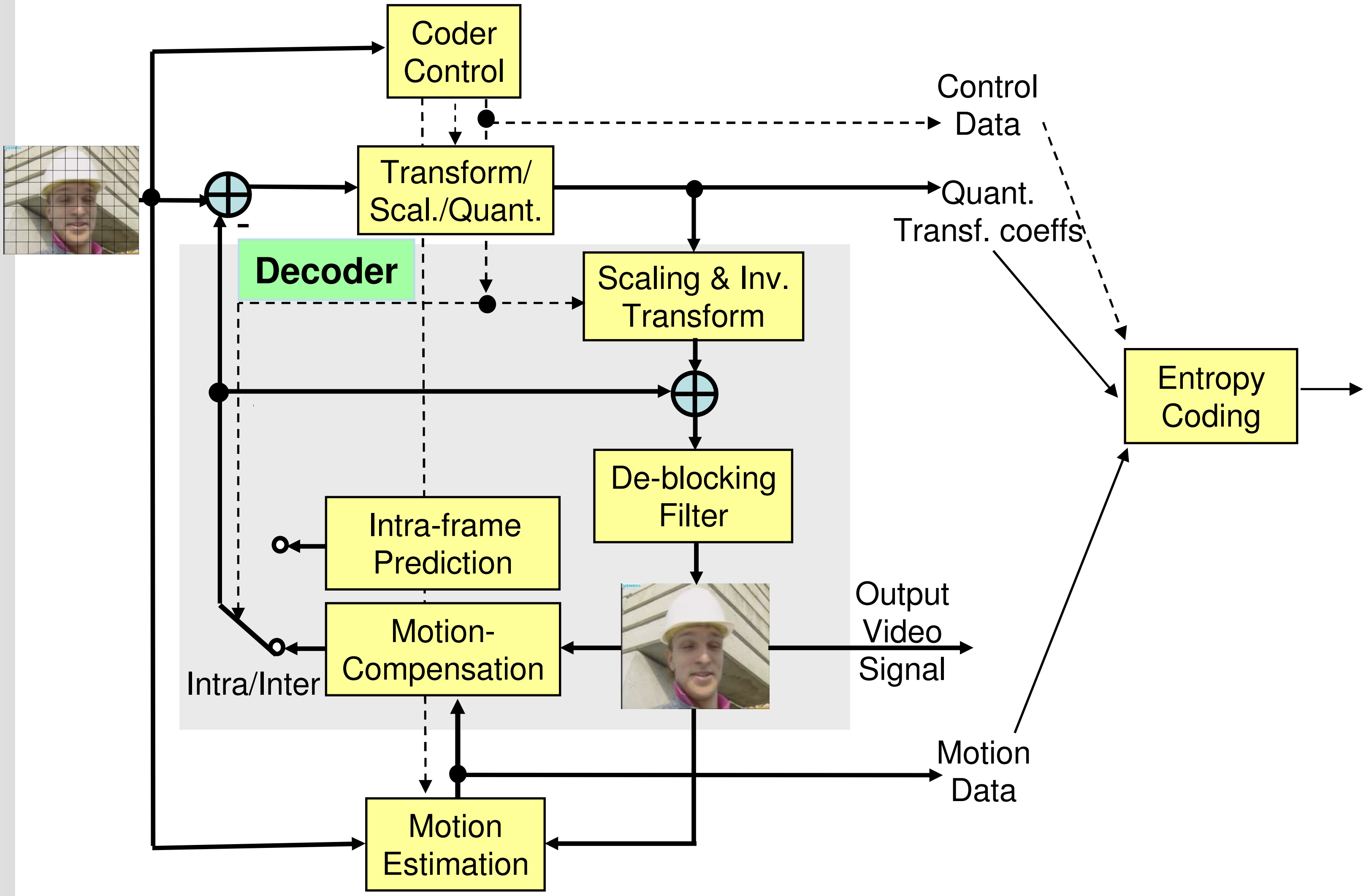
Agenda

- Scope and Context
- H.264 overview
- Profiles
- New Features
- Comparison with existing standards
- AVC coding tools
  - CAVLC
  - CABAC
  - Intra predication
  - Loop Filter

Scope and Context

- H.264/AVC is a joint project of ITU and MPEG,
- Standard defines:
  - Decoder functionality (but not encoder)
  - File and stream structure
- It aims to provide high-quality compression for various services:
  - IP streaming media (50-1500 kbps)
  - SDTV and HDTV Broadcast and video-on-demand (1 - 8+ Mbps)
  - DVD
  - Conversational services (<1 Mbps, low latency)

H.264 Encoder block diagram



Profiles

H.264 has three profiles:

- Baseline (IP Video phone, Simple streaming)
- Main Profile (Broadcast, VOD)
- Extended profile (Streaming profiles)

Profiles

There are four High Profiles (Fidelity range extensions)

- High Profile is to support the 8-bit video with 4:2:0 sampling for applications using high resolution.
- High 10 Profile is to support the 4:2:0 sampling with up to 10 bits of representation accuracy per sample.

Profiles

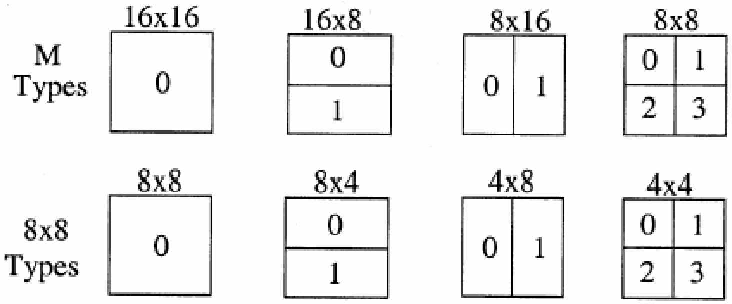
– High 4:2:2 Profile is to support up to 4:2:2 chroma sampling and up to 10 bits per sample.

– High 4:4:4 Profile is to support up to 4:4:4 chroma sampling, up to 12 bits per sample, and integer residual color transform for coding RGB signal.

– Mainly targeting the HD DVD, Broadcast, Editing, Digital still camera market

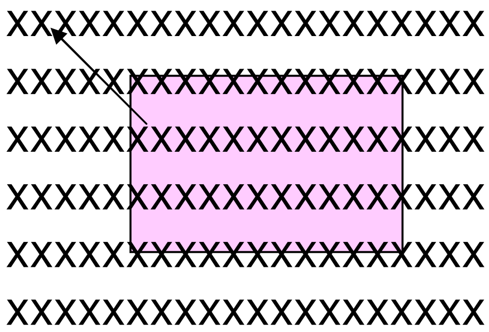
Features

- Variable block size motion compensation.



- Quarter sample accuracy.

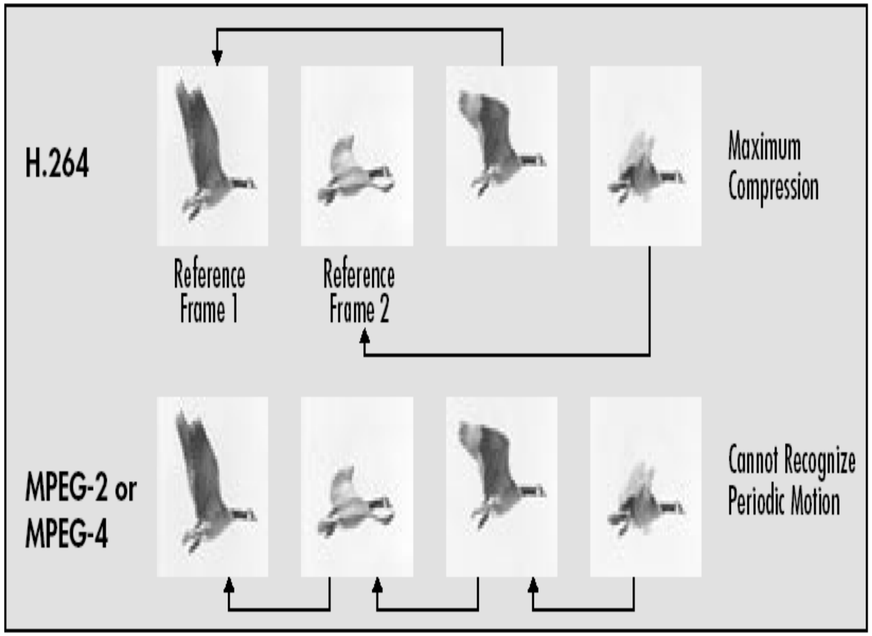
- Unrestricted motion vector.



Features

Support for multiple reference pictures. It gives significant compression when motion is periodic in nature.

H.264 Motion Estimation – Multiple Reference Frames



Features

- Weighted prediction.
  - Different weight can be given to the signal
    Ex: X1 and X2 are prediction signals (B - frame).
    w1 and w2 are the weights,
    Then Final predicted value = w1 * x1 + w2 * x2.

- Intra prediction.
  - Many new modes
  - Happens in spatial domain

- 4*4 Transform
  - 16 bit word length transform.
  - Bit exact transform.
  - Can be implemented by add and shift.

Features

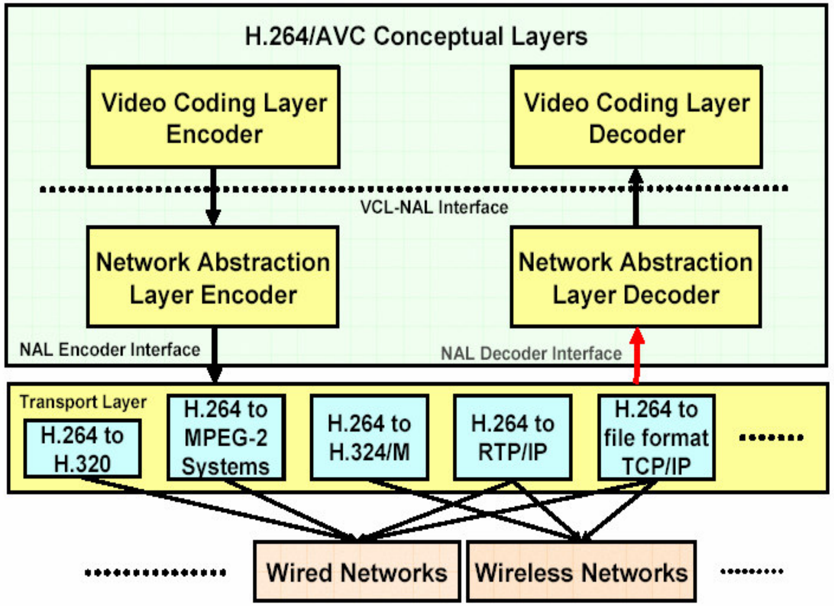
- Arithmetic coding.
  - Used to create a very powerful technique CABAC
- Parameter set structure
  - Loss of few key bits can have severe negative impact
  - Header information handled separately in a more flexible manner.
- Flexible slice size.
  - Increase coding efficiency by reducing the header data.

Features

- PAFF (Picture adaptive frame/field)
  - Combine the two fields together and to code them as one single coded frame (frame mode).
  - Not combine the two fields and to code them as separate coded fields (field mode).
- MBAFF (Macro block adaptive frame/field)
  - The decision of field/frame happens at macro block pair level.

Features

- NAL unit syntax structure
  - Each structure Is placed into logical data unit depending on the network.



Features

- Flexible macro block ordering
  - Picture can be partitioned into regions (slices)
  - Each region can be decoded independently.

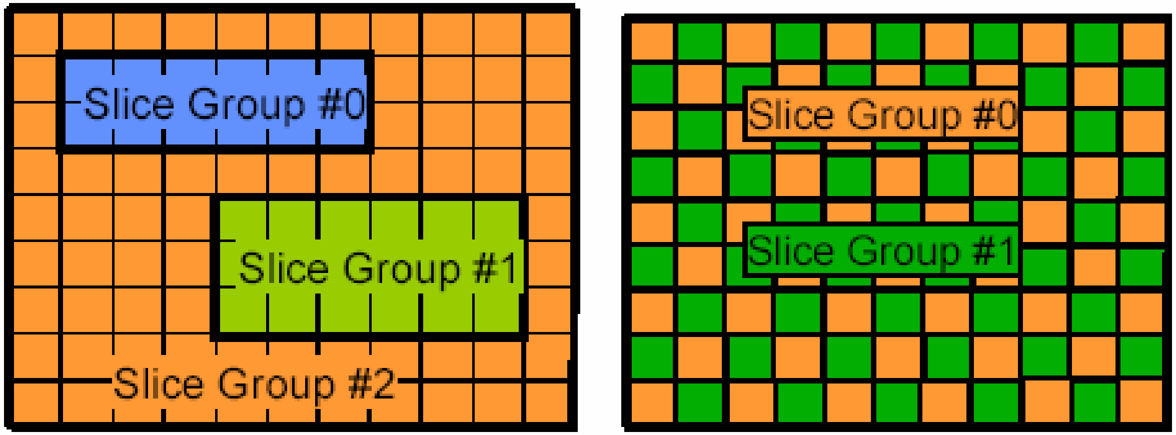


Fig. 7: Subdivision of a QCIF frame into slices when utilizing FMO.

Features

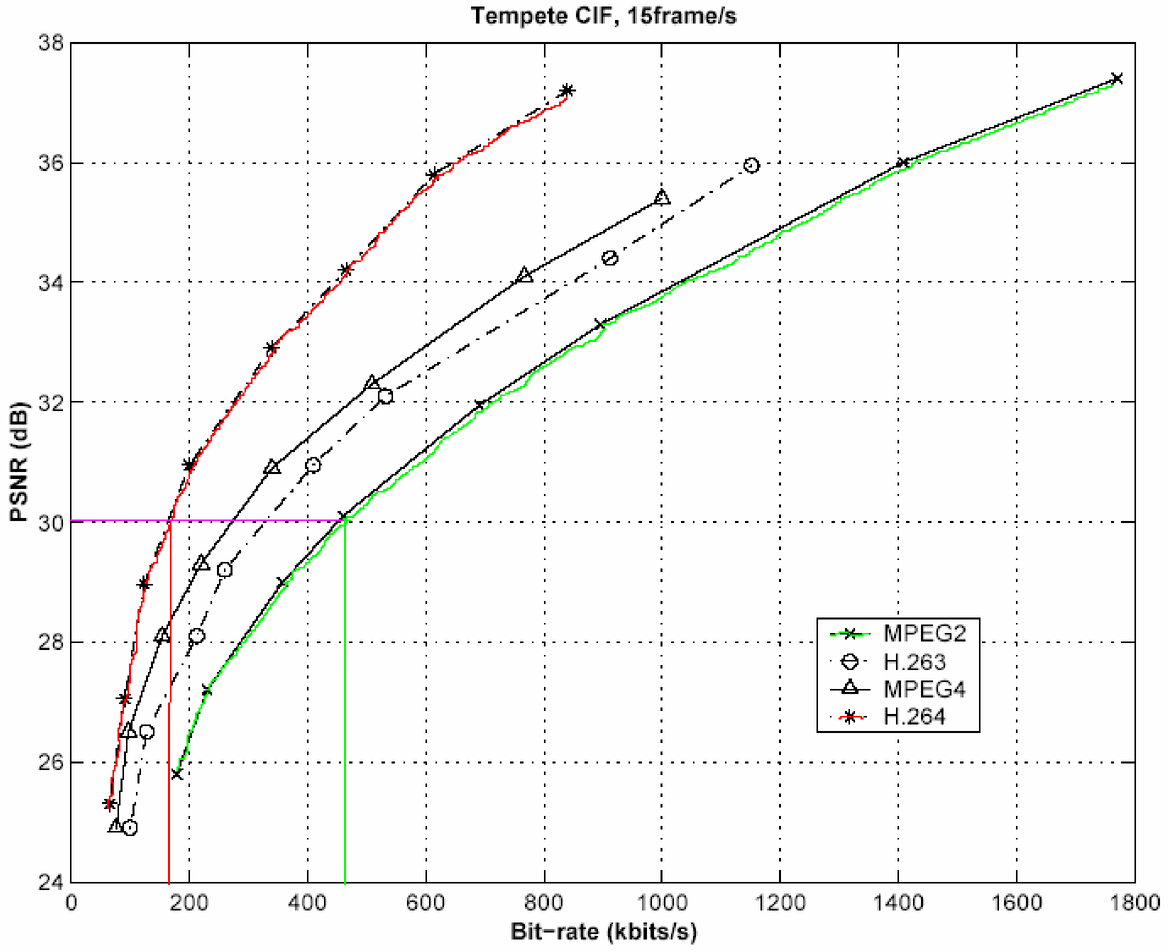
- Arbitrary slice ordering.
  - Since each slice can be decoded independently. It can be sent out of order
  - It reduces end to end delay in some networks.

- Redundant pictures
  - Encoder has the flexibility to send redundant pictures. These pictures can be used during loss of data.

Comparison

Feature	MPEG4	WMV9	H.264
Prediction Block size	16*16, 8*8	16*16, 16*8, 8*8 , 4*4	4*4,4*8,8*4,8*8, 8*16,16*8,16*16
Intra Prediction	Ac Prediction (Transform Domain)	Ac Prediction (Transform Domain)	Intra Prediction (Spatial Domain)
Entropy coding	VLC	VLC	CAVLC,CABAC
Reference frame	One picture	Two (Interlace)	Multiple pictures
Weighted Prediction	No	No	Yes
De-blocking Filter	No (Optional)	Yes	Yes
Transform	8*8 DCT	4*4,4*8,8*4,8*8	4*4,8*8(High Profile)

Comparison



Baseline Profile

Entropy Coding Tools of H.264

Two modes in H.264

- Entropy coding mode = 0 : Residual block coded using Context Adaptive Variable Length Coding (CAVLC) and other syntax elements are coded using Exp-Golomb codes.
- Entropy coding mode = 1 : Residual block and other syntax elements are coded using Context Adaptive Binary Arithmetic Coding (CABAC).
- Baseline Profile has support for mode 0 only.
- Main Profile and higher profiles support mode 0 and mode 1.

Context Adaptive Variable Length Coding (CAVLC)

Past Deficiencies

- The usage of fixed VLC tables does not allow an adaptation to the actual symbol statistics.
- Entropy coding such as MPEG-2, H.263, MPEG-4 (SP) is based on fixed tables of VLCs i.e Probability distribution is static
- Since there is a fixed assignment of VLC tables and syntax elements, existing inter-symbol redundancies cannot be exploited i.e. no conditional probabilities are used

Context Adaptive VLC of Residual Coefficients

- Run-length encoding is used.
- Number of non-zero coefficients and trailing ones (coeff_token).
- Sign of each trailing one.
- Levels of the remaining non-zero coefficients.
- Total number of zeros before the last coefficient.
- Each run of zeros.

Example of encoding a 4x4 residual block

4x4 block:

0	3	-1	0
0	-1	1	0
1	0	0	0
0	0	0	0

Reordered block:
0,3,0,1,-1,-1,0,1,0…

TotalCoeffs = 5 (indexed from highest frequency [4] to lowest frequency [0])
TotalZeros = 3
T1s = 3 (in fact there are 4 trailing ones but only 3 can be encoded as a "special case")

Encoding:

Element	Value	Code
coeff_token	TotalCoeffs=5, T1s=3	0000100
T1 sign (4)	+	0
T1 sign (3)	-	1
T1 sign (2)	-	1
Level (1)	+1 (use Level_VLC0)	1
Level (0)	+3 (use Level_VLC1)	0010
TotalZeros	3	111
run_before(4)	ZerosLeft=3; run_before=1	10
run_before(3)	ZerosLeft=2; run_before=0	1
run_before(2)	ZerosLeft=2; run_before=0	1
run_before(1)	ZerosLeft=2; run_before=1	01
run_before(0)	ZerosLeft=1; run_before=1	No code required; last coefficient.

The transmitted bitstream for this block is 000010001110010111101101 .

Example of decoding

The output array is "built up" from the decoded values as shown below. Values added to the output array at each stage are underlined.

Code	Element	Value	Output array
0000100	coeff_token	TotalCoeffs=5, T1s=3	Empty
0	T1 sign	+	1
1	T1 sign	-	-1, 1
1	T1 sign	-	-1, -1, 1
1	Level	+1	1, -1, -1, 1
0010	Level	+3	3, 1, -1, -1, 1
111	TotalZeros	3	3, 1, -1, -1, 1
10	run_before	1	3, 1, -1, -1, 0, 1
1	run_before	0	3, 1, -1, -1, 0, 1
1	run_before	0	3, 1, -1, -1, 0, 1
01	run_before	1	3, 0, 1, -1, -1, 0, 1

The decoder has inserted two zeros; however, TotalZeros is equal to 3 and so another 1 zero is inserted before the lowest coefficient, making the final output array:
0, 3, 0, 1, -1, -1, 0, 1

Example Courtesy: www.vcodex.com (H.264 white papers)

TEXAS INSTRUMENTS 75 YEARS OF INNOVATION

Adapting based on Context

- Encoding of coefficients in reverse zig-zag order
- Seven VLC tables
- Level_VLC0 biased towards lower magnitude, Level_VLC1 biased towards slightly higher magnitude and so on.
- Start with Level_VLC0 table
- If the magnitude of the coefficient is larger than a predefined threshold, switch to next table.

Current table	Threshold to increment table
Level_VLC0	0
Level_VLC1	3
Level_VLC2	6
Level_VLC3	12
Level_VLC4	24
Level_VLC5	48
Level_VLC6	-

TEXAS INSTRUMENTS 75 YEARS OF INNOVATION

Exp-Golomb entropy coding

- Variable Length codes with a regular construction
- Easy to parse
- Optimal for one / two-sided geometric pdfs
- No need for tables
- Syntax elements covered – mb_type, sub_mbtype, MVD, ref_idx etc.

Exp-Golomb code words

- Format :
  [M zeros] [1] [M bits of info]
- Code_num = 2^M + info – 1
- For signed numbers another level of mapping is used –

V	Code_num
0	0
1	1
-1	2
2	3
-2	4
3	5

Code_num	Codeword	2^M + info – 1
0	1	1 + 0 – 1
1	010	2 + 0 – 1
2	011	2 + 1 – 1
3	00100	4 + 0 – 1
4	00101	4 + 1 – 1
5	00110	4 + 2 – 1
6	00111	4 + 3 – 1
7	0001000	8 + 0 – 1

Context Adaptive Binary Arithmetic Coding (CABAC)

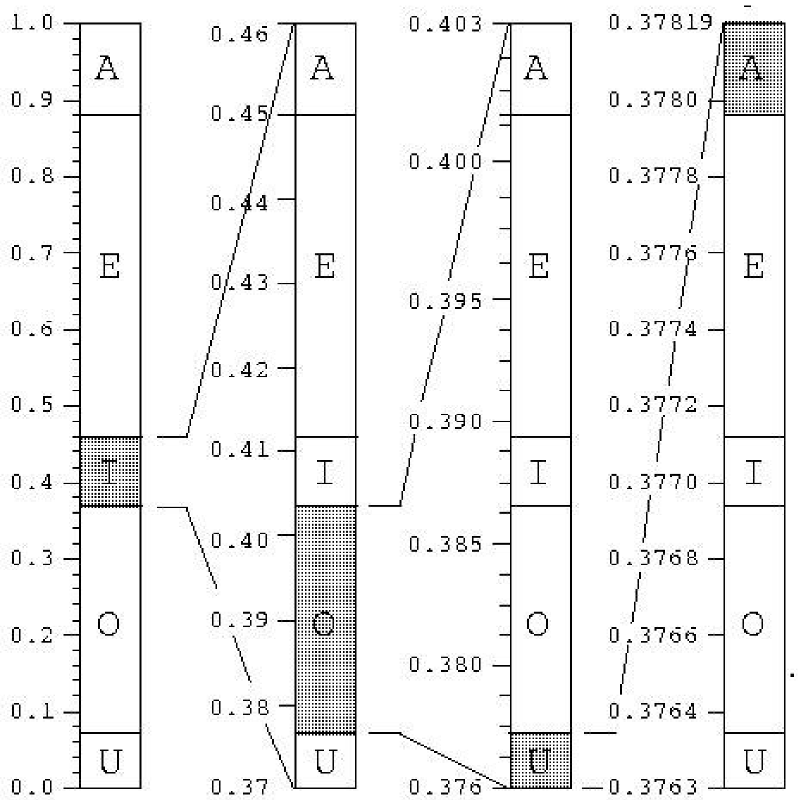
Arithmetic Coding Basics

- From an Information Theory point of view Minimum no of bits for a symbol = $-\log_2 p$, where p is probability of occurrence of the symbol.
- Huffman coding takes fractional number of bits, unless the probabilities are powers of .5.
- Further, if the probability is more than .5 compulsorily one bit is allocated.
- Solution would be to allocate bits for multiple symbols.
- Arithmetic coding achieves this by coding a group of symbols together.

Arithmetic Coding Basics

- Basic Idea: Working interval is divided based on probability of symbols and the sub-interval corresponding to the current symbol is used as interval for the next step.

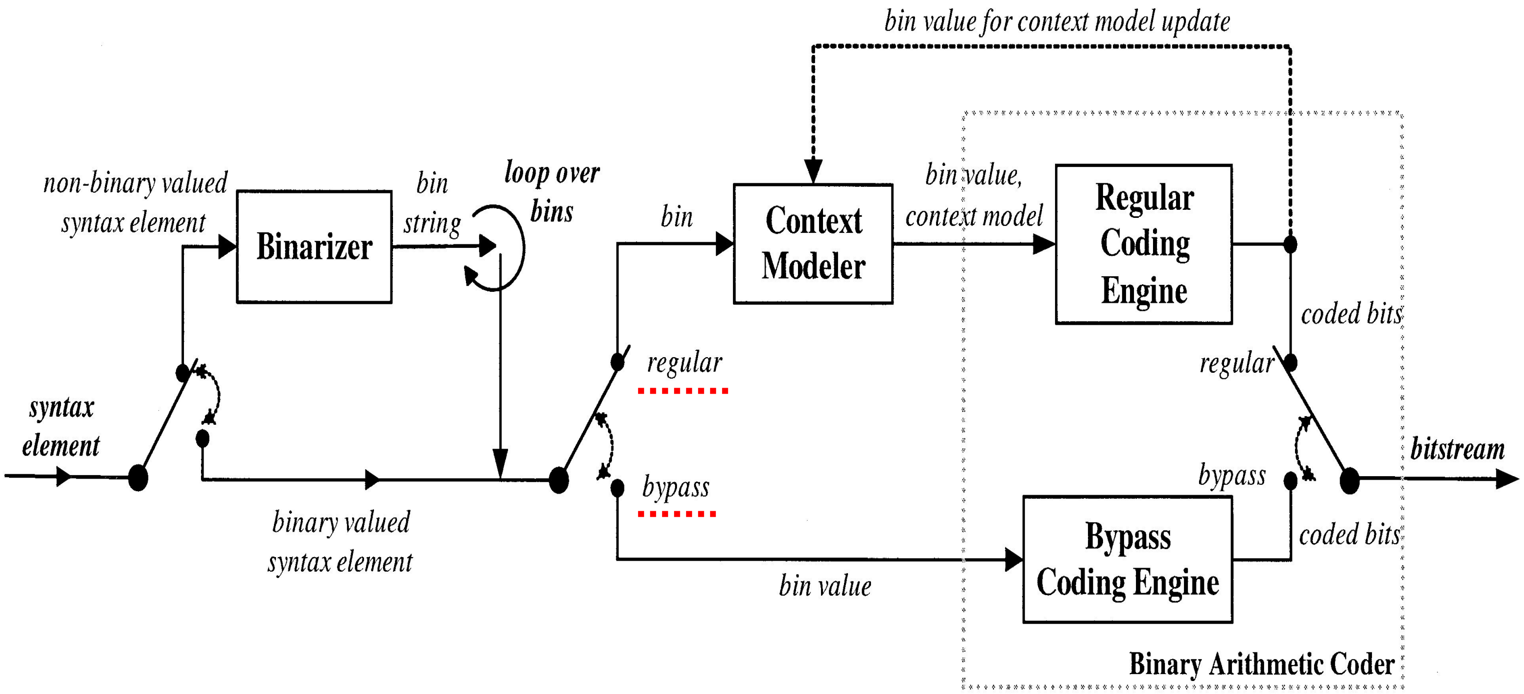
- Example: Source emitting five possible symbols – A, E, I, O, U. Sequence of symbols shown is IOUA.



Moving towards CABAC

- Usage of adaptive probability models
- Restriction to binary arithmetic coding, for Simple and fast adaptation mechanism
- Exploiting symbol correlations by using contexts

CABAC Framework



binarization ➔ context modeling ➔ arithmetic coding

TEXAS INSTRUMENTS 75 YEARS OF INNOVATION

Binarization schemes

- Unary Binarization: For each unsigned integer valued symbol x>=0, the code word consists of x “1” bits followed by a terminating “0” bit.

  Example - U: 6 => 1111110

- Truncated unary (TU) code: This is only defined for x with 0 <= x <= S, where for x < S the code is given by the unary code, whereas for x=S the terminating “0” bit is neglected.

  Example – TU with s=9.

  6 => 1111110

  9 => 111111111

- Fixed-Length Binarization

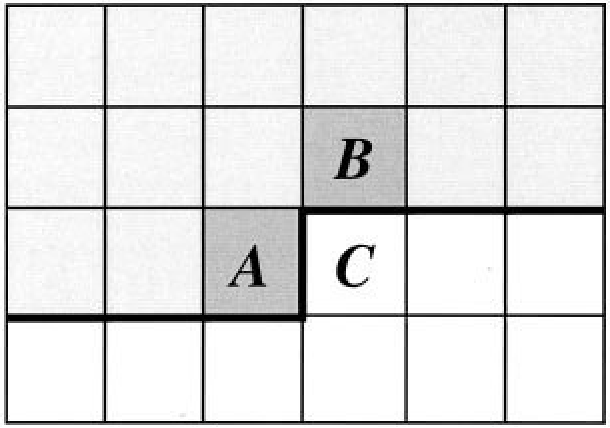
- kth order Exp-Golomb Binarization

- Concatenated schemes -

TU Code	**UEGk Code**

Context Modeling

- Encode the bin based on context info.
- Four types of context models :
  - Based on neighboring blocks info
  - Based on previous bins
  - Based on scanning position (Significance map)
  - Based on previously encoded levels (Transform coefficients)



context template consisting of two neighboring syntax element A and B to the left and on top of the current syntax element C.

Context Modeling Example

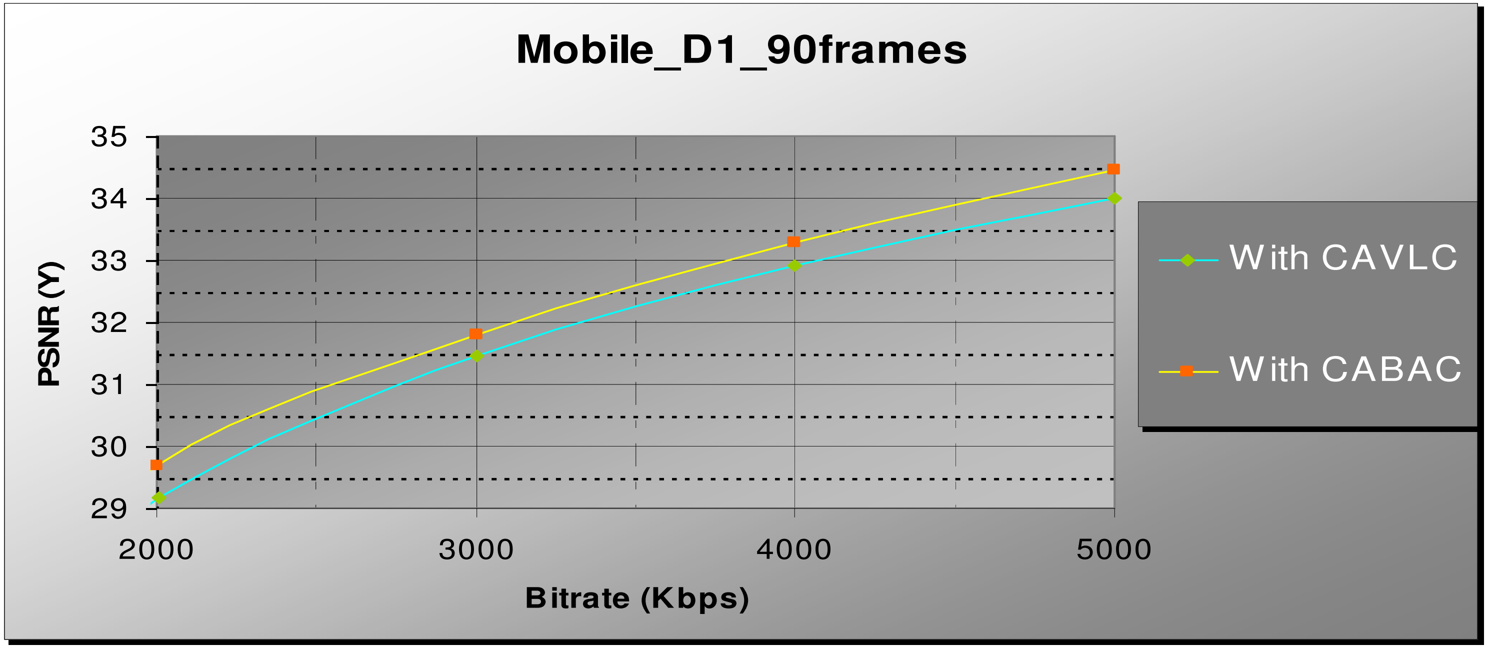
- Context selection for mvd –

	Bin0	Bin1	Bin2	Bin3	Bin >= 4 & <= 8	Other bins
mvdx	40/ 41/ 42	43	44	45	46	-
mvdy	47/ 48/ 49	50	51	52	53	-

$$\text{Sum\_abs\_mvdx} = |\text{mvdx\_A}| + |\text{mvdx\_B}|$$

$$\text{For mvdx, ctx\_idx} = \begin{cases} 40 & \text{if Sum\_abs\_mvdx} < 3 \\ 41 & \text{if Sum\_abs\_mvdx} <= 32 \\ 42 & \text{if Sum\_abs\_mvdx} > 32 \end{cases}$$

CAVLC Vs CABAC



Typically CABAC provides 10-15 % reduction in bit rate compared to CAVLC, for the same PSNR.

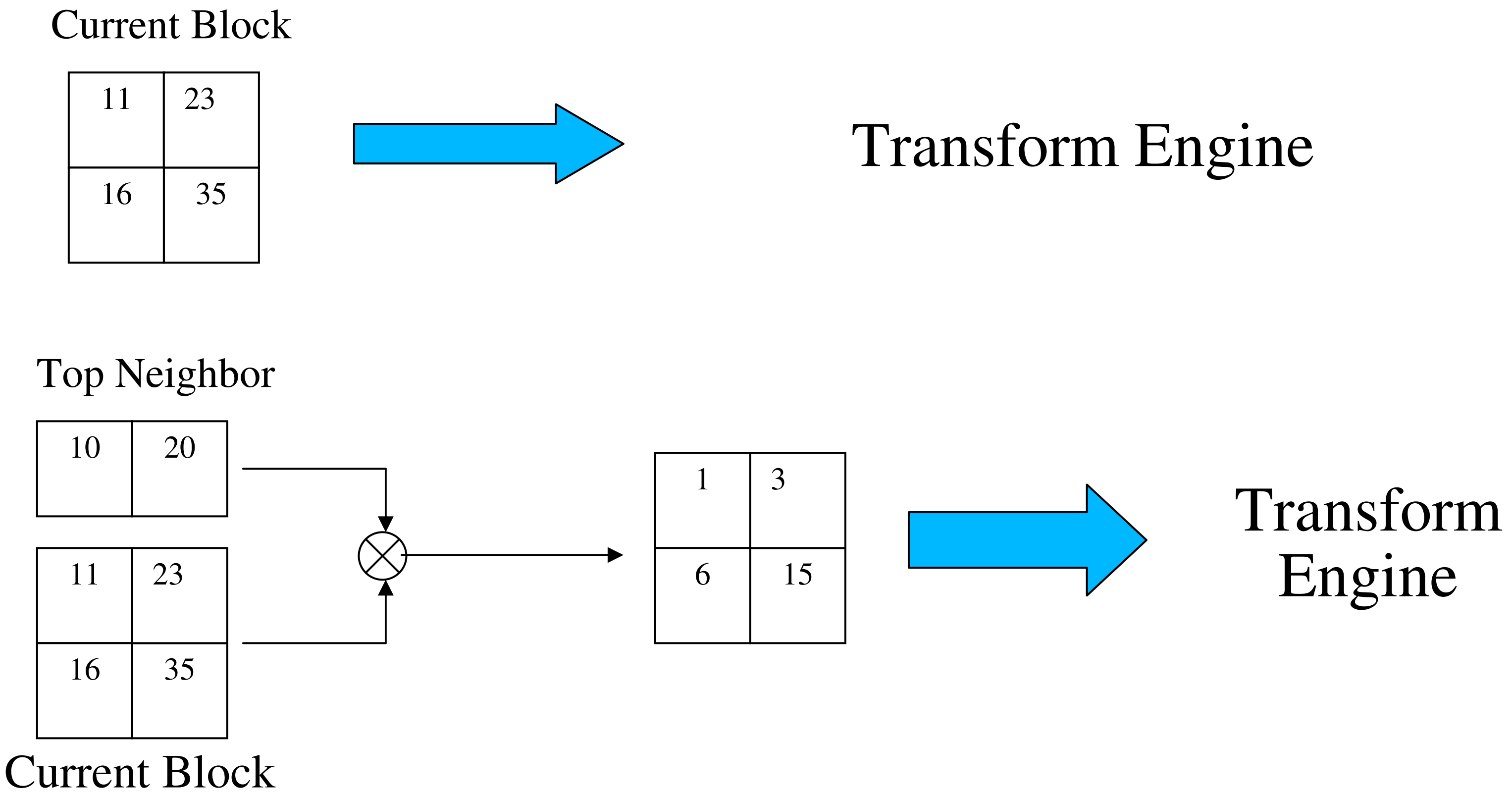
INTRA PREDICTION IN H.264

Agenda

- What is Intra Prediction.
- New Approach to Prediction…
- Advantages of the spatial domain prediction…
- The Big Picture…
- Intra-Prediction Modes
- Compression Benefits achieved.
- Implementation Challenges for Intra-Prediction

What is Intra Prediction…

- Intra Prediction is a process of using the pixel data predicted from the neighboring blocks for the purpose of sending information regarding the current macro-block instead of the actual pixel data.



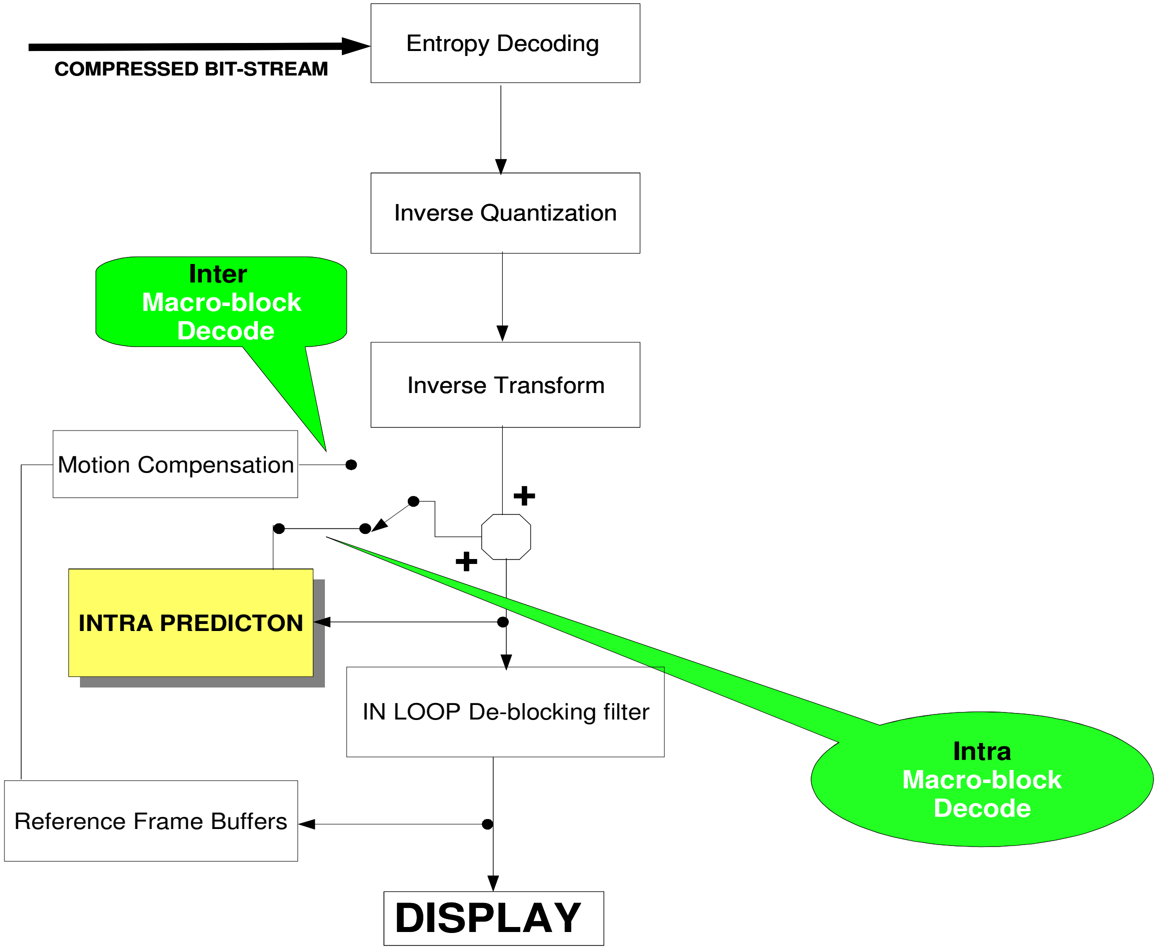
New approach to Prediction...

- The H.264/AVC uses a new approach to the prediction of intra blocks by doing the prediction in the spatial domain rather than in frequency domain like other codecs.
- The H.264 /AVC uses the reconstructed but unfiltered macroblock data from the neighboring macroblocks to predict the current macroblock coefficients.

Advantages of spatial domain predictions…

- Intuitively, the prediction of pixels from the neighbouring pixels (top/left) of macro-blocks in the raster scan would be much efficient as compared to the prediction of the transform domain values.
- Predicting from samples in the pixel domain helps is better compression for intra blocks in a inter frame.
- Allows to better compression and hence a flexible bit-rate control by providing the flexibility to eliminate redundancies across multiple directions.

Intra-Prediction in the big picture…

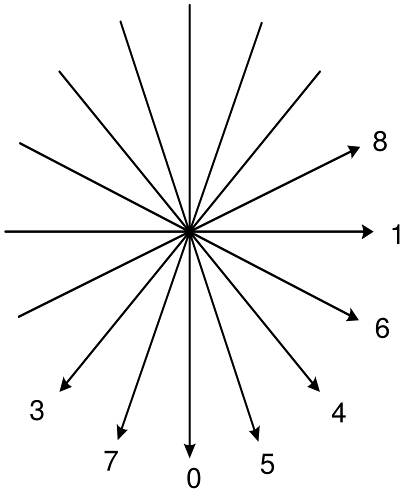


Intra Prediction Modes

- H.264/AVC supports intra-prediction for blocks of 4 x 4 to help achieve better compression for high motion areas.
  - Supports 9 prediction modes.
  - Supported only for luminance blocks
- H.264/AVC also has a 16 x 16 mode, which is aimed to provide better compression for flat regions of a picture at a lower computational costs. This mode is also helpful to avoid the irritating gradients that show up in flat regions of the picture quantized with high quantization parameters.
  - Supports 4 direction modes.
  - Supported for 16x16 luminance blocks and 8x8 chrominance blocks

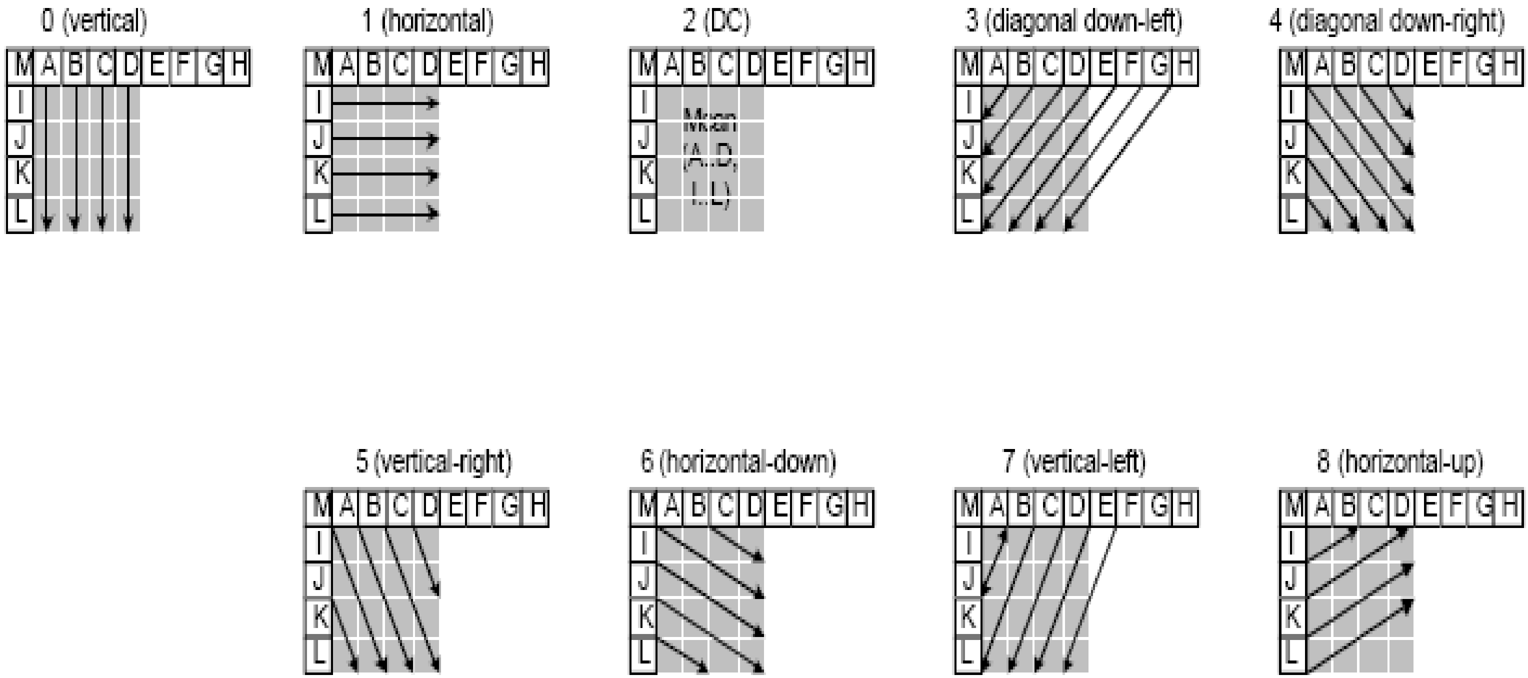
Luma 4x4 Intra-Prediction Modes explained...

- The H264 /MPEG4 AVC provides for eliminating redundancies in almost all directions using the 9 modes as shown below.

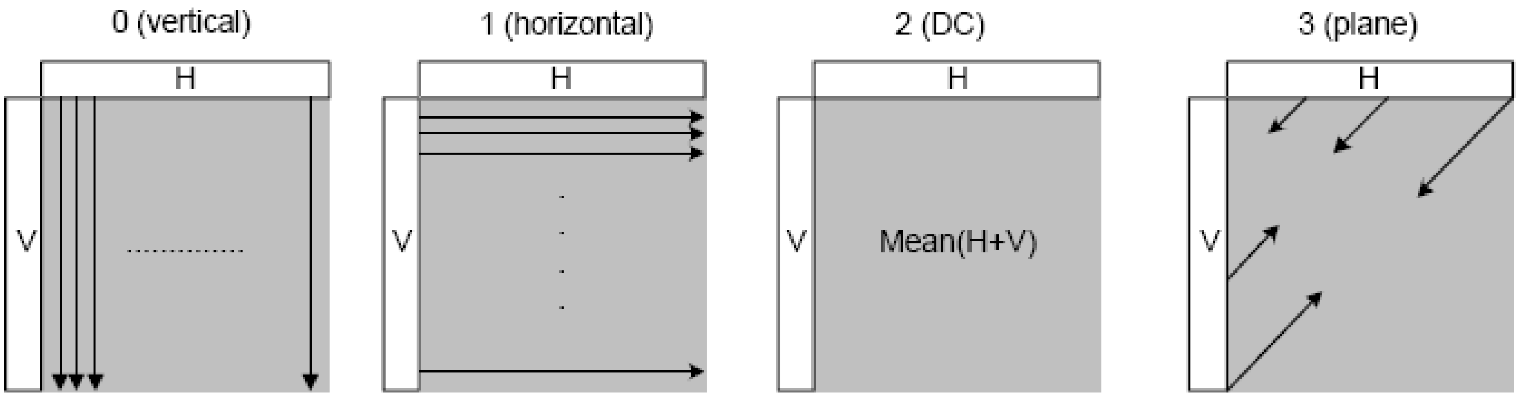


Intra4x4PredMode[luma4x4BlkIdx]	Name of Intra4x4PredMode[luma4x4BlkIdx]
0	Intra_4x4_Vertical (prediction mode)
1	Intra_4x4_Horizontal (prediction mode)
2	Intra_4x4_DC (prediction mode)
3	Intra_4x4_Diagonal_Down_Left (prediction mode)
4	Intra_4x4_Diagonal_Down_Right (prediction mode)
5	Intra_4x4_Vertical_Right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

Luma 4x4 Intra-Prediction Modes explained...



LUMA 16x16 / CHROMA Intra-Prediction Modes explained...



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)

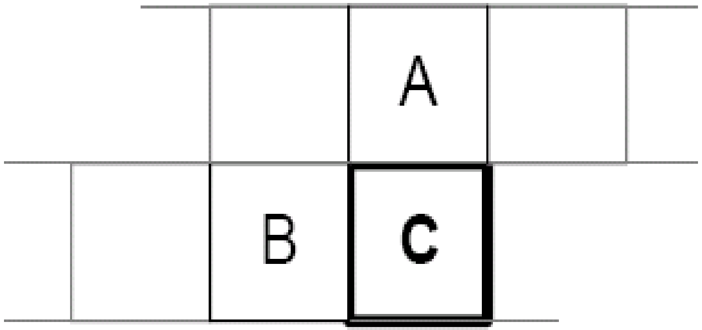
intra_chroma_pred_mode	Name of intra_chroma_pred_mode
0	Intra_Chroma_DC (prediction mode)
1	Intra_Chroma_Horizontal (prediction mode)
2	Intra_Chroma_Vertical (prediction mode)
3	Intra_Chroma_Plane (prediction mode)

Intra-Prediction Process…

1. **Determining the prediction mode (Only for a 4x4 block size mode).**
2. Determination of samples to predict the block data.
3. Predict the block data.

Determining the prediction mode

(Only for a 4x4 block size mode)



-Flag in the bit-stream indicates, whether prediction mode is present in the bit-stream or it has to be Implicitly calculated.

-In case of Implicit mode, the prediction mode is the minimum of prediction modes of neighbors ‘A’ and ‘B’.

Intra-Prediction Process…

1. Determining the prediction mode (Only for a 4x4 block size mode).
2. **Determination of samples to predict the block data.**
3. Predict the block data.

Determination of samples to predict the block data.

- To Predict a 4x4 block (a-p), a set of 13 samples (A-M) from the neighboring pixels have to be chosen.
- For a 8x8 chrominance block a set if 17 neighboring pixels are chosen as sample values.
- Similarly for predicting a 16x16 luminance block, a set of 33 neighboring pixels are selected as the samples

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

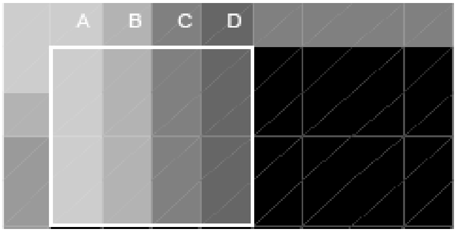
Intra-Prediction Process…

1. Determining the prediction mode (Only for a 4x4 block size mode).
2. Determination of samples to predict the block data.
3. **Predict the block data.**

Intra-Prediction Process…

- **Vertical prediction mode**

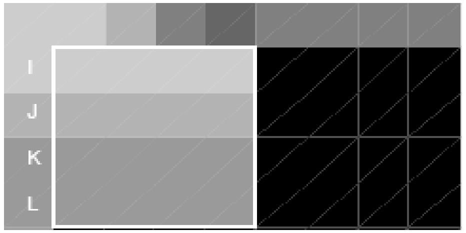
M	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
I	A	B	C	D				
J	A	B	C	D				
K	A	B	C	D				
L	A	B	C	D				



Intra-Prediction Process…

- **Horizontal prediction mode**

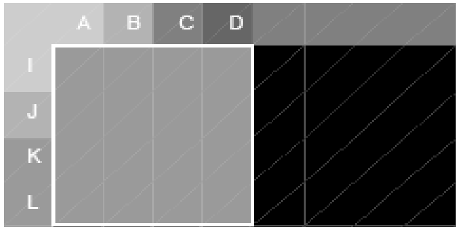
M	A	B	C	D	E	F	G	H
I	I	I	I	I				
J	J	J	J	J				
K	K	K	K	K				
L	L	L	L	L				



Intra-Prediction Process…

- **DC prediction mode**

M	A	B	C	D	E	F	G	H
I	X	X	X	X				
J	X	X	X	X				
K	X	X	X	X				
L	X	X	X	X				



X = Mean

Intra-Prediction Process…

- **Remaining 4x4 block size modes**
  - **Intra_4x4_Diagonal_Down_Left prediction mode (Mode 3)**
  - **Intra_4x4_Diagonal_Down_Right prediction mode (Mode 4)**
  - **Intra_4x4_Vertical_Right prediction mode (Mode 5)**
  - **Intra_4x4_Horizontal_Down prediction mode (Mode 6)**
  - **Intra_4x4_Vertical_Left prediction mode (Mode 7)**
  - **Intra_4x4_Horizontal_Up prediction mode (Mode 7)**
- **Remaining 16x16 block size modes**
  - **Intra_16x16_Plane prediction mode**

Implementation challenges with the intra-Prediction…

- The dependence of the blocks prediction samples on it's neighbors, which itself may a part of current MB prevent parallel processing of block data.
- Each of the 16 blocks in a given MB can choose any one of the nine prediction modes, With each mode entire processing changes. Each mode has a totally different mathematical weighting function used for deriving the predicted data from the samples.

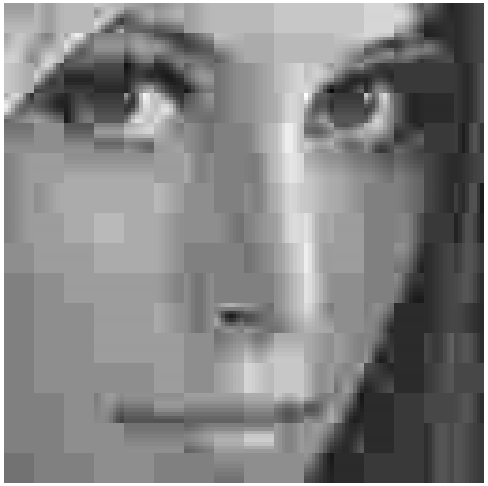
H.264/MPEG4 AVC
In Loop De-blocking filter

Agenda

- Introduction
- H.264 /MPEG4 AVC adaptive De-blocking filter process…
- Implementation Challenges

Introduction

- Coarse quantization of the block-based image transform produce disturbing blocking artifacts at the block boundaries of the image.
- Motion compensation of the macro-block by interpolation of data from previous reference frames might never give a perfect match, discontinuities appear at the edges of the copied blocks.
- When the later P/B frames reference these images having blocky edges, the blocking artifacts further propagates to the interiors of the current blocks block worsening the situation further.

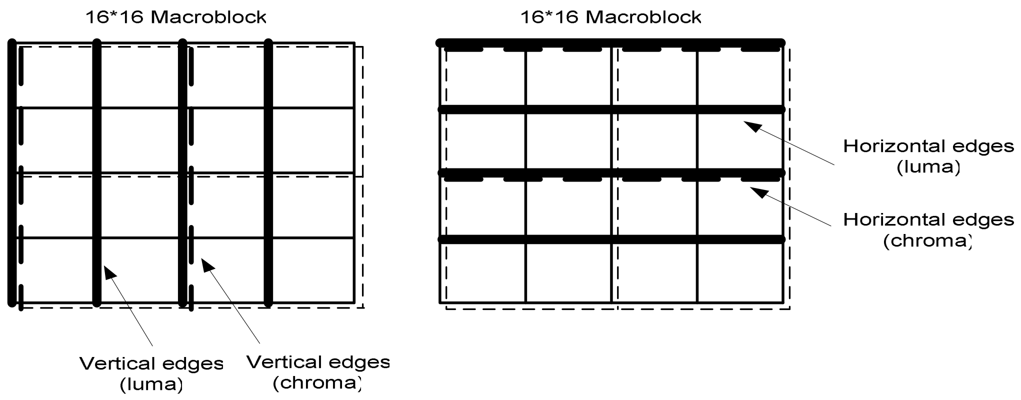


Introduction

- The best way to deal with these artifacts is to filter the blocky edges to have a smoothed edge. This filtering process in known as the de-block filtering.
- Till recently, the coding standards, defined the de-blocking filter, but not mandating the use of the same, as the implementation is cycle consuming and is a function of the quality needed at the user end.
- But it was soon figured out that if the de-block filter is not compulsorily implemented the frames suffered from blockiness caused in the past frames used as reference.
- This coupled with the increasing number crunching powers of the modern day DSP's, made it a easier choice for the standards body to make this de-block filter mandatory tool or a block in the decode loop – IN LOOP DEBLOCK FILTER.
- This filter not only smoothened the irritating blocky edges but also helped increase the rate-distortion performance.

H.264 /MPEG4 AVC adaptive De-blocking filter process…

- Last process in the process of frame decode, which ensures all the top/left neighbors have been fully reconstructed and available as inputs for de-blocking the current MB.
- Applied to all 4x4 blocks except at the boundaries of the picture.
- Filtering for block edges of any slice can be selectively disabled by means of flag.
- Vertical edges filtered first (left to right) followed by horizontal edges (top to bottom)



H.264 /MPEG4 AVC adaptive De-blocking filter process…

- For de-blocking an edge, 8 pixel samples in all are required in which 4 are from one side of the edge and 4 from the other side.

p3	p2	p1	p0	q0	q1	q2	q3

❑Of these 8 pixel samples the de-block filter updates 6 pixels for a luminance block and 4 pixels for a chrominance block.

p3	**p2’**	**p1’**	**p0’**	**q0’**	**q1’**	**q2’**	q3

Luminance pixels after filtering

p3	p2	**p1’**	**p0’**	**q0’**	**q1’**	q2	q3

Chrominance pixels after filtering

H.264 /MPEG4 AVC adaptive De-blocking filter process…

- **Determine Boundary strengths**
- Determine edge

H.264 /MPEG4 AVC adaptive De-blocking filter process…

- The Boundary strengths are a method of implementing adaptive filtering for a given edge based on certain conditions.
- H264/MPEG4 AVC has a set of four boundary strengths.
- The Boundary strengths for a chrominance block is determined from the boundary strength of the corresponding luminance macro block.
  - For a chrominance sample in a frame macro-block located at pix (x, y), the boundary strength is same as the luminance pixel located at pix(2*x, 2*y)
  - For a chrominance sample in a field macro-block located at pix (x, y), the boundary strength is same as the luminance pixel located at pix (2*x, 2*y) in the same field.

H.264 /MPEG4 AVC adaptive De-blocking filter process…

Boundary Strength	Conditions
4 (Strongest)	Samples p_0 and q_0 are from intra MB and across an MB edge
3	Samples p_0 and q_0 are from intra MB within a MB.
2	Samples p_0 and q_0 are from inter MB with at-least one non-zero residual in each MB.
1	Samples p_0 and q_0 are from frame type MB and are both in inter MB's and p_0 and q_0 have: ▪ Different reference frames and different numbers of MV's ▪Only one MV each having different MV's. ▪Two MV's pointing to two different references and having different MV values.
0 (None)	All other cases

H.264 /MPEG4 AVC adaptive De-blocking filter process…

- Determine Boundary strengths
- **Determine valid edge**

H.264 /MPEG4 AVC adaptive De-blocking filter process…

- In a given MB an edge is selected for de-blocking if :
  - the slice to which the current MB belongs is enabled for de-blocking
  - the corresponding neighboring block is available
- For the horizontal edge of a field mode block in a “Mbaff” frame, both the lines 0 and 1 of the block are selected for de-blocking if the top-right macro-block pair is a field MB

H.264 /MPEG4 AVC adaptive De-blocking filter process…

Filtering Process for Boundary strength Bs == 4

- Determine the Δ_1 defined as:
  $\Delta_1 = (((p_0 - q_0) << 2) + (p_1 - q_1) + 4) >> 3)$
- Clip Δ between $-t_c$ and t_c

p3	p2	**p1'**	**p0'**	**q0'**	**q1'**	q2	q3

p0' = Clip(0, 255, p0 + Δ_1)

q0' = Clip(0, 255, q0 – Δ_1)

if((p_2-p_0) < β)

P1' = p1 + Clip3(–tC0, tC0, (p2 + ((p0 + q0 + 1) >> 1) – (p1 << 1)) >> 1)

if((q_2-q_0) < β)

q1' = q1 + Clip3(–tC0, tC0, (q2 + ((p0 + q0 + 1) >> 1) – (q1 << 1)) >> 1)

H.264 /MPEG4 AVC adaptive De-blocking filter process…

Filtering Process for Boundary strength Bs < 4

- If the following threshold condition holds true for luminance blocks, filtering is done.

$\Delta_1 = ((p_2-p_0) < \beta\ )\ \&\&\ abs(p_0-q_0) < ((\alpha >> 2) + 2)$

$\Delta_2 = ((p_2-p_0) < \beta\ )\ \&\&\ abs(p_0-q_0) < ((\alpha >> 2) + 2)$

p3	p2'	p1'	p0'	q0'	q1'	q2'	q3

if (Δ_1)

$p'_0 = (p_2 + 2*p_1 + 2*p_0 + 2*q_0 + q_1 + 4) >> 3$
$p'_1 = (p_2 + p_1 + p_0 + q_0 + 2) >> 2$
$p'_2 = (2*p_3 + 3*p_2 + p_1 + p_0 + q_0 + 4) >> 3$

if(Δ_2)

$q'_0 = (p_1 + 2*p_0 + 2*q_0 + 2*q_1 + q_2 + 4) >> 3$
$q'_1 = (p_0 + q_0 + q_1 + q_2 + 2) >> 2$
$q'_2 = (2*q_3 + 3*q_2 + q_1 + q_0 + p_0 + 4) >> 3$

Chrominance blocks

$p'_0 = (2*p_1 + p_0 + q_1 + 2) >> 2$
$p'_1 = p_1$
$p'_2 = p_2$

$q'_0 = (2*q_1 + q_0 + p_1 + 2) >> 2$
$q'_1 = q_1$
$q'_2 = q_2$

Implementation Challenges

- To much dependency in the processing of the second sub-block on the pixels of the first sub-block due to pixels getting modified after filtering.
- Macro-block adaptive frame-field mode –Mbaff – introduces added complexity in terms of doing two lines at the start of a block and determining the boundary strengths.
- Choice of adapting the filter strengths from sub-block to sub-block causes increase in control code decreasing the efficiency of the DSP pipeline.
- Too many calculations for determining the boundary strengths, specially for P frames.

References

- Overview of the H.264/AVC Video Coding Standard
  ThomasWiegand, Gary J. Sullivan, *Senior Member, IEEE*, Gisle Bjøntegaard, and Ajay Luthra, *Senior Member, IEEE*
- Material from IEEE Transactions on Circuits and Systems for Video Technology, July 2003
- http://www.h263l.com/h264/www.utdallas.edu.bhong.h264.pdf
- Multimedia Communication laboratory
- www.vcodex.com – H.264 White Papers
- D. Marpe, H. Schwarz, and T.Wiegand, "Context-adaptive binary arithmetic coding in the H.264/AVC video compression standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 620–636, July 2003.
- "Introduction to Data Compression" – Khalid Sayood.
- Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)
- Gisle Bjontegaard, "Coding improvement by using 4x4 blocks for motion vectors and transform", Nov. 1997
- Gisle Bjontegaard, Adding Intra mode suitable for coding of flat regions.
- Peter List, Anthony Joch, Jani Lainema, Gisle Bjontegaard, and Marta Karczewicz, Adaptive Deblocking Filter - IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 13, NO. 7, JULY 2003

References

- D. Marpe, H. Schwarz, and T.Wiegand, "Context-adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 620–636, July 2003.
- "Introduction to Data Compression" – Khalid Sayood.
- Real-time MPEG-4 AVC/H.264 CABAC entropy coder Ha, V.H.S.; Woo-Sung Shim; Jung-Woo Kim; Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on 8-12 Jan. 2005 Page(s):255 - 256
  Digital Object Identifier 10.1109/ICCE.2005.1429814
- Comparison of context-based adaptive binary arithmetic coders in video compression Mrak, M.; Marpe, D.; Grgic, S.; Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on Volume 1, 2-5 July 2003 Page(s):277 - 286 vol.1 Digital Object Identifier 10.1109/VIPMC.2003.1220475
- A hardware accelerator for context-based adaptive binary arithmetic decoding in H.264/AVC, Jian-Wen Chen; Cheng-Ru Chang; Youn-Long Lin; Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on 23-26 May 2005 Page(s):4525 - 4528 Vol. 5, Digital Object Identifier 10.1109/ISCAS.2005.1465638

Thank You

Questions?

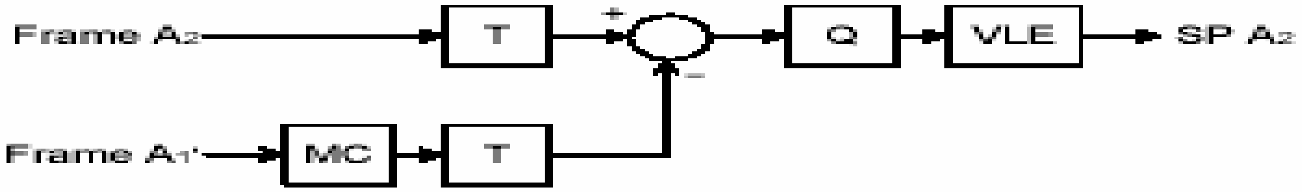


Figure 2-3 Encoding SP-slice A_2 (simplified)

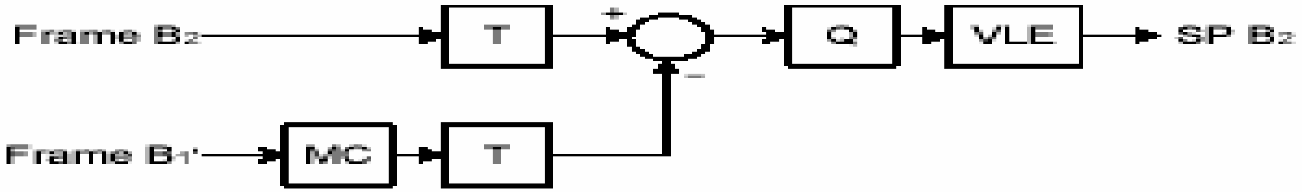


Figure 2-4 Encoding SP-slice B_2 (simplified)

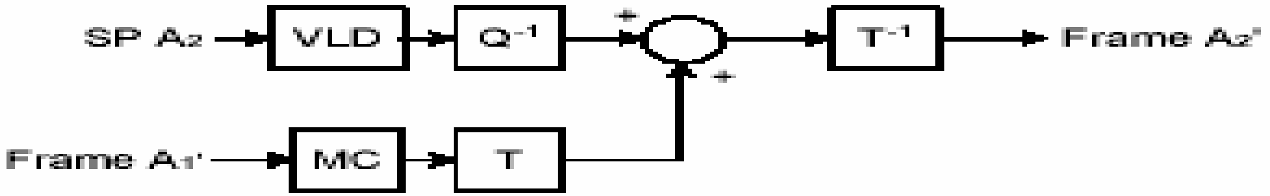


Figure 2-5 Decoding SP-slice A_2 (simplified)

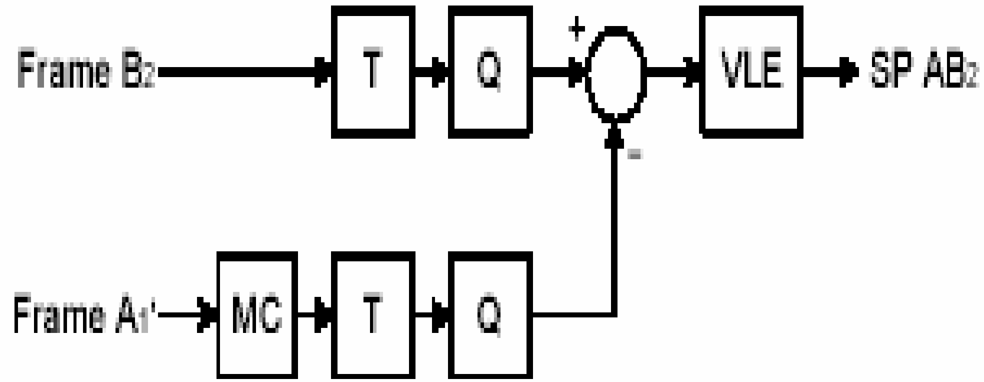


Figure 2-6 Encoding SP-slice AB_2 (simplified)

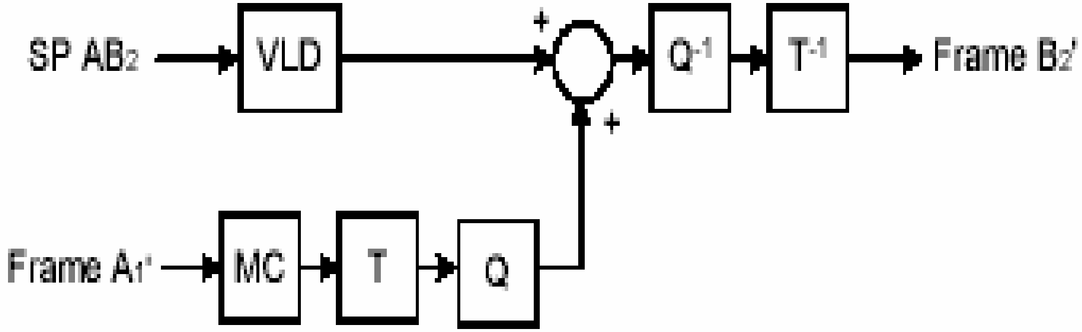


Figure 2-7 Decoding SP-slice AB_2