Audio **V**ideo Coding **S**tandard (AVS) – Video Techniques

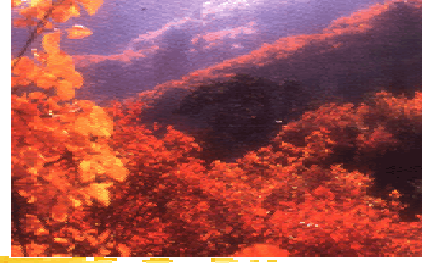
Prof. YU, Lu (虞露)

Zhejiang University
Institute of Information and Communication Engineering
yul@zju.edu.cn

Oct. 20, 2004

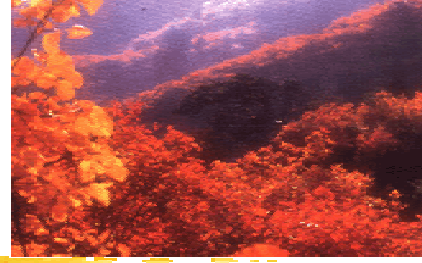
Outline

- Standard structure of AVS-video
  - AVS-video 1.0
  - AVS-M
- Technical characteristics of AVS-video 1.0
- Outline of AVS-M



Short History of AVS

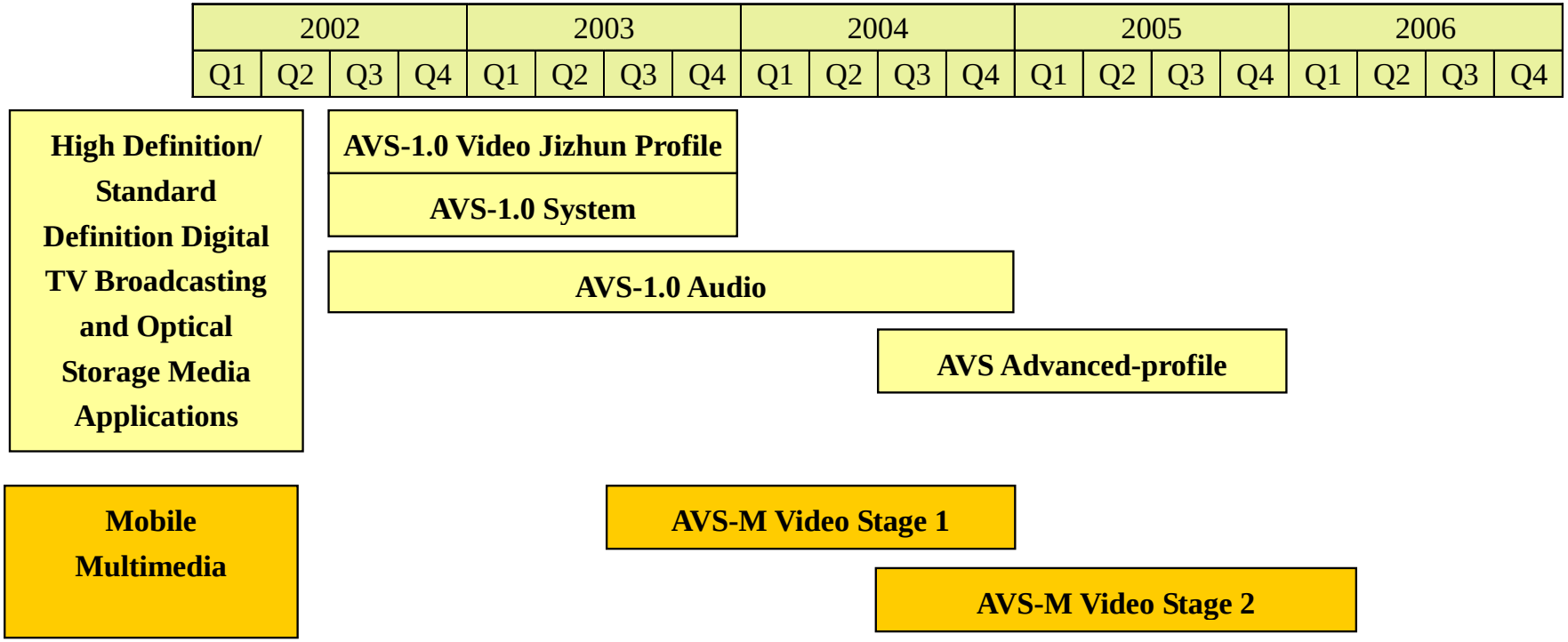
- Mar18-21. 2002
  - 178th Xiangshan Science Conference, Beijing, “Broad-band Network and Security Stream Media Technology”
- June 11, 2002
  - Science and Technology Department of MII released a bulletin about setting up “Audio Video Coding Standard Workgroup” on China Electronics News
- June 21, 2002
  - “Audio Video Coding Standard Working Group” was set up in Beijing.
- Aug 23-24, 2002
  - first meeting of AVS, AVS united with MPEG-China. Website of AVS opened to the members formally.
- Dec 9, 2002
  - Department of Science and Technology of Ministry of Information Industry issued the notice of Setting up “Audio Video Coding Standard Working Group” and assigned the task of the group
- Dec 19, 2003
  - On the 7th AVS meeting, AVS-video (1.0) and AVS-system (1.0) was finalized.



Short History of AVS

- Mar 29, 2004
  - Industry forum of AVS video coding technology towards 3G, ShenZhen, sponsored together with universities and companies of Hong Kong
- Mar. 30-31, 2004
  - Start up the video coding standardization for new generation of mobile communication

Standard Structure of AVS-Video



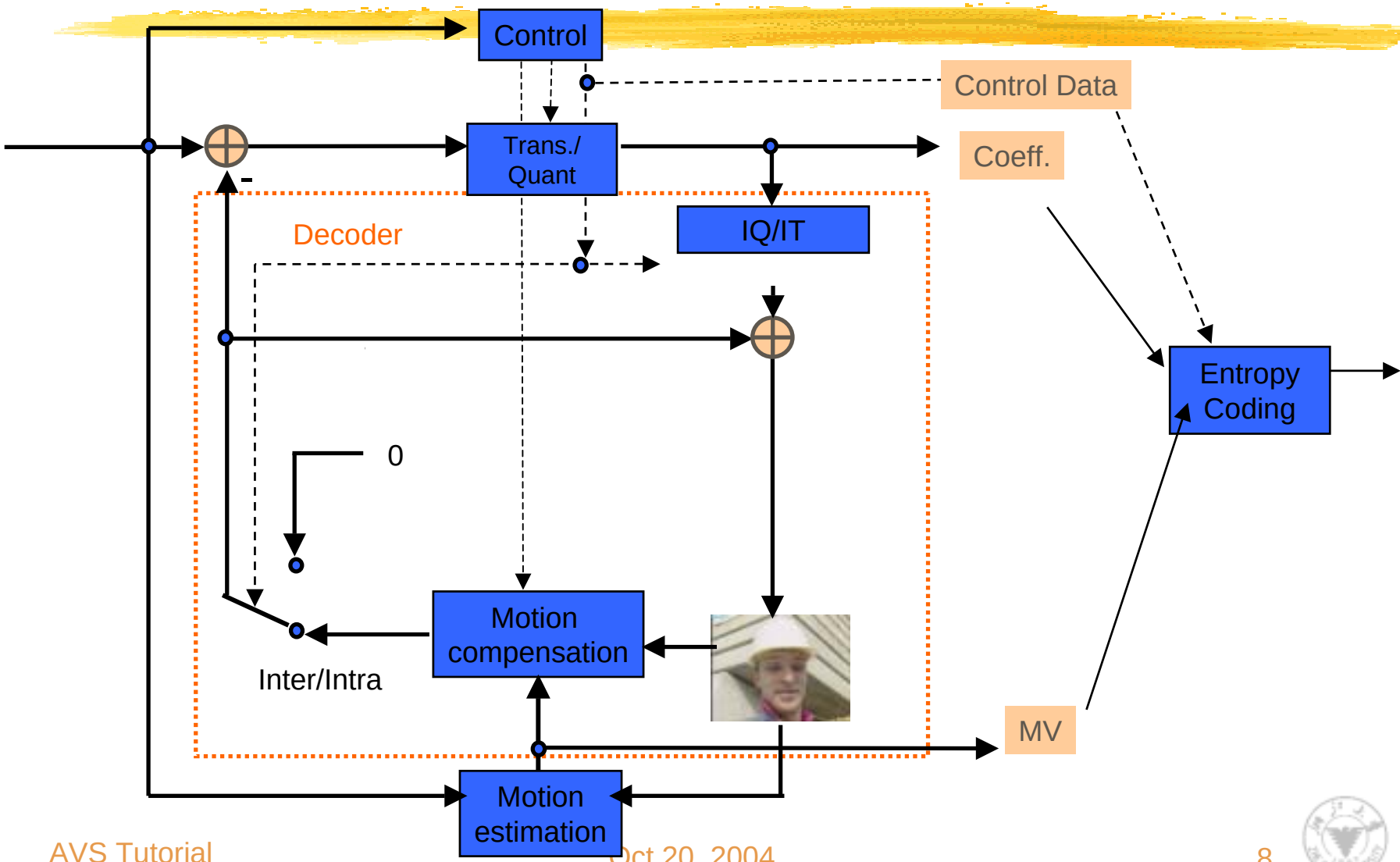
AVS Video Working Style

- Requirement
- Call for proposal & common test condition
- Technical proposal with technical explanation and experiment result (under common test condition)
- Review, cross check, core experiment
- Make decision
- Edit standard document and release reference software
- Corrigenda

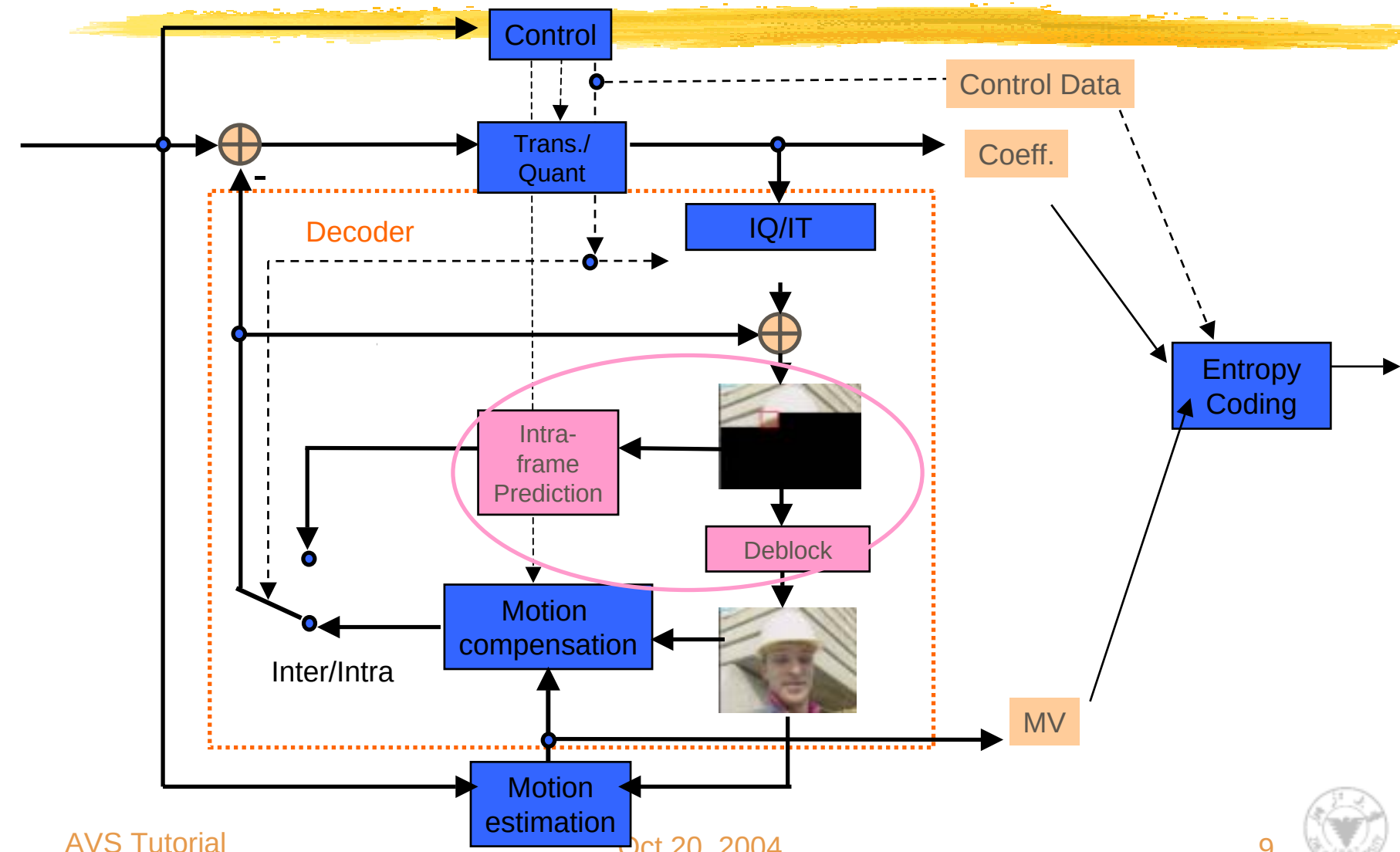
- 430 technical proposals in all 10 AVS meetings (up to Sep. 2004)

Technical Framework of AVS-video

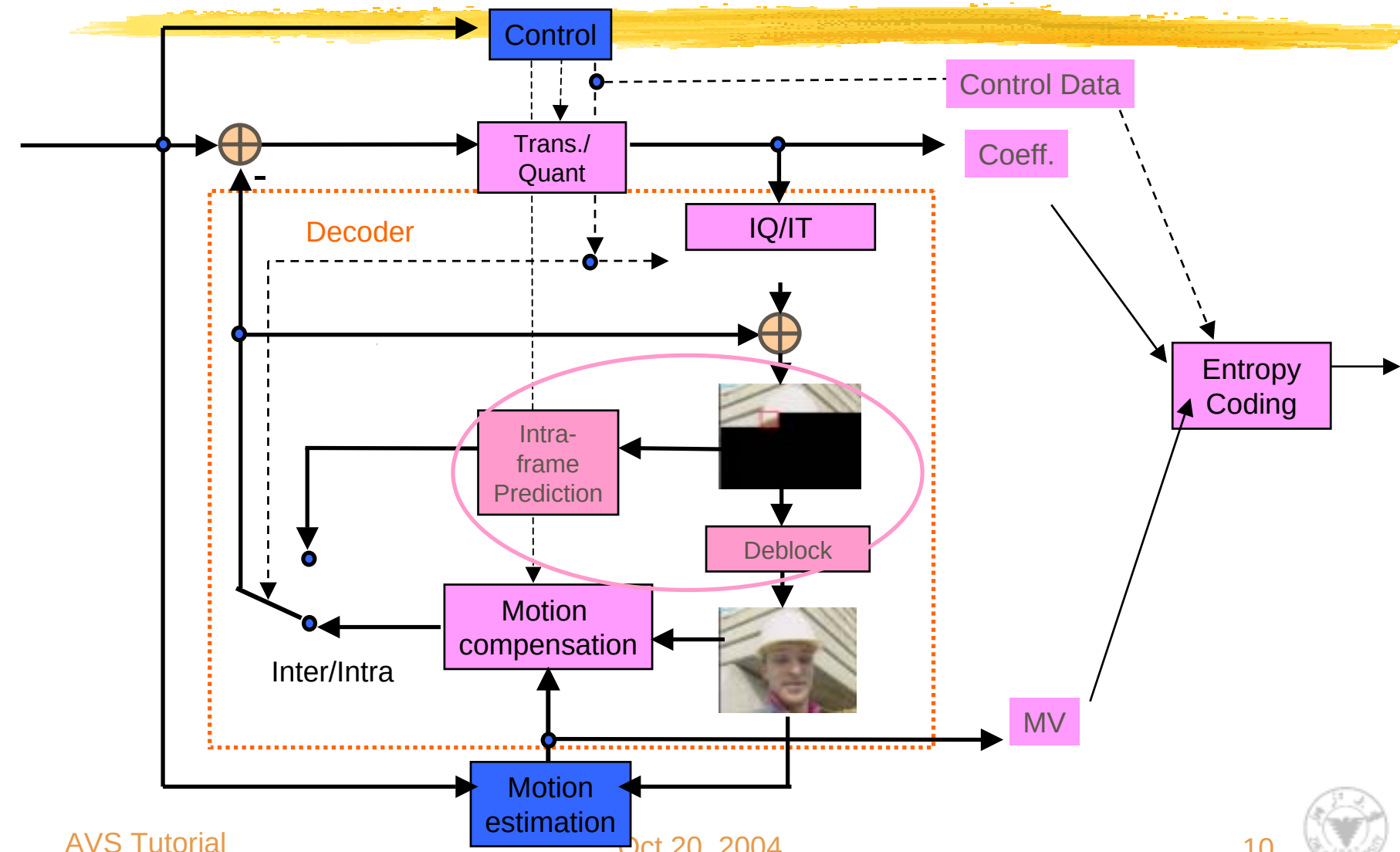
MPEG-2 Video


Control
Control Data
Trans./
Quant
Coeff.
Decoder
IQ/IT
Entropy
Coding
0
Motion
compensation
Inter/Intra
MV
Motion
estimation

AVS Video Version 1 (AVS-HD)



AVS Video Version 1 (AVS-HD)



Profile and Level

- JiZhun (基准) Profile (Dec. 2003)
  - 4 level
    - 4.0 : up to Standard Definition with 4:2:0
    - 4.2 : up to Standard Definition with 4:2:2
    - 6.0 : up to High Definition with 4:2:0
    - 6.2 : up to High Definition with 4:2:2
- X Profile (2005)

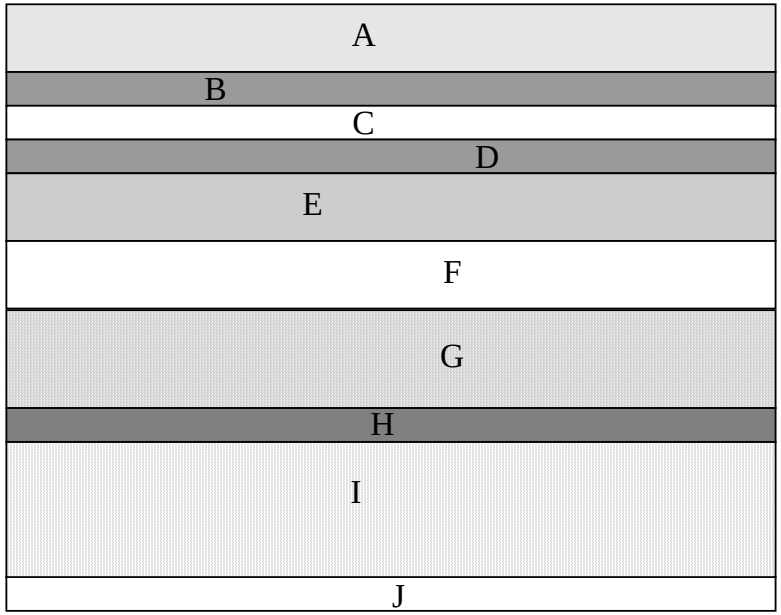
AVS Video 1.0 Tools

- AVS video 1.0 accepted 42 technical proposals from more than 200.
- Major tools
  - Transform – 16bit-implemented 8x8 integer transform
  - Quantization and scaling - scaling only in encoder
  - Intra prediction – 5 modes
  - Motion compensation – 16x16/16x8/8x16/8x8 modes
  - Quarter-pel interpolation – 4-taps interpolation filter
  - Deblocking
  - Entropy coding
- Minor tools
  - Motion vector prediction
  - Skipped mode and Coded block pattern
  - Adaptive scan

Bit Stream Structure

- Sequence
- Frame
- Slice
- Macroblock
- Block

- Block size:
  - 8x8-based

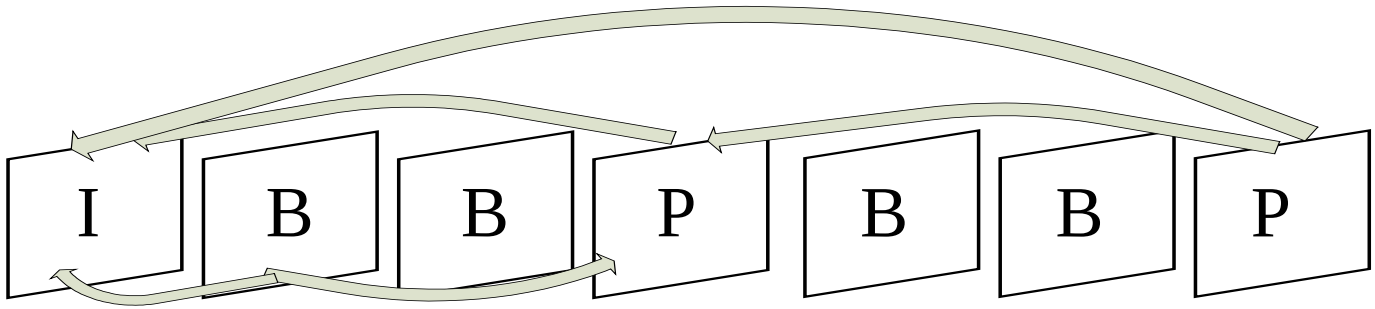


Basic Block Size – 8x8 vs. 4x4

Sequence	Foreman	Coastguard	Container	Mother Daughter	
Size	176×144	176×144	176×144	176×144	QCIF
△PSNR (dB)	-0.16	0.02	-0.91	-0.53	-0.394
△Rate (%)	0.04	-0.01	0.20	0.12	0.0882
Sequence	Foreman	Coastguard	Container	Mother Daughter	
Size	352×288	352×288	352×288	352×288	CIF
△PSNR (dB)	-0.063	0.386	-0.363	-0.192	-0.058
△Rate (%)	1.56	-9.77	9.93	5.05	1.6925
Sequence	Tempete	Mobile	Flowergarden	Football	
Size	352×288	352×288	352×240	352×240	SIF
△PSNR (dB)	-0.185	-0.767	-0.493	-0.231	-0.419
△Rate (%)	4.5	17.54	8.93	4.15	8.78
Sequence	Tempete	Mobile	Flowergarden	Football	
Size	720×480	720×576	720×576	720×576	CCIR
△PSNR (dB)	0.009	-0.442	-0.197	0.147	-0.121
△Rate (%)	-0.24	40.34	3.82	-2.67	10.313
Sequence	Night	City	Crew	Harbour	
Size	1280×720	1280×720	1280×720	1280×720	HD-p
△PSNR (dB)	0.092	0.144	0.27	0.46	0.2024
△Rate (%)	-2.85	-5	-9.12	-12.8	-6.564
Sequence	Spin&Calendar	Flamingo	Fireworks	Kayak	
Size	1280×720	1920×1088	1920×1088	1920×1088	HD-I
△PSNR (dB)	0.046	0.286	-0.32	0.519	0.1617
△Rate (%)	-3.05	-5.46	5.19	-8.53	-2.933

- JM5.0
- 4x4
  - Transform
  - Intra/Inter pred all modes enable
- 8x8
  - Transform
  - Intra/Inter pred modes no smaller then 8x8 enabled

Picture Type

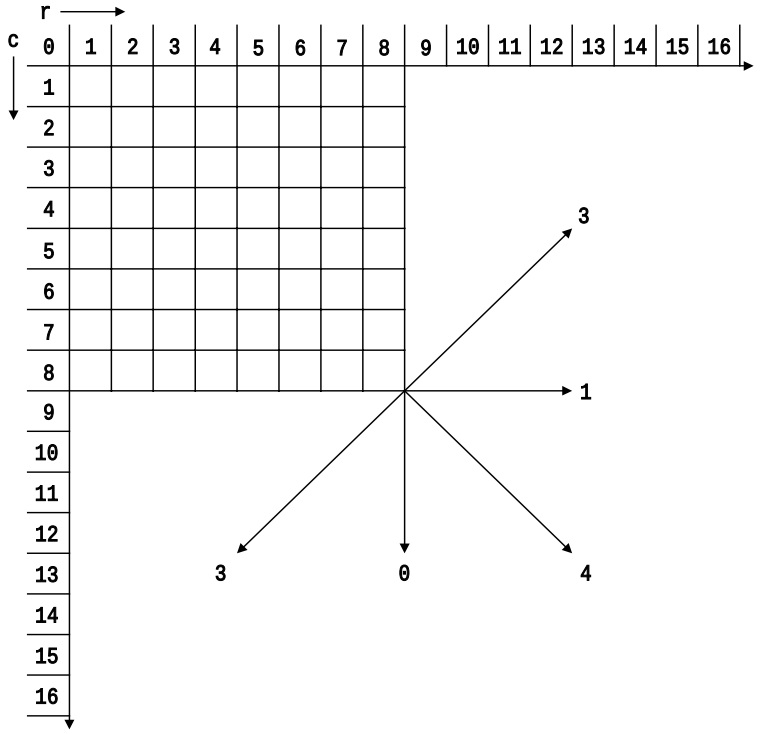


- Picture type:
  - I-frame
  - P-frame: at most 2 reference frames
  - B-frame: 2 reference frames

Intra-frame Prediction and Compensation

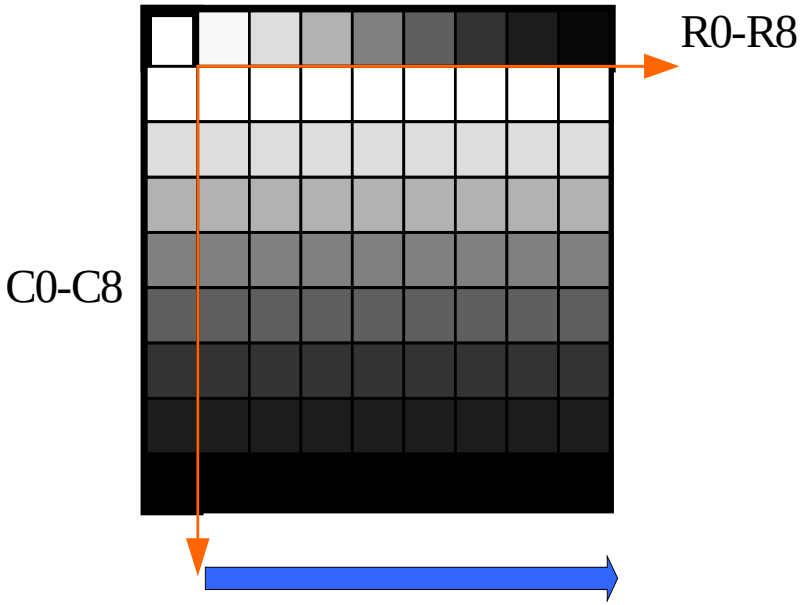
Intra Prediction Modes

- 8x8-based
- Luma 5 modes:
  - 0. Intra_8x8_Vertical
  - 1. Intra_8x8_Horizontal
  - 2. Intra_8x8_DC
  - 3. Intra_8x8_Down_Left
  - 4. Intra_8x8_Down_Right
- Chroma 4 modes:
  - 0. Intra_Chroma_DC
  - 1. Intra_Chroma_Horizontal
  - 2. Intra_Chroma_Vertical
  - 3. Intra_Chroma_Plane

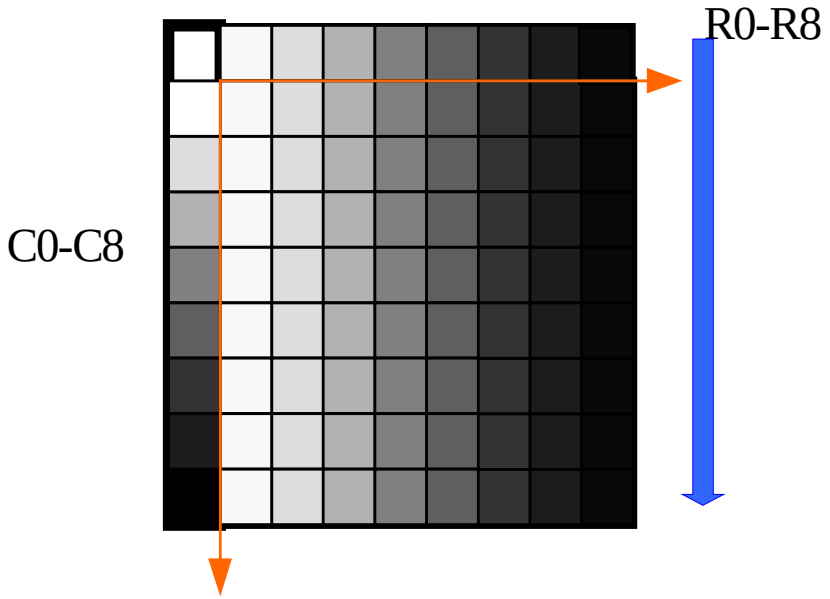


Luma-Vertical & Horizontal Mode

- Intra_8x8_Vertical
- Intra_8x8_Horizontal

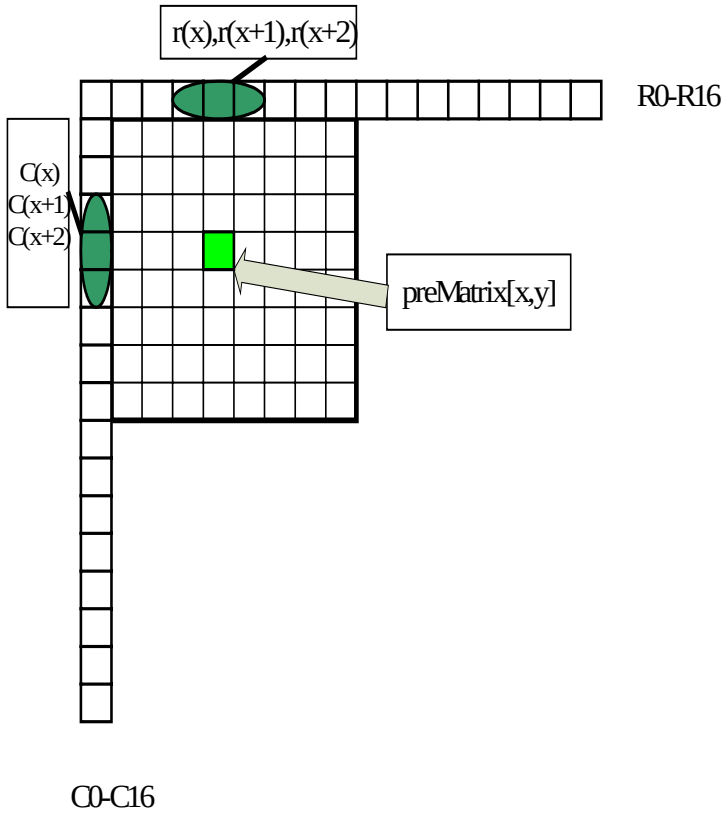


Horizontal



Vertical

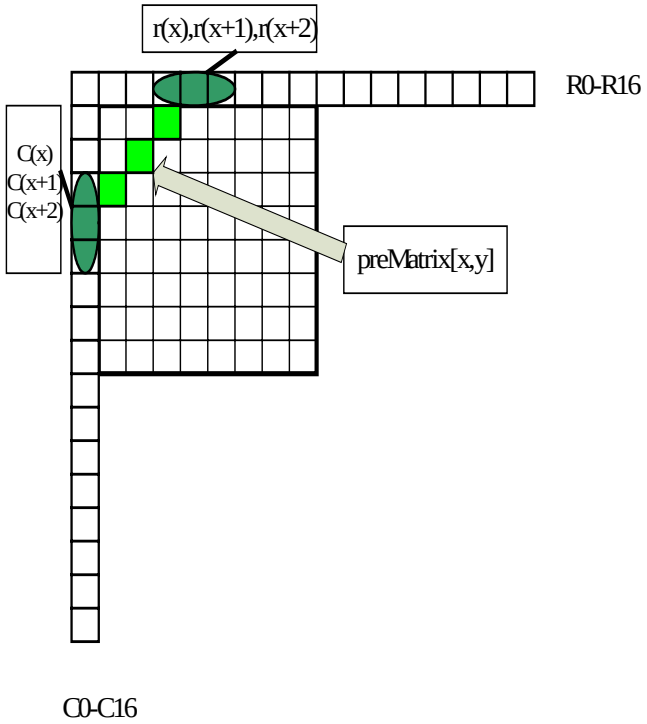
Luma-DC Mode

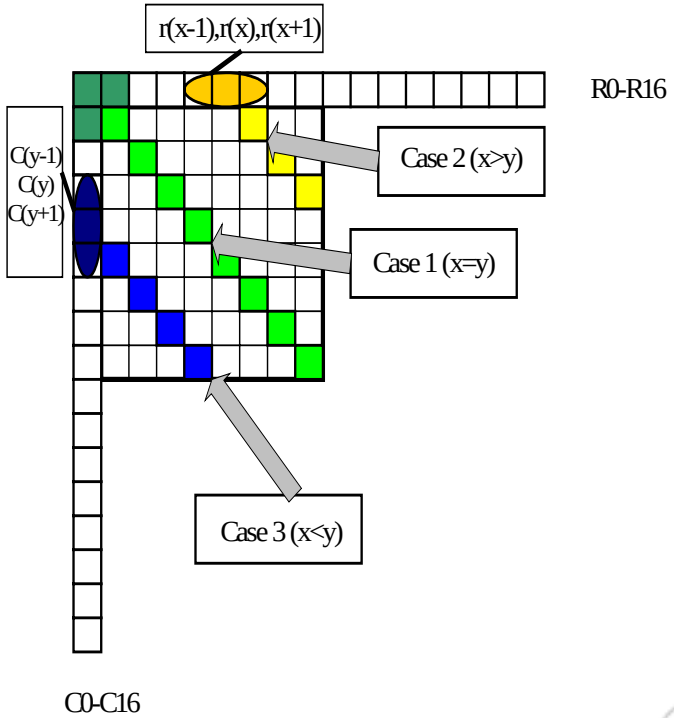


- Intra_8x8_DC
  - The prediction for each pixel is different
  - Low-pass filter

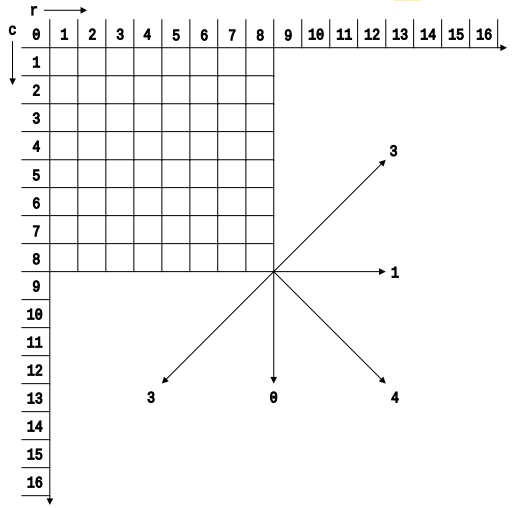
Luma Down_Left & Down_Right Mode

- Only if r[i]、c[i] (i=1..16) all available, use these two mode
  - Intra_8x8_Down_Left (left)
  - Intra_8x8_Down_Right (right)

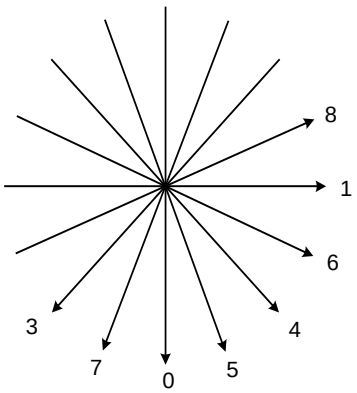




Intra Luma - Compare with H.264

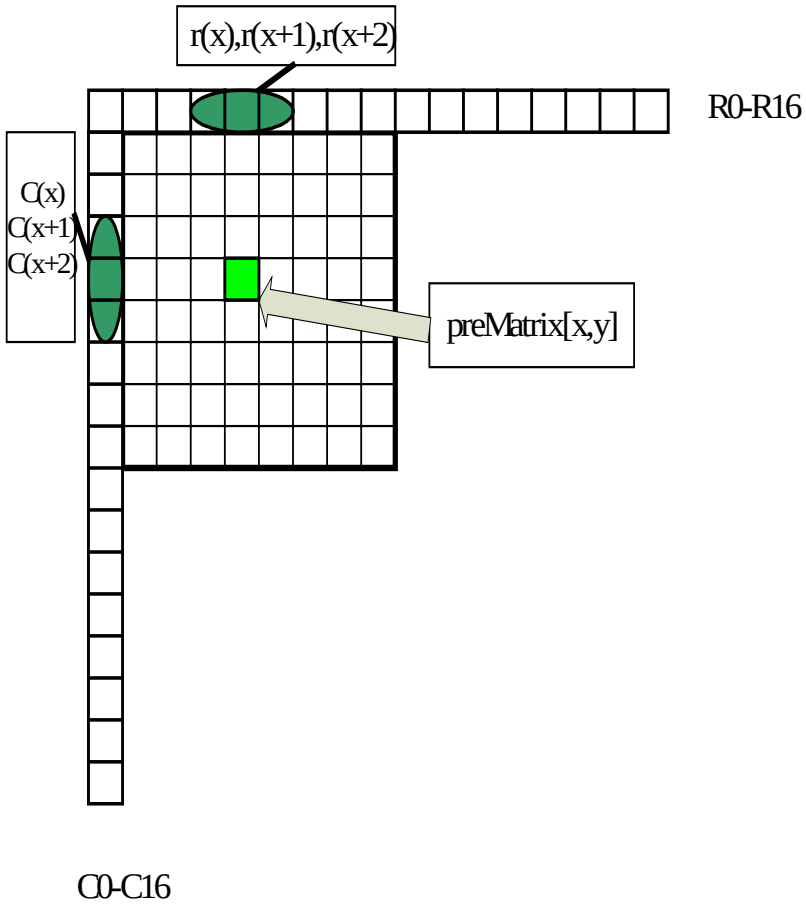


- AVS :
  - 8x8 intra predict
  - 5 modes
  - DC with lowpass filter
  - Advantages:
    - less complexity with less modes



- H.264 :
  - Adaptive 4x4 and 16x16 intra prediction
  - 9 modes for 4x4 + 4 modes for 16x16
  - Advantages:
    - better prediction
  - Disadvantages:
    - more complexity

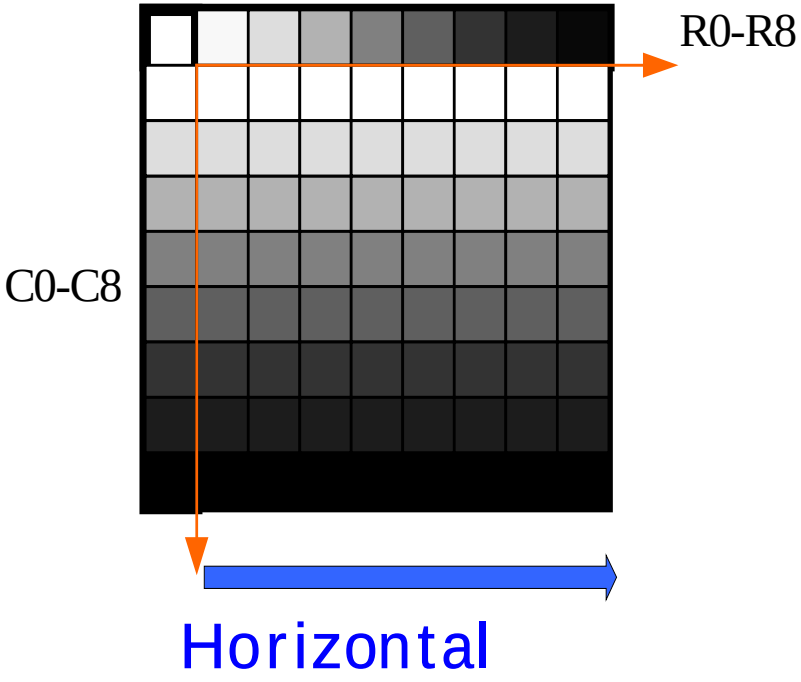
Chroma DC mode

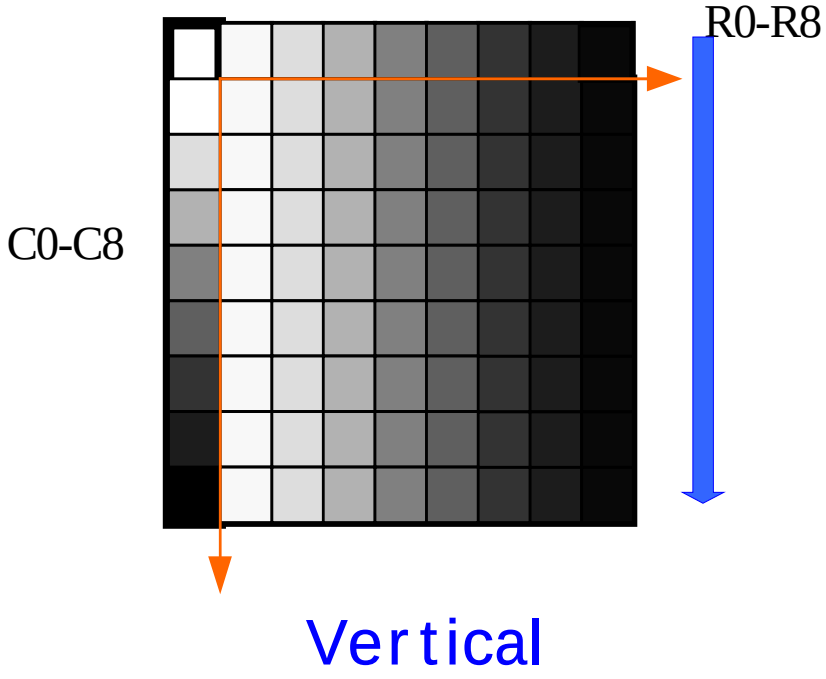


- Intra_Chroma_DC
  - The prediction for each pixel is different
  - Low-pass filter

Chroma Horizontal & Vertical mode

- Intra_Chroma_Horizontal
- Intra_Chroma_Vertical





Chroma Plane Mode

- Intra_Chroma_Plane
- If r[i]、c[i] (i=0..9) all available

predMatrix[x,y] = Clip1((ia+(x-3)×ib+(y-3)×ic+16)>>5) (x,y=0..7)

In which,

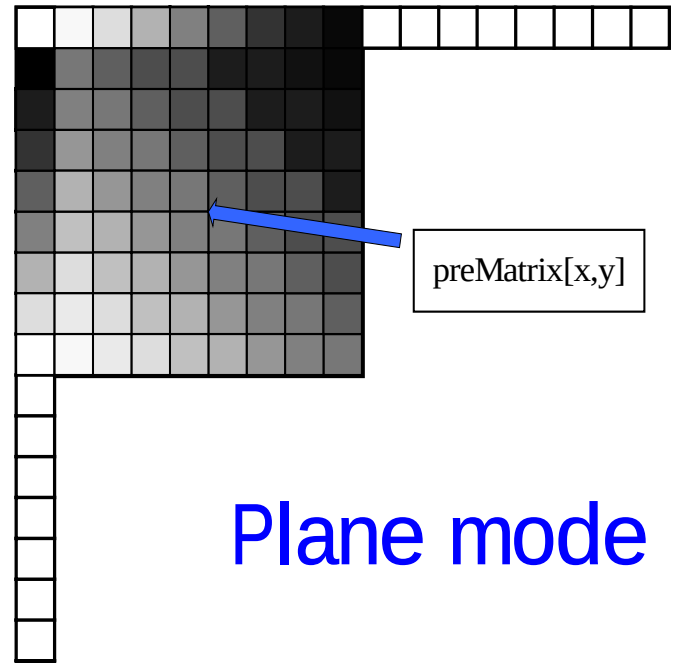
ia = (r[8]+c[8])<<4,

ib = (17×ih+16)>>5,

ic = (17×iv+16)>>5,

ih = Σ(i+1)×(r[5+i]-r[3-i]) ,

iv = Σ(i+1)×(r[5+i]-r[3-i])

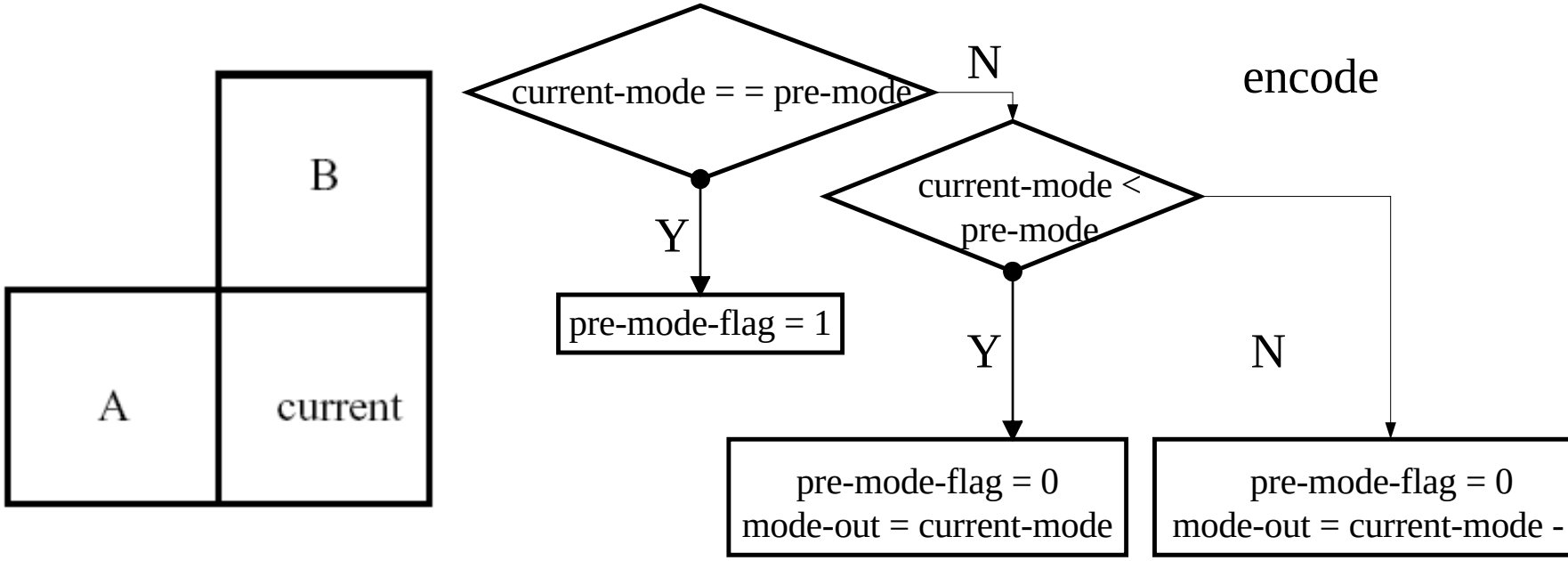


Plane mode

Intra Chroma - Compare with H.264

- Differences:
  - Transform block size
    - AVS: 8x8-based integer DCT transform
    - H.264: 4x4-based integer DCT transform + hadamard DC transform
  - Chroma-DC
    - AVS's DC: low-pass filter of the surrounding pixels
    - H.264's DC: mean of all the surrounding pixels

Luma Intra Modes Coding



Pre_mode=min(modeA, modeB)

Intra Modes Coding

- Luma
  - 4 blocks in each macroblock
  - Each block
    - 1 bit pre_mode-flag
    - 2 bits intra_luma_pred_mode (if necessary)
- Chroma
  - 2(4:2:0) or 4(4:2:2) 8x8 blocks in each macroblock
  - Each pair of blocks (U & V)
    - intra_chroma_pred_mode: 0-order Exp-Golomb entropy coding

Transform and Quantization

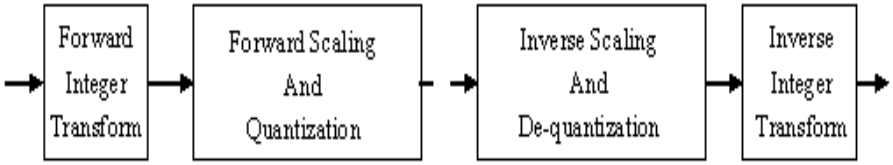
Characteristics of AVS' Transform

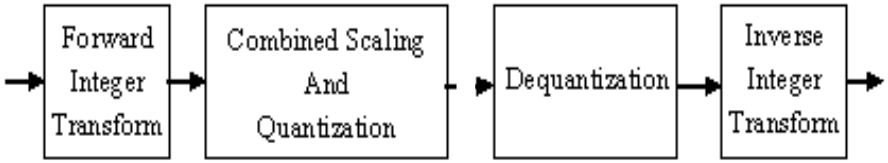
- 8x8-based
  - 8x8 inter-frame and intra-frame prediction better for high-definition video
- 16-bit implementation with no mis-match
- Asymmetric pre-scaled transform

AVS-video transform

- 2003.3 (4th)decision:
  - 8x8-transform-only for HD
  - Common condition
- 2003.7.4 (video group meeting) accept:
  - avs-m1075, 8x8 integer transform and relative quantization [10 9 6 2], from ZJU, because of better performance (at lease 0.05dB averagely)
- 2003.7.31 (5th) accept:
  - avs-m1103, 8x8 integer transform and relative quantization [5 4 3 1] , from ICT, because of lower computational complexity
  - 1-D trans: 18 shift & 40 add vs forward trans. 6 shift & 32 add and inverse trans. 6 shift 28 add
- 2003.8.31 (video group meeting) accept:
  - avs-m1115, 8×8 integer transform and relative quantization [10 9 6 2], from ZJU, because of better performance (0.1dB averagely), especially for I frame (0.3dB gain)
- 2003.10 (6th)
  - Ad hoc group of transform
  - 16-bit implementation of transform
- 2003.11.6&10 (video group meeting) :
  - avs-m1178, 16-bit 8×8 integer transform and relative quantization, from digipro
  - avs-m1182, 8×8 integer transform and relative quantization, from ZJU & CUHK
  - Over-nights cross-check
  - 2003.11.15 final decision

Asymmetric Pre-scaled Transform





- Normal integer transform and quantization
  - encoder
    - Forward transform
    - Forward scaling
    - Quantization
  - decoder
    - Inverse quantization
    - Inverse scaling
    - Inverse transform
- AVS integer transform and quantization
  - encoder
    - Forward transform
    - Forward scaling and inverse scaling
    - Quantization
  - decoder
    - Inverse quantization
    - Inverse transform

Transform Kernel Matrix of AVS

$$T_8 = \begin{bmatrix} 8 & 10 & 10 & 9 & 8 & 6 & 4 & 2 \\ 8 & 9 & 4 & -2 & -8 & -10 & -10 & -6 \\ 8 & 6 & -4 & -10 & -8 & 2 & 10 & 9 \\ 8 & 2 & -10 & -6 & 8 & 9 & -4 & -10 \\ 8 & -2 & -10 & 6 & 8 & -9 & -4 & 10 \\ 8 & -6 & -4 & 10 & -8 & -2 & 10 & -9 \\ 8 & -9 & 4 & 2 & -8 & 10 & -10 & 6 \\ 8 & -10 & 10 & -9 & 8 & -6 & 4 & -2 \end{bmatrix}$$

Quantization

- Quant Parameter QP∈ [0..63]
- Quantization
  - QuantCoeff [i,j] =(Coeff [i,j] ×Quant (QP)+$2^{QuantShift\ (QP)-1}$) >> QuantShift (QP)
- Inverse quantization
  - Coeff ‘[i,j] =(QuantCoeff [i,j]×Dequant (QP)+$2^{DequantShift\ (QP)-1}$) >> DequantShift (QP)
- Quant (QP)x Dequant (QP) ≈ $2^{QuantShift\ (QP)+\ DequantShift\ (QP)}$
- Nearly **8**-order exponential periodic

Inverse quant table

QP	0	1	2	3	4	5	6	7
DequantTable(QP)	32768	36061	38968	42495	46341	50535	55437	60424
ShiftTable(QP)	14	14	14	14	14	14	14	14
QP	8	9	10	11	12	13	14	15
DequantTable(QP)	32932	35734	38968	42495	46177	50535	55109	59933
ShiftTable(QP)	13	13	13	13	13	13	13	13
QP	16	17	18	19	20	21	22	23
DequantTable(QP)	65535	35734	38968	42577	46341	50617	55027	60097
ShiftTable(QP)	13	12	12	12	12	12	12	12
QP	24	25	26	27	28	29	30	31
DequantTable(QP)	32809	35734	38968	42454	46382	50576	55109	60056
ShiftTable(QP)	11	11	11	11	11	11	11	11
QP	32	33	34	35	36	37	38	39
DequantTable(QP)	65535	35734	38968	42495	46320	50515	55109	60076
ShiftTable(QP)	11	10	10	10	10	10	10	10
QP	40	41	42	43	44	45	46	47
DequantTable(QP)	65535	35744	38968	42495	46341	50535	55099	60087
ShiftTable(QP)	10	9	9	9	9	9	9	9
QP	48	49	50	51	52	53	54	55
DequantTable(QP)	65535	35734	38973	42500	46341	50535	55109	60097
ShiftTable(QP)	9	8	8	8	8	8	8	8
QP	56	57	58	59	60	61	62	63
DequantTable(QP)	32771	35734	38965	42497	46341	50535	55109	60099
ShiftTable(QP)	7	7	7	7	7	7	7	7

Entropy coding

Characteristics of AVS Entropy Coding

- Exp-Golomb coding for all syntax elements
  - Transform coefficients
    - 2D-VLC (run-level pair)
    - 19 VLC-tables
    - Adaptive switch VLC-table by 'level'
    - Special ESCAPE tools (0.05~0.08dB gain)
      - 'run' hidden in 'escape'
      - Use escape-'delta-level' instead of 'level'
  - Ohters syntax elements
    - Exp-Golomb coding with or without mapping

Entropy Coding of Transform Coefficients

Run	EOB	Level > 0									RefAbsLevel
		1	2	3	4	5	6	7	8	9	
	2	-	-	-	-	-	-	-	-	-	
0	-	0	3	7	13	17	27	35	43	55	10
1	-	5	11	21	33	51	-	-	-	-	6
2	-	9	23	37	57	-	-	-	-	-	5
3	-	15	29	47	-	-	-	-	-	-	4
4	-	19	41	-	-	-	-	-	-	-	3
5	-	25	49	-	-	-	-	-	-	-	3
6	-	31	-	-	-	-	-	-	-	-	2
7	-	39	-	-	-	-	-	-	-	-	2
8	-	45	-	-	-	-	-	-	-	-	2
9	-	53	-	-	-	-	-	-	-	-	2

- Level>0
  - direct CodeNum (run, level)
- level<0
  - CodeNum (run, -level) +1

Example

Run	EOB	Level > 0									RefAbsLevel
		1	2	3	4	5	6	7	8	9	
	2	-	-	-	-	-	-	-	-	-	
0	-	0	3	7	13	17	27	35	43	55	10
1	-	5	11	21	33	51	-	-	-	-	6
2	-	9	23	37	57	-	-	-	-	-	5
3	-	15	29	47	-	-	-	-	-	-	4
4	-	19	41	-	-	-	-	-	-	-	3
5	-	25	49	-	-	-	-	-	-	-	3
6	-	31	-	-	-	-	-	-	-	-	2
7	-	39	-	-	-	-	-	-	-	-	2
8	-	45	-	-	-	-	-	-	-	-	2
9	-	53	-	-	-	-	-	-	-	-	2

- Run = 1, Level = 2

  CodeNum = 11

- Run = 1, Level = -2

  CodeNum = 11+1

Entropy Coding of Transform Coefficients with ESCAPE

Run	EOB	Level > 0									RefAbsLevel
		1	2	3	4	5	6	7	8	9	
	2	-	-	-	-	-	-	-	-	-	
0	-	0	3	7	13	17	27	35	43	55	10
1	-	5	11	21	33	51	-	-	-	-	6
2	-	9	23	37	57	-	-	-	-	-	5
3	-	15	29	47	-	-	-	-	-	-	4
4	-	19	41	-	-	-	-	-	-	-	3
5	-	25	49	-	-	-	-	-	-	-	3
6	-	31	-	-	-	-	-	-	-	-	2
7	-	39	-	-	-	-	-	-	-	-	2
8	-	45	-	-	-	-	-	-	-	-	2
9	-	53	-	-	-	-	-	-	-	-	2

- ESCAPE
  - (run, level) combination not included in table
  - the probability is low
- One ESCAPE event 2 codes:
  - ESCAPE flag (CodeNum = 59) + run
  - Delta-Level
- Reason:
  - CodeNum = 59~123 has the same code-length
  - Recovered Level = RefAbsLevel+ Delta-Level

Example

Run	EOB	Level > 0									RefAbsLevel
		1	2	3	4	5	6	7	8	9	
	2	-	-	-	-	-	-	-	-	-	
0	-	0	3	7	13	17	27	35	43	55	10
1	-	5	11	21	33	51	-	-	-	-	6
2	-	9	23	37	57	-	-	-	-	-	5
3	-	15	29	47	-	-	-	-	-	-	4
4	-	19	41	-	-	-	-	-	-	-	3
5	-	25	49	-	-	-	-	-	-	-	3
6	-	31	-	-	-	-	-	-	-	-	2
7	-	39	-	-	-	-	-	-	-	-	2
8	-	45	-	-	-	-	-	-	-	-	2
9	-	53	-	-	-	-	-	-	-	-	2

- Run = 2, Level = 6:
  - 2 codes:
    - CodeNum 1 = 59+2xrun=63
    - CodeNum 2 = delta-level = 6-5=1

Entropy Coding Compare with H.264

- AVS
  - Exp-Golomb coding for all syntax elements including transform coefficients
  - Advantages:

    Less complexity
- H.264
  - Exp-Golomb coding for syntax elements except for transform coefficients
  - For transform coefficients: CAVLC in Baseline Profile and CABAC in Main Profile

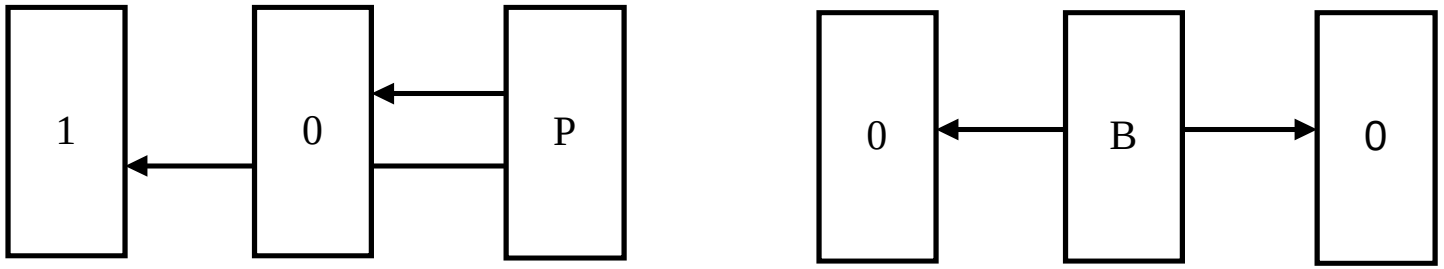
Inter-Picture Prediction and Compensation

Inter-Frame Prediction and Compensation

- Picture-level frame/field adaptive coding

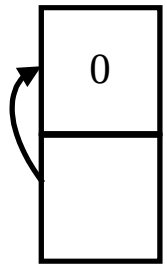
- P-frame frame-coding
- B-frame frame-coding
- I-frame field-coding
- P-frame field-coding
- B-frame field-coding

Reference Picture Index – Frame-Based



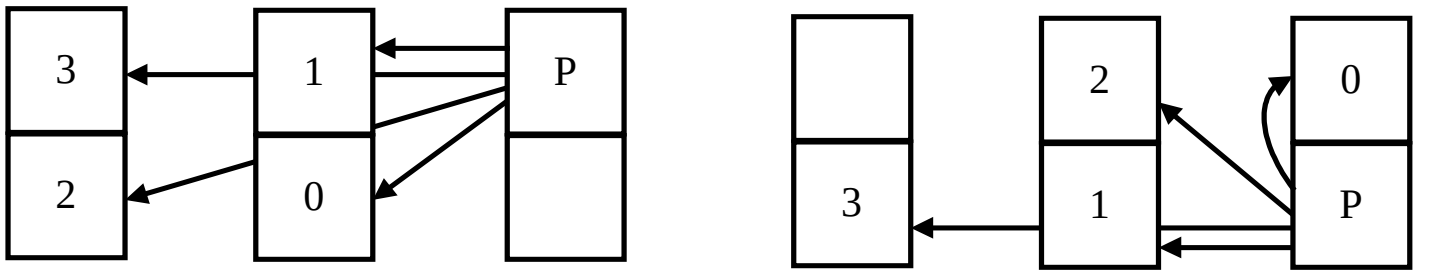
- P-frame frame-coded
  - At most 2 reference frames
  - Each motion compensation block 1 reference frame
- B-frame frame-coded
  - 1 forward reference frame, and
  - 1 backward reference frame

Reference Picture Index – Field-Based (1)



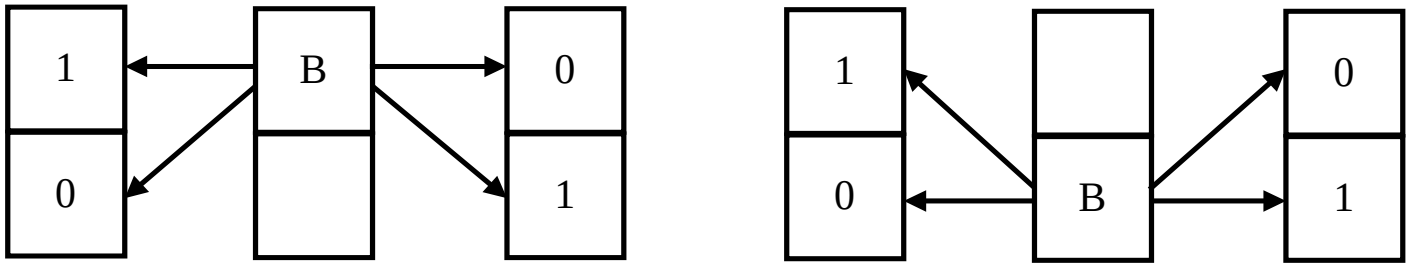
- I-frame field-coded
  - First field: intra prediction
  - Second field: predicted from the top field

Reference Picture Index – Field-Based (2)



- P-frame field-coded
  - First field: predicted from the closest previous coded 4 fields
  - Second field: predicted from the closest previous coded 4 fields, containing the top field of current frame

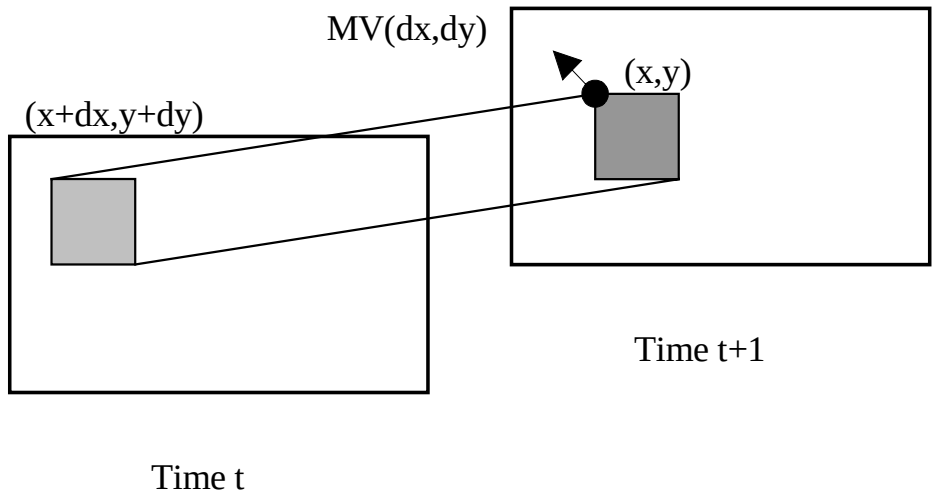
Reference Picture Index – Field-Based (3)



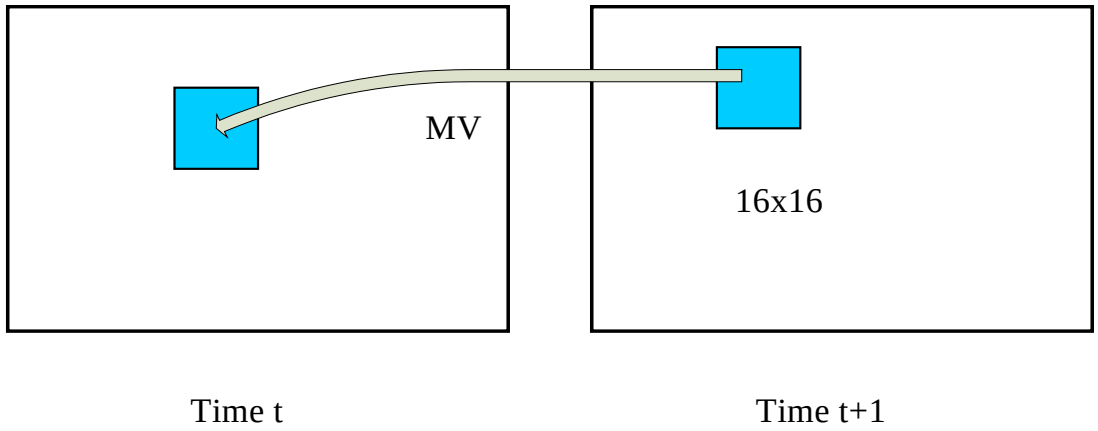
- B-frame field coded
  - First field: predicted from the forward and/or backward frame's top or bottom filed
  - Second field: predicted from the forward and/or backward frame's top or bottom filed

P-frame

- Macroblock modes:
  - P_Skip
  - P_16x16
  - P_16x8
  - P_8x16
  - P_8x8
  - I_8x8

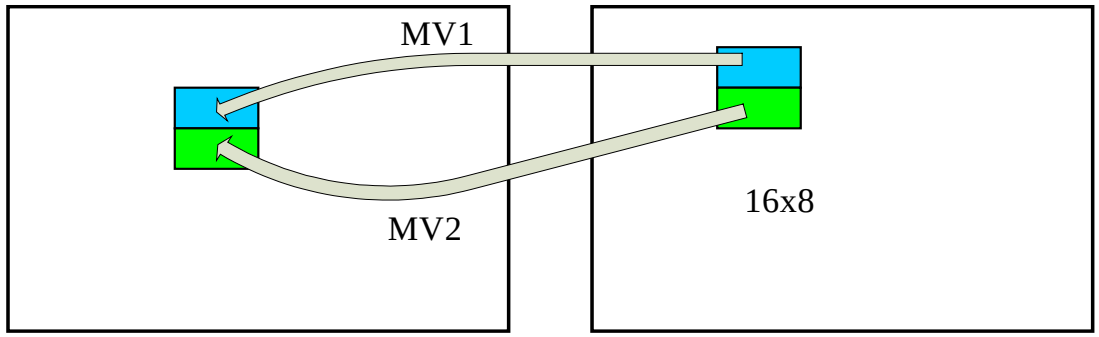


P_16x16 , P_Skip



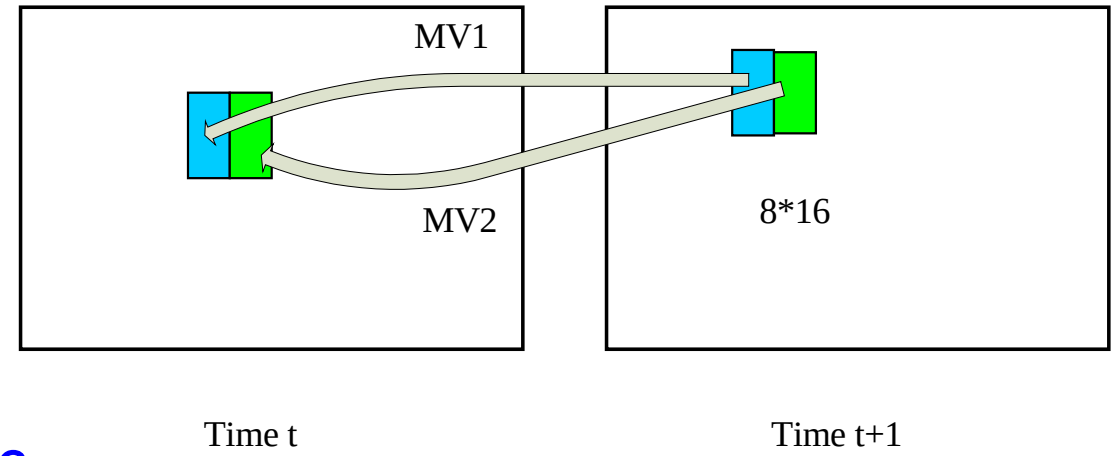
- P_16x16
  - 1 reference picture
  - 1 Motion vector (MVx, MVy) each macroblock
- P_Skip
  - 1 reference picture
  - 1 Motion vector (MVx, MVy) =predict MV
  - No coded motion vector

P_16x8



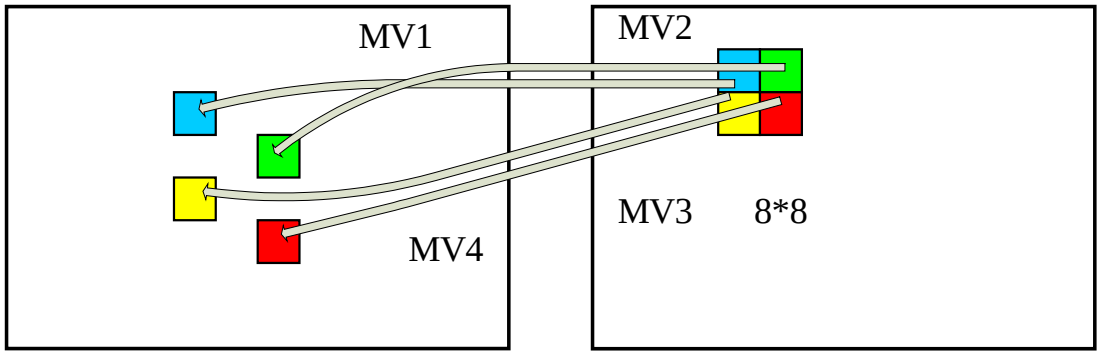
- P_16x8:
  - 2 motion vectors each macroblock
  - 1 motion vector each 16x8 block
  - Top and bottom 16x8 blocks have different motion vectors
  - Top and bottom 16x8 blocks may have at most 2 different reference pictures

P_8x16



- P_8x16:
  - 2 motion vectors each macroblock
  - 1 motion vector each 8x16 block
  - Right and left 8x16 blocks have different motion vectors
  - Right and left 8x16 blocks may have at most 2 different reference pictures

P_8x8



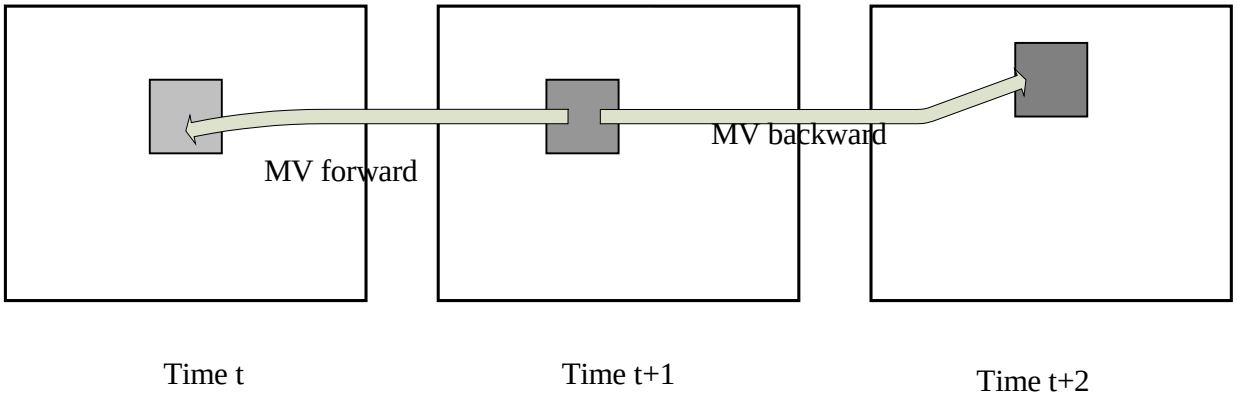
- P_8x8:
  - 4 motion vectors each macroblock
  - 1 motion vector each 8x8 block
  - Each 8x8 block has different motion vector
  - 4 8x8 blocks may have at most 2 different reference pictures

P Modes Compare with H.264

- Motion compensation block size
  - AVS – 8x8 or bigger
  - H.264 – 4x4 or bigger
- Always 2 reference frames

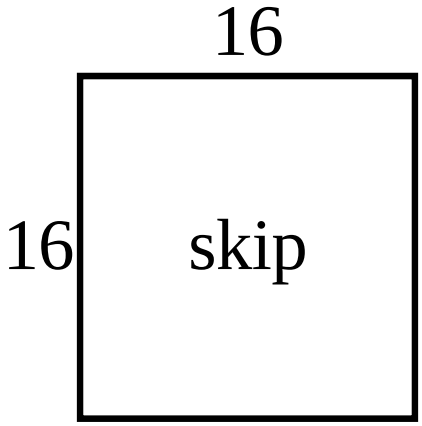
- Advantages
  - Block-size
    - Less complexity especially in encoding side
    - Save bits in transmiting motion vectors and modes
    - Better de-correlation spacially
  - 2 reference frames
    - Take full advantage of memory and bandwidth

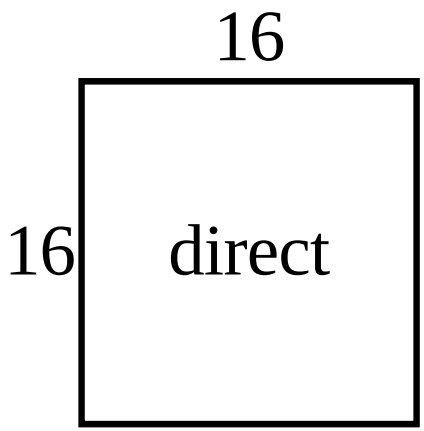
B-frame



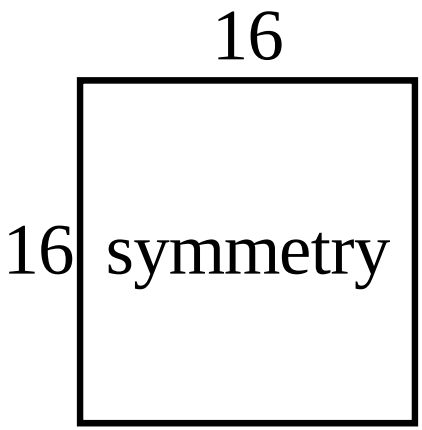
- 24 Macroblock modes:
  - 16x16: B_Skip, B_Direct, B_Fwd, B_Bck, B_Sym
  - 16x8: B_Fwd_Fwd, B_Bck_Bck, B_Fwd_Bck, B_Bck_Fwd, B_Fwd_Sym, B_Bck_Sym, B_Sym_Fwd, B_Sym_Bck, B_Sym_Sym
  - 8x16: B_Fwd_Fwd, B_Bck_Bck, B_Fwd_Bck, B_Bck_Fwd, B_Fwd_Sym, B_Bck_Sym, B_Sym_Fwd, B_Sym_Bck, B_Sym_Sym
  - 8x8: B_8x8, I_8x8
    - B_8x8: sub modes

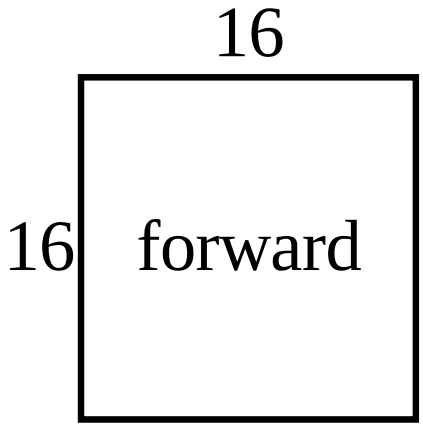
B_16x16 Modes

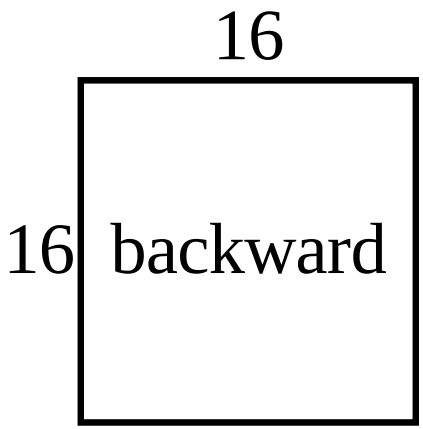




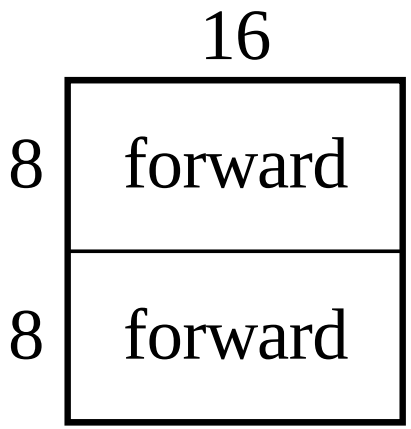
- B_Skip
- B_Direct_16x16
- B_Sym_16x16
- B_Fwd_16x16
- B_Bck_16x16



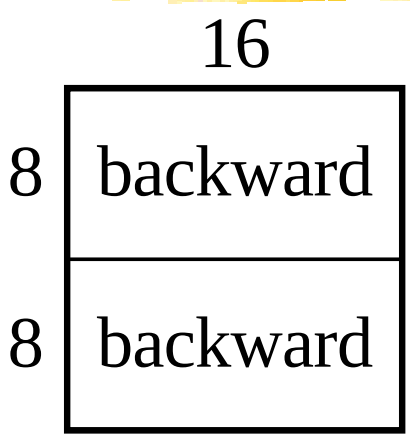




B_16x8 Modes

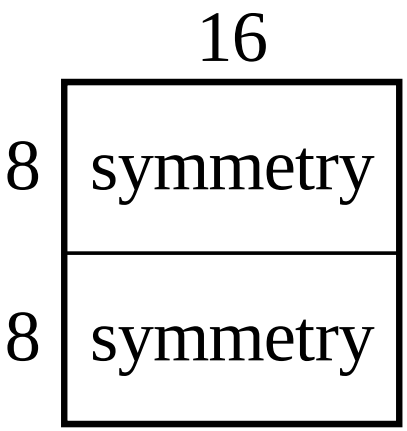


B_fwd_fwd_16*8

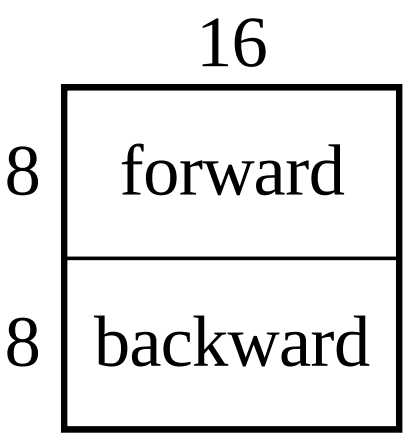


B_bck_bck_16*8

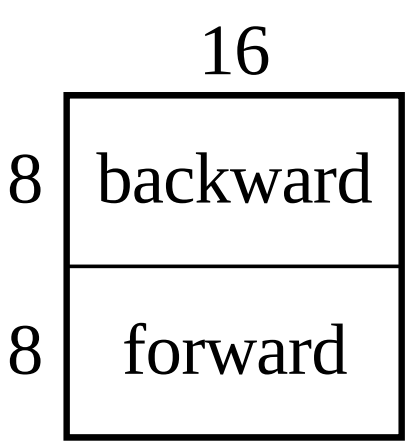
- B_Fwd_Fwd_16x8
- B_Bck_Bck_16x8
- B_Sym_Sym_16x8
- B_Fwd_Bck_16x8
- B_Bck_Fwd_16x8



B_sym_sym_16*8

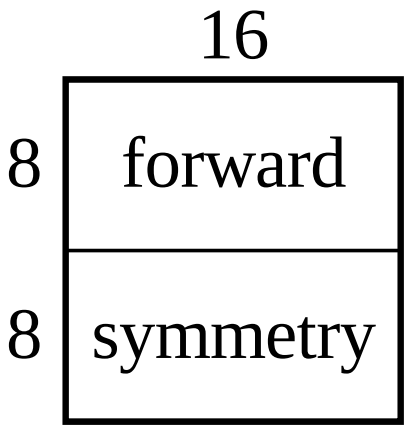


B_fwd_bck_16*8

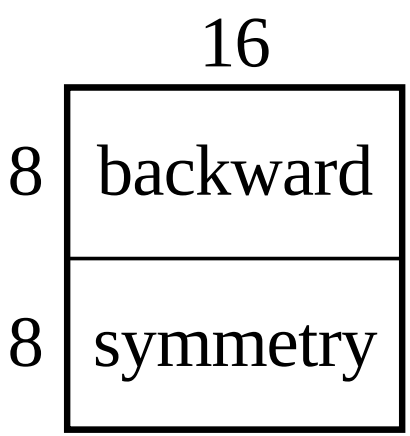


B_bck_fwd_16*8

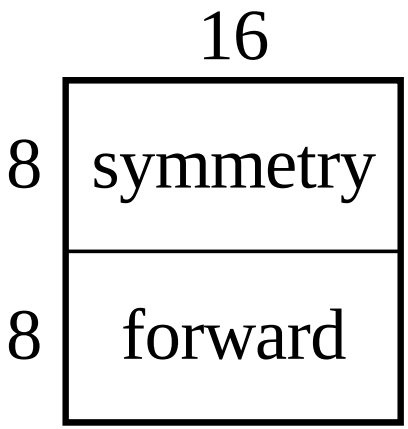
B_16x8 Modes (continued)



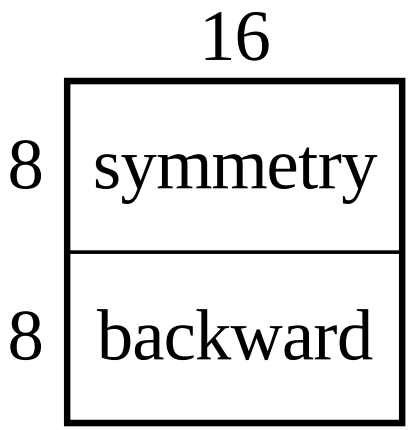
B_fwd_sym_16*8



B_bck_sym_16*8



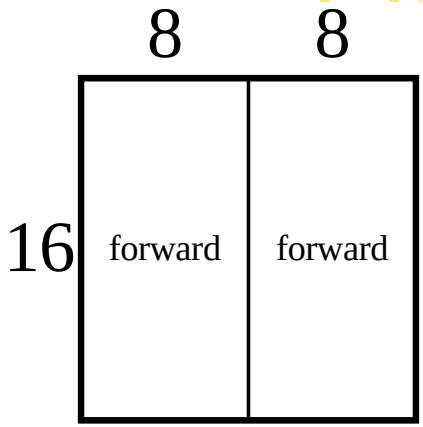
B_sym_fwd_16*8



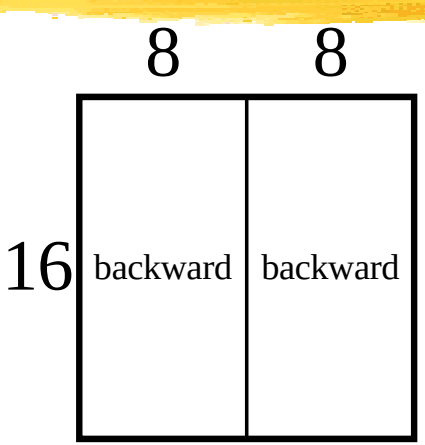
B_sym_fwd_16*8

- B_Fwd_Sym_16x8
- B_Bck_Sym_16x8
- B_Sym_Fwd_16x8
- B_Sym_Bck_16x8

B_8x16 Modes

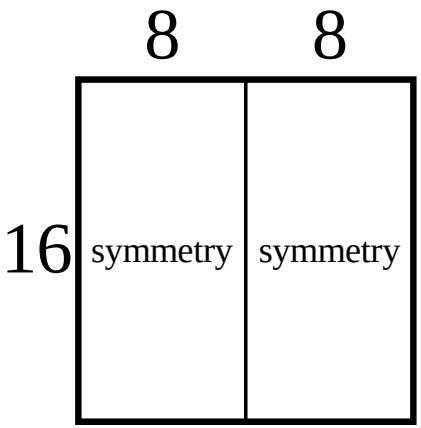


B_fwd_fwd_8*16

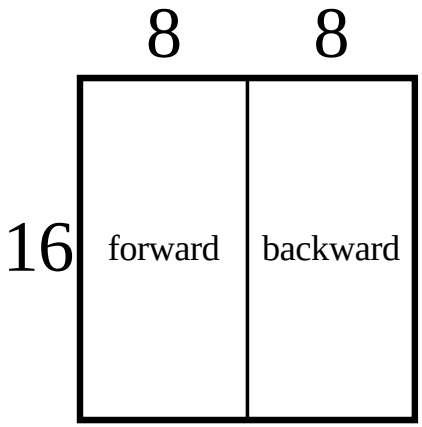


B_bck_bck_8*16

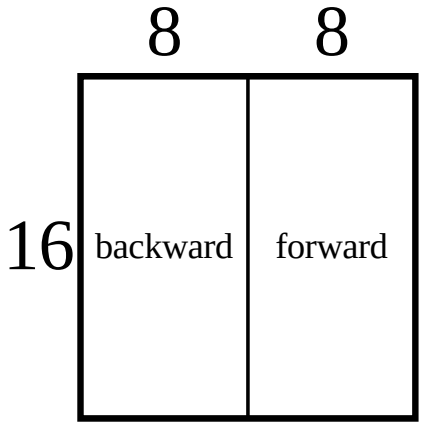
- B_Fwd_Fwd_8x16
- B_Bck_Bck_8x16
- B_Sym_Sym_8x16
- B_Fwd_Bck_8x16
- B_Bck_Fwd_8x16



B_sym_sym_8*16

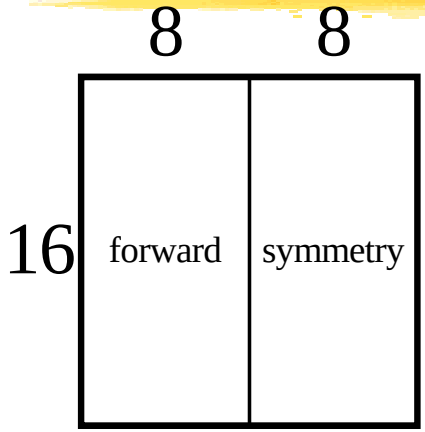


B_fwd_bck_8*16

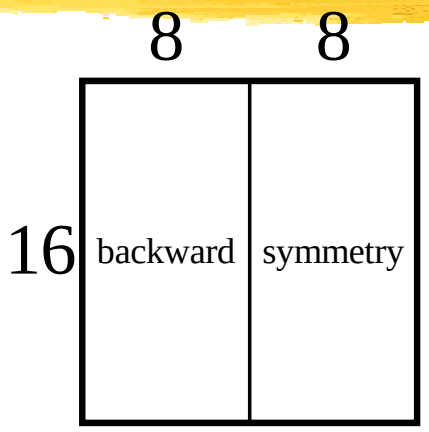


B_bck_fwd_8*16

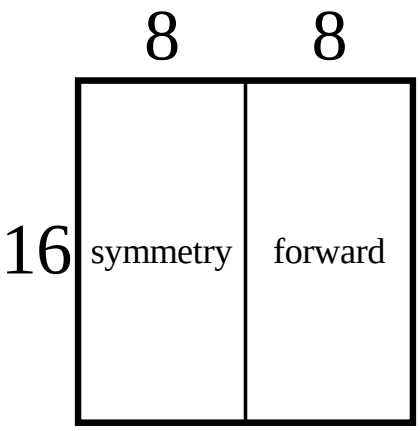
B_8x16 Modes (continued)



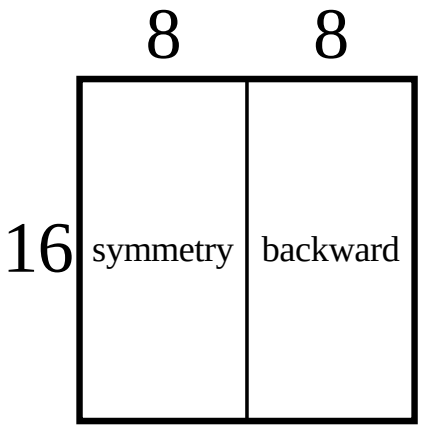
B_fwd_sym_8*16



B_bck_sym_8*16



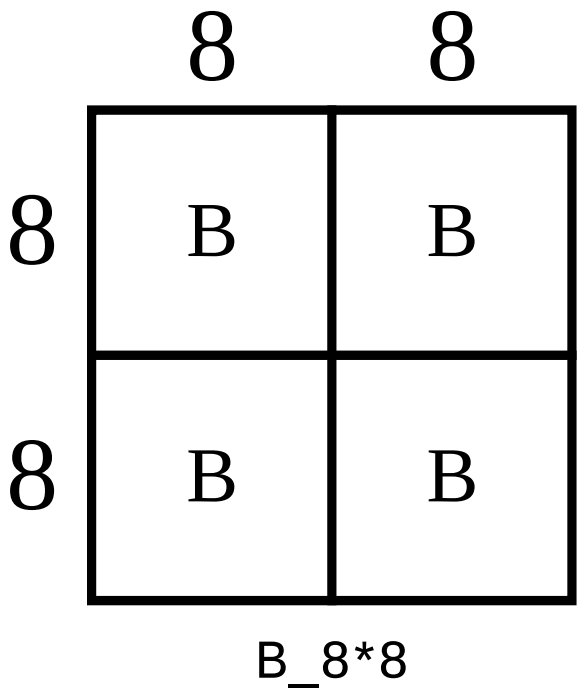
B_sym_fwd_8*16



B_sym_bck_8*16

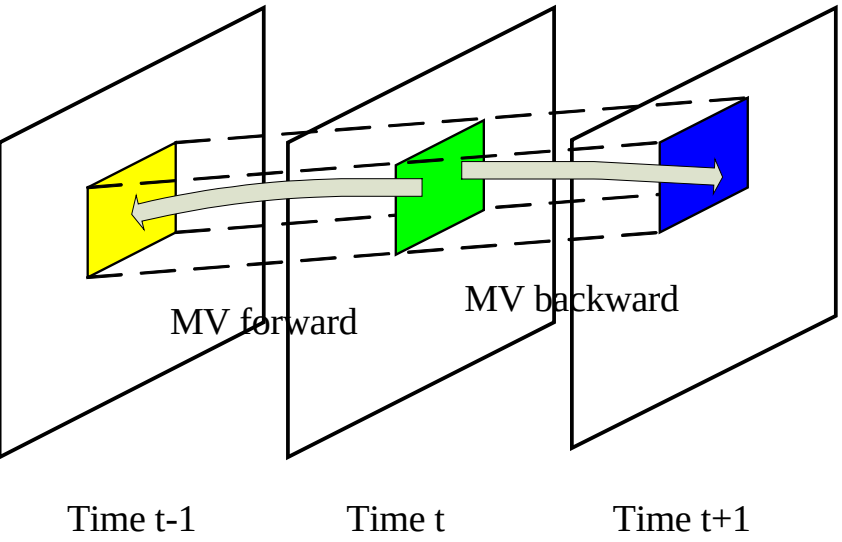
- B_Fwd_Sym_8x16
- B_Bck_Sym_8x16
- B_Sym_Fwd_8x16
- B_Sym_Bck_8x16

B_8x8 Modes



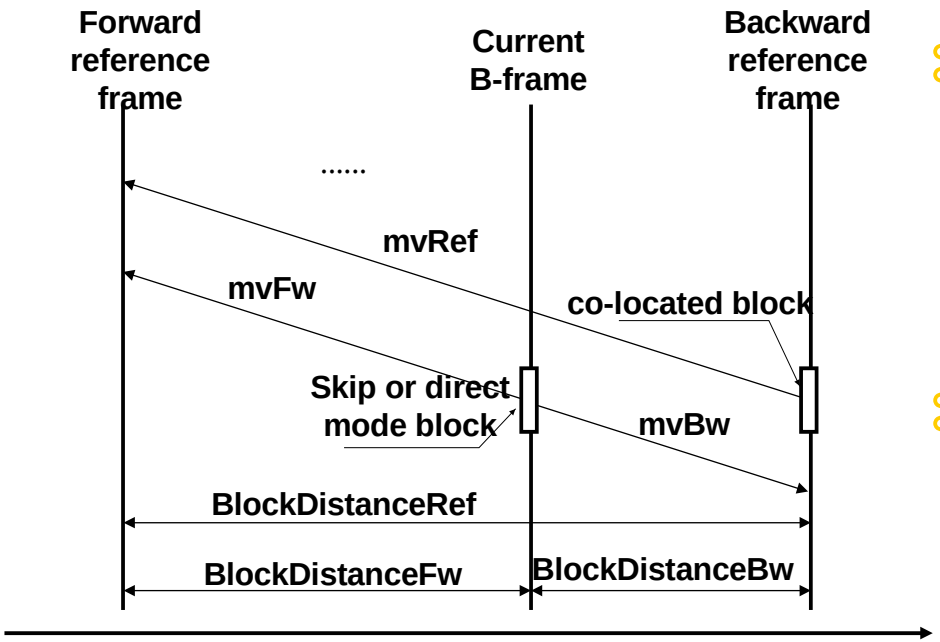
- Sub-Macroblock modes:
  - SB_Direct_8x8
  - SB_Sym_8x8
  - SB_Fwd_8x8
  - SB_Bck_8x8

Skip, Direct and Symmetry Modes in B-frame



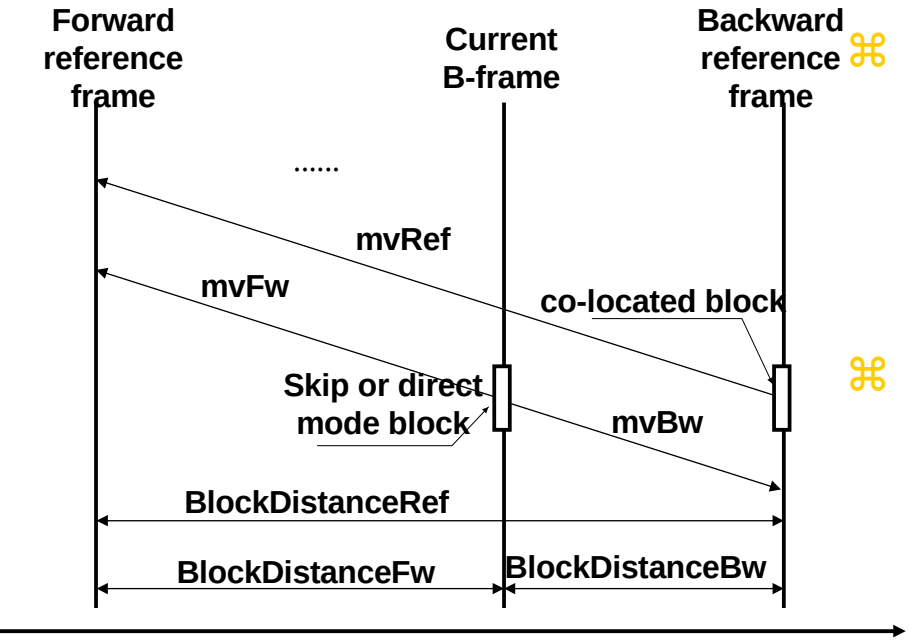
- Same:
  - Use both forward and backward reference frame to predict
- Different:
  - Skip: no coded MVD, no coded coefficient
  - Direct: no coded MVD, with coded coefficient
  - Symmetry: with coded MVD and coded coefficient

Luma Motion Vector of Bi-prediction -- Skip and Direct Modes (1)



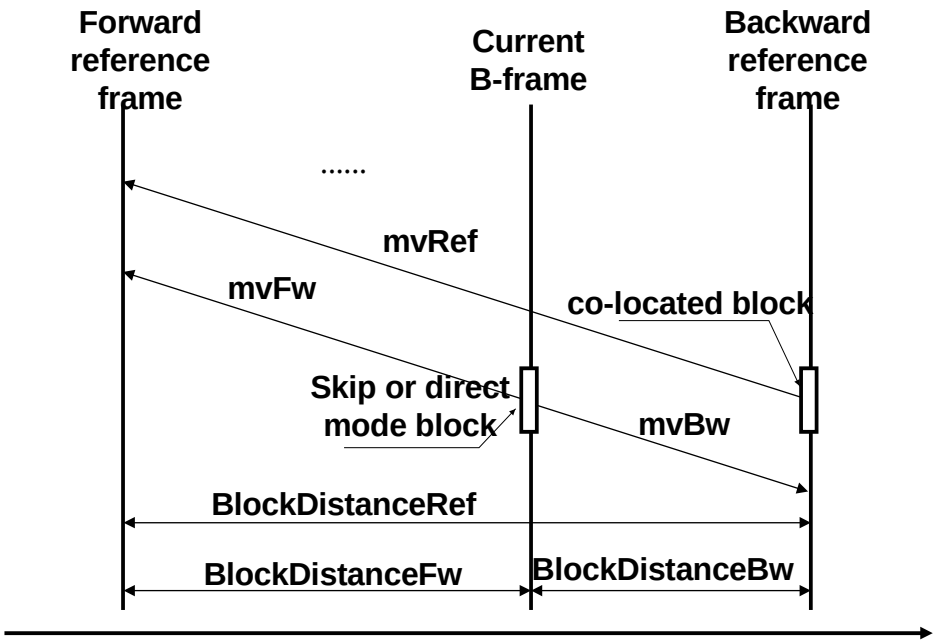
- Current picture Frame-Coded
- Co-located block in Backward reference frame
  - Reference Index = 0 (near)
  - mvRef
- mvBw and mvFw deduced from mvRef according to temporal-distance
- No MV information coded

Example



- mvFw=((16384 / BlockDistanceRef) × (1 + mvRef × BlockDistanceFw) – 1) >> 14
- mvBw=-(((16384 / BlockDistanceRef) × (1 + mvRef × BlockDistanceBw) –1) >> 14)

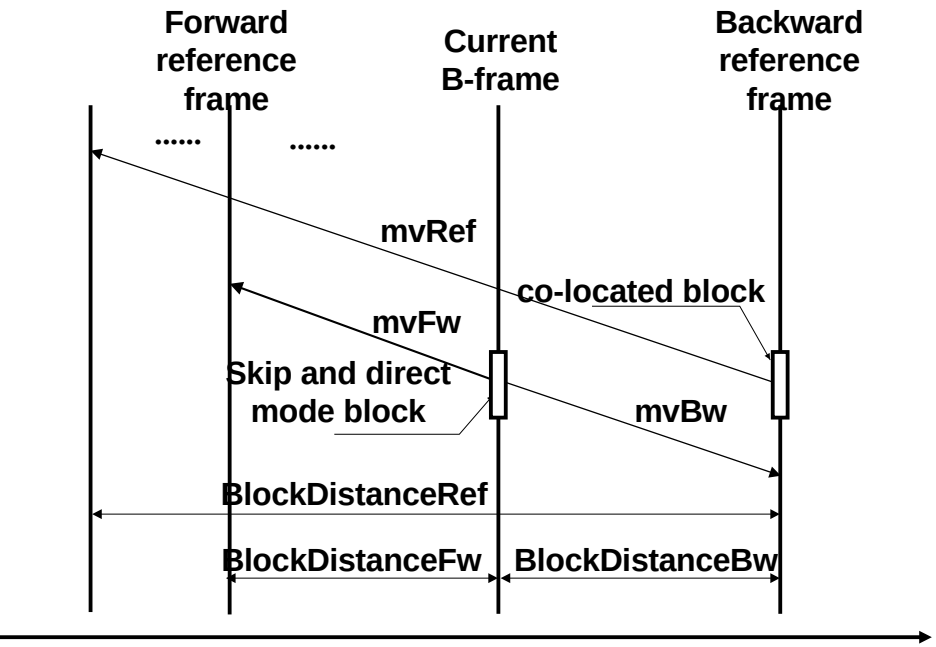
Example(continued)



$$mvFw = mvRef \times \frac{BlockDistanceFw}{BlockDistanceRef}$$

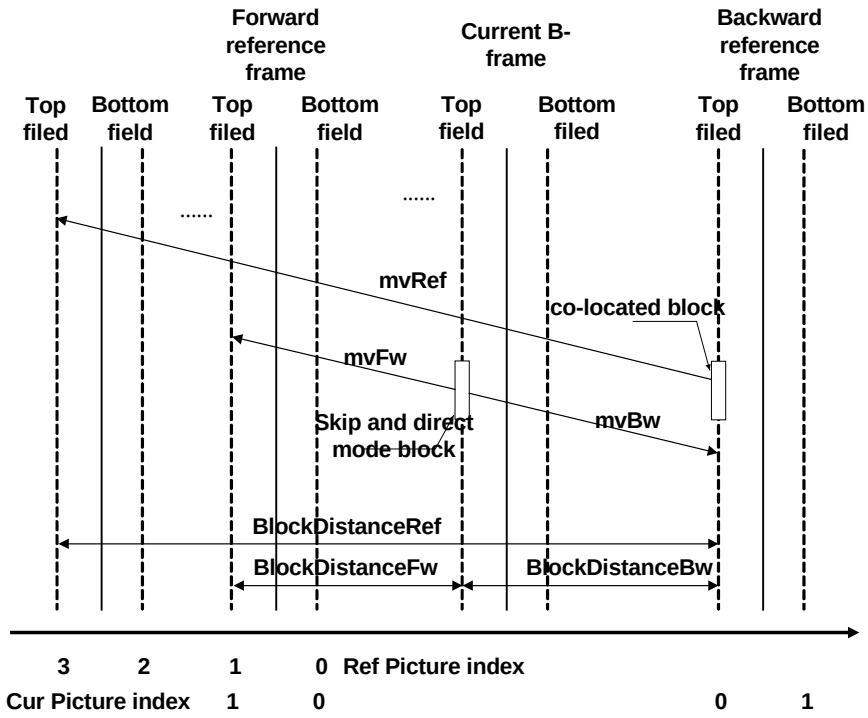
$$mvBw = mvRef \times \frac{BlockDistanceBw}{BlockDistanceRef}$$

Luma Motion Vector of Bi-prediction -- Skip and Direct Modes (2)



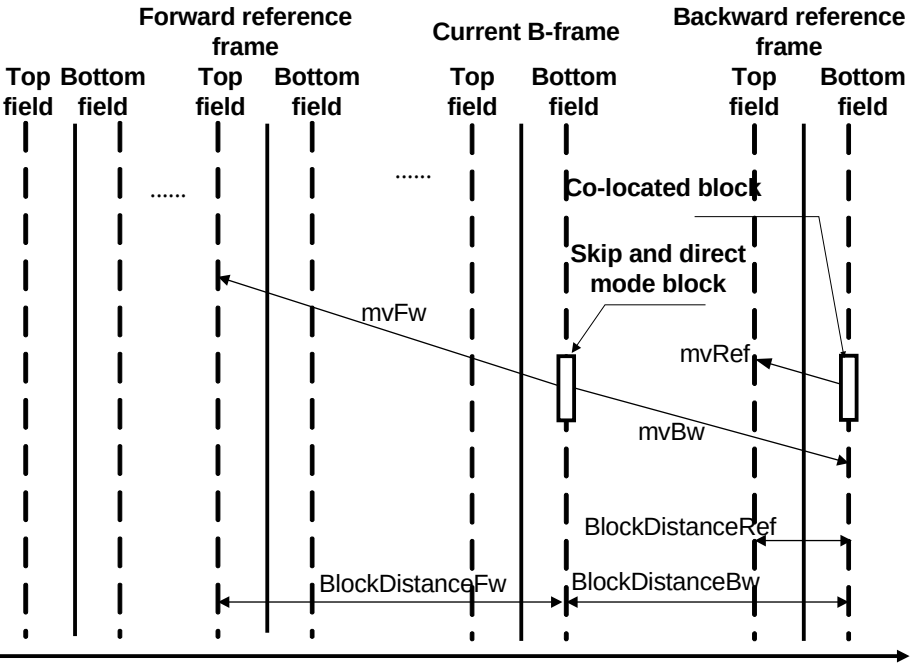
- Current picture Frame-Coded
- Co-located block in Backward reference frame
  - Reference Index =1 (far)
  - mvRef
- mvBw and mvFw deduced from mvRef according to temporal-distance
- No MV information coded

Luma Motion Vector of Bi-prediction -- Skip and Direct Modes (3)



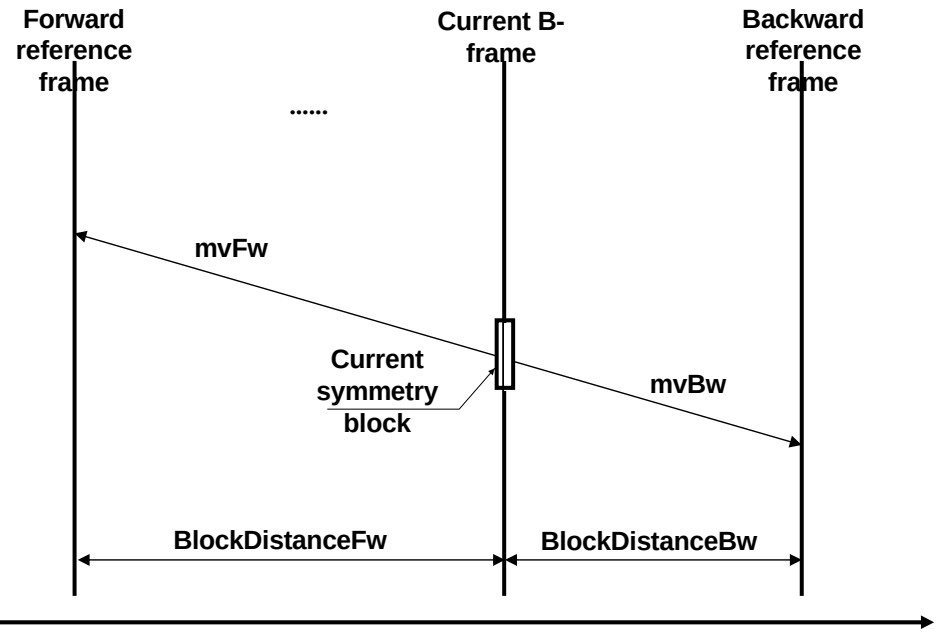
- Current picture Field-Coded
- Co-located block in Backward reference frame
  - Reference Index = 3 (far)
  - mvRef
- mvBw and mvFw deduced from mvRef according to temporal-distance
- No MV information coded
- Top field always uses top field as co-located block

Luma Motion Vector of Bi-prediction -- Skip and direct modes (4)



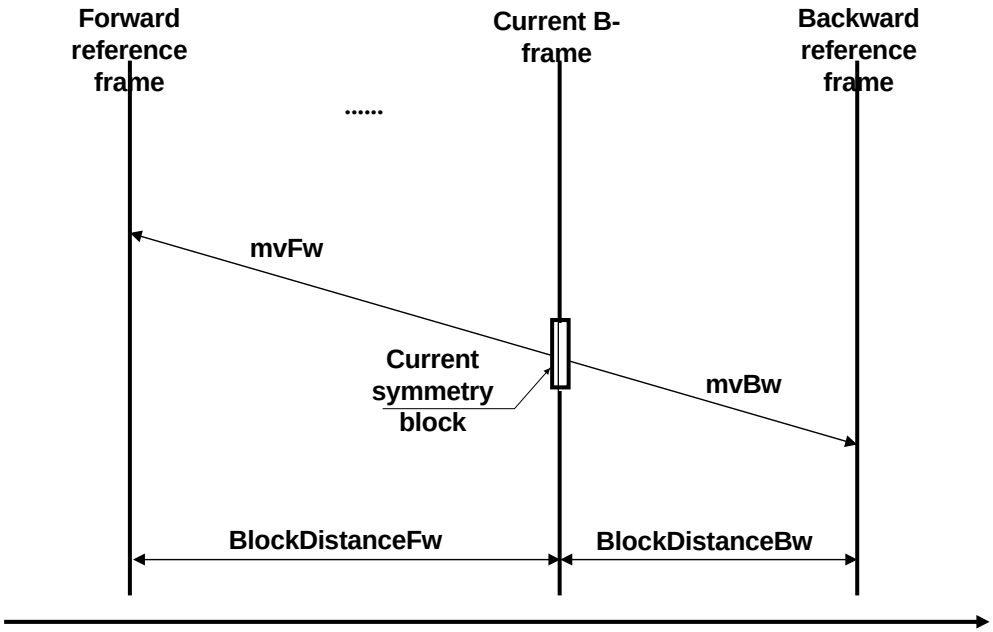
- Current picture Field-Coded
- Co-located block in Backward reference frame
  - Reference Index = 0 (far)
  - mvRef
- mvBw and mvFw deduced from mvRef according to temporal-distance
- No MV information coded
- Bottom field always uses bottom field as co-located block

Luma Motion Vector of Bi-prediction -- Symmetric (1)



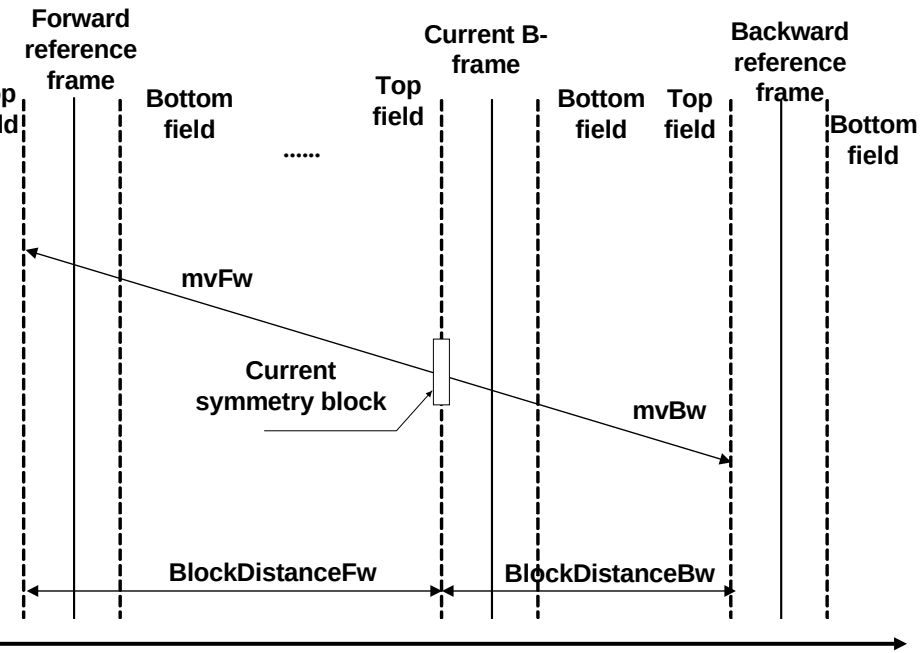
- Current picture Frame-Coded
- Forward motion vector mvFw coded
- Backward motion vector deduced from mvFw according to temporal-distance

Example



$$mvBw = -((mvFw \times BlockDistanceBw \times \frac{512}{BlockDistanceFw} + 256) >> 9)$$

Luma Motion Vector of Bi-prediction -- Symmetric (2)



- Current picture Field-Coded
- Forward motion vector mvFw coded
- Backward motion vector deduced from mvFw according to temporal-distance
- Notice:
  - Forward prediction uses top field, then backward also use top field
  - Forward prediction uses bottom field, then backward also uses bottom field

AVS B-frame Compared with Other Standards

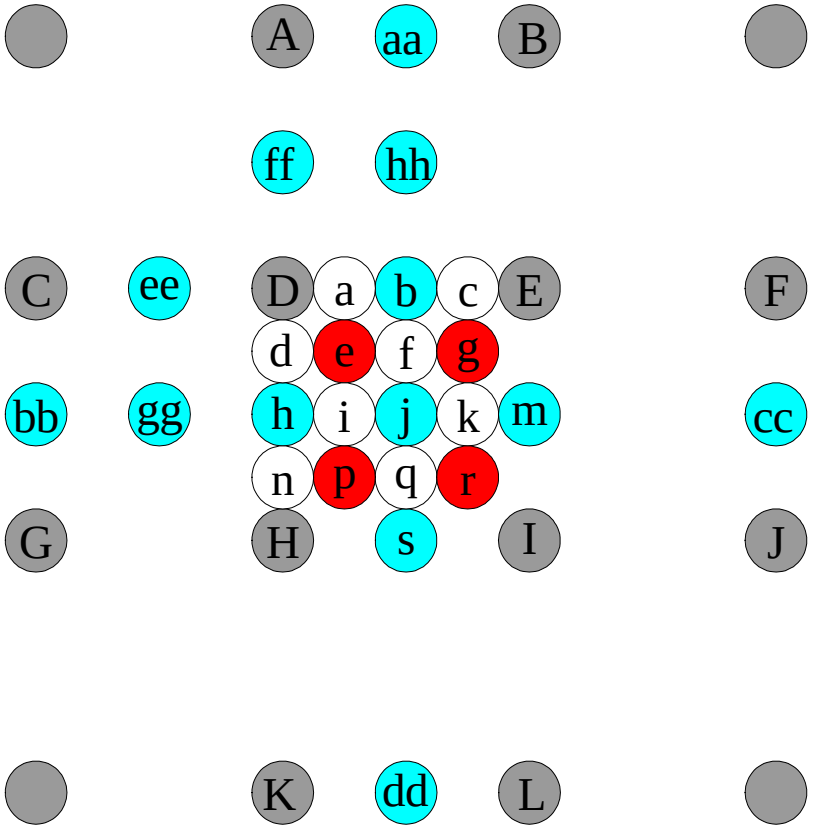
- AVS
  - Bi-prediction
    - Skip, direct
      - No motion vector
    - Symmetric
      - 1 coded motion vector
- Higher compression ratio

- Other standards
  - Bi-prediction
    - Skip, direct
      - No motion vector
    - Bi-prediction
      - 2 coded motion vectors

AVS Video 1.0 Tools

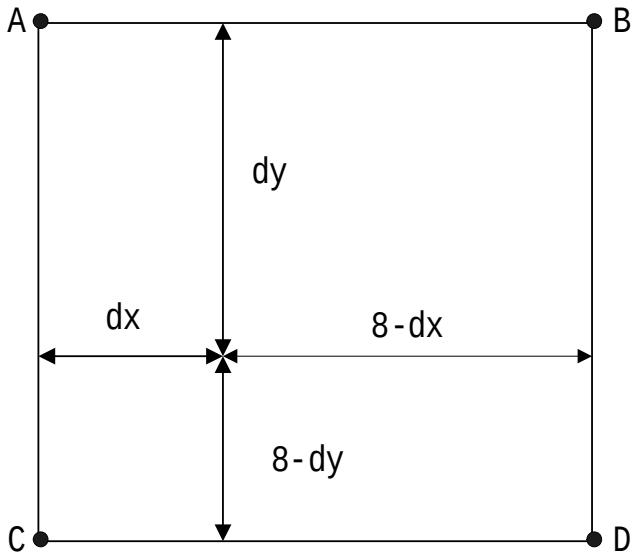
- AVS video 1.0 accepted 42 technical proposals from more than 200.
- Major tools
  - Transform – 16bit-implemented 8x8 integer transform
  - Quantization and scaling - scaling only in encoder
  - Intra prediction – 5 modes
  - Motion compensation – 16x16/16x8/8x16/8x8 modes
  - Quarter-pel interpolation – 4-taps interpolation filter
  - Deblocking
  - Entropy coding
- Minor tools
  - Motion vector prediction
  - Skipped mode and Coded block pattern
  - Alternative scan
  - Start Code simulation

Interpolation of Luma



- 4-tap filter for both ½ and ¼ pixels
- Filter matrix is
  - ½ : [-1 ,5 ,5 ,-1]/8
  - ¼ : [1 ,7 ,7 , 1]/8

Interpolation of Chroma



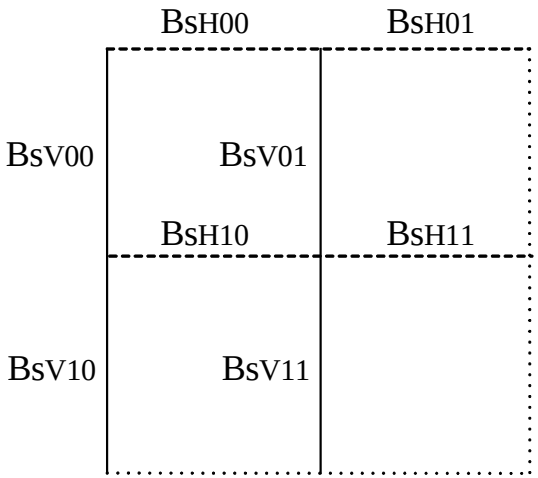
- Accuracy: 1/8 pixel
- predMatrix[x,y] =

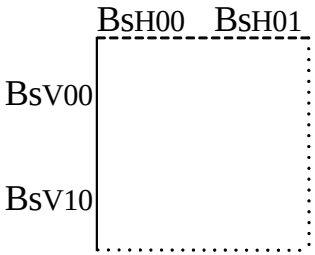
  ((8–dx)×(8–dy)×A + dx×(8–dy)×B + (8–dx)×dy×C + dx×dy×D + 32) >> 6

Interpolation Compare with H.264

- AVS
  - ½ pixels 4-tap
  - ¼ pixels 4-tap
- Advantages:
  - Data fetch bandwidth
  - complexity lower
  - Even a little gain on HD
- H.264
  - ½ pixels 6-tap
  - ¼ pixels 2-tap
- Filter matrix is:
  - ½ [1,-5,20,20,-5,1]/32
  - ¼ [1,1]/2

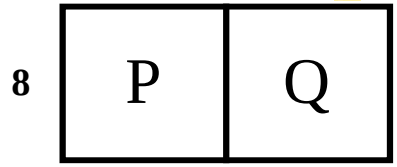
De-blocking





- 8x8-based
- Boundary strength
  - Compensation modes
  - Motion vector
- Thresholds
  - Quantization parameter
- Filtering

Boundary Strength-Decision Tree



refP: reference image of P 8x8 block

refQ: reference image of Q 8x8 block

DMVX: |MVP.x-MVQ.x|

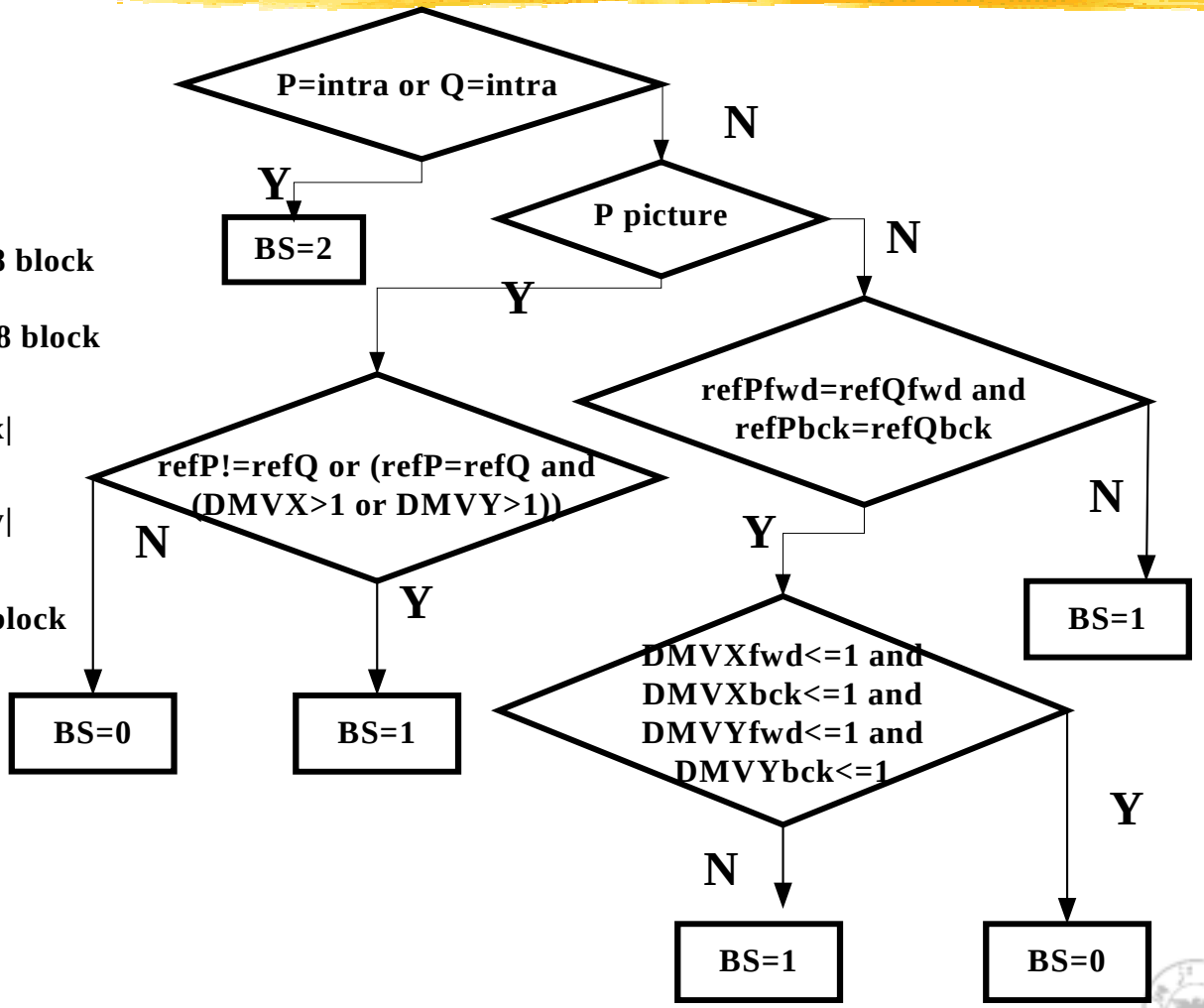
DMVY: |MVP.y-MVQ.y|

Intra: intra prediction 8x8 block

BS: boundary strength

Fwd: forward

Bck: backward



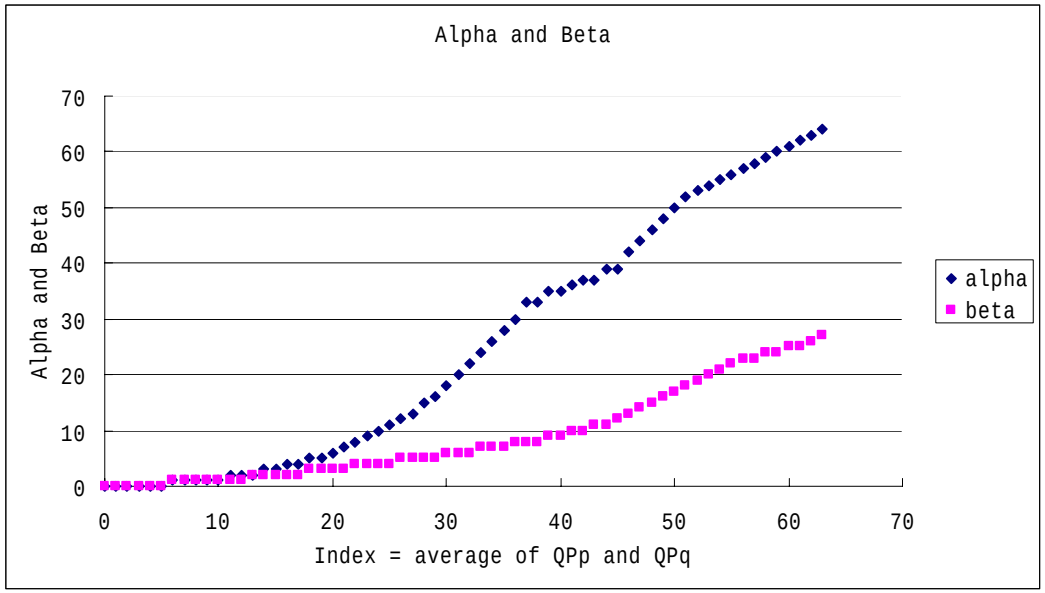
Boundary Strength-Decision Tree

- Boundary strength
  - At least one Intra block, then Bs=2;
  - Different reference frame or different motion vector, then Bs=1;
  - Otherwise (coherent), Bs=0.

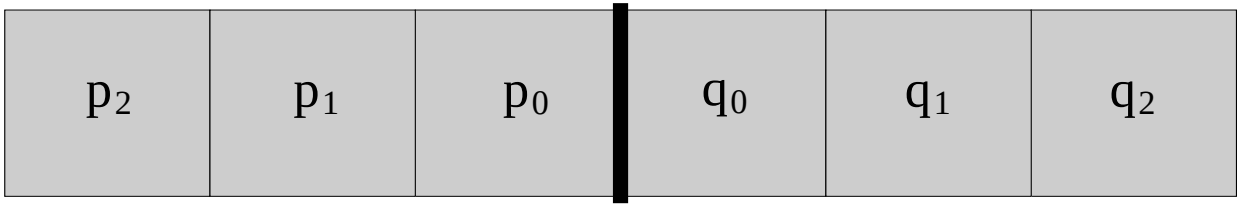
Thresholds α and β

- Calculate with QP
  - α: cross-block gradient
  - β: inner-block gradient

p_2	p_1	p_0	q_0	q_1	q_2

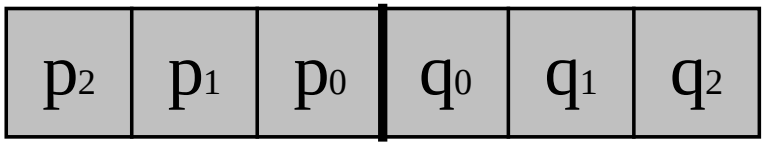


Filter Decision

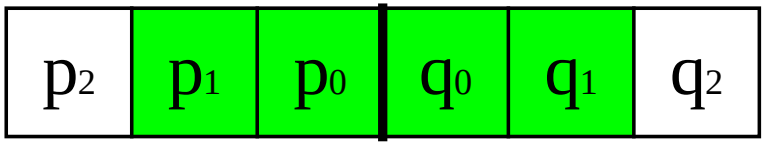


- if cross-block gradient < α, and
- inner-block gradient < β, and
- Bs != 0
  - artificial boundaries → filtering
- Otherwise
  - No artificial boundaries → no filtering

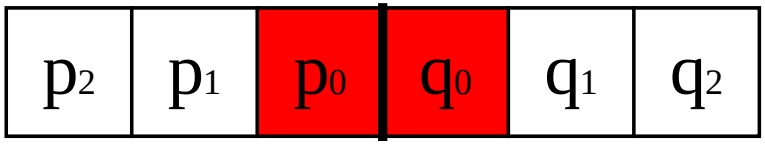
Filtering (Bs=2)



Original



Case 1

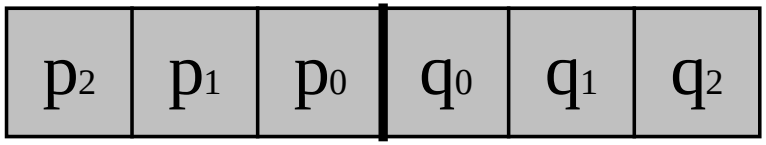


Case 2

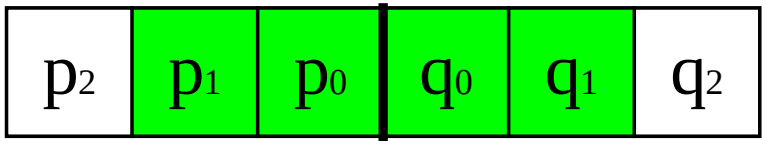
BS=2

- Case 1: (Abs(p2-p0) < β && Abs(p0–q0) < ((α >> 2) + 2))
  - 2 pixels filtered
    - P0: [1, 2, 1] /4
      [p1, p0, q0]
    - P1: [2, 1, 1]/4
      [p1, p0, q0]
- Case 2:
  - 1 pixels filtered
    - P0: [2, 1, 1]/4
      [p1, p0, q0]
- Chroma: only P0, Q0 filtered

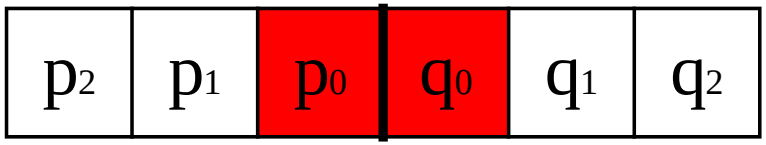
Filtering (Bs=1)



Original



Case 1



Case 2

BS=1

- P0, Q0:
  - Δ =[1, -3, 3, -1]/8
    [p1, p0, q0, q1]
  - P0 = p0 + Δ
  - Q0 = q0 + Δ

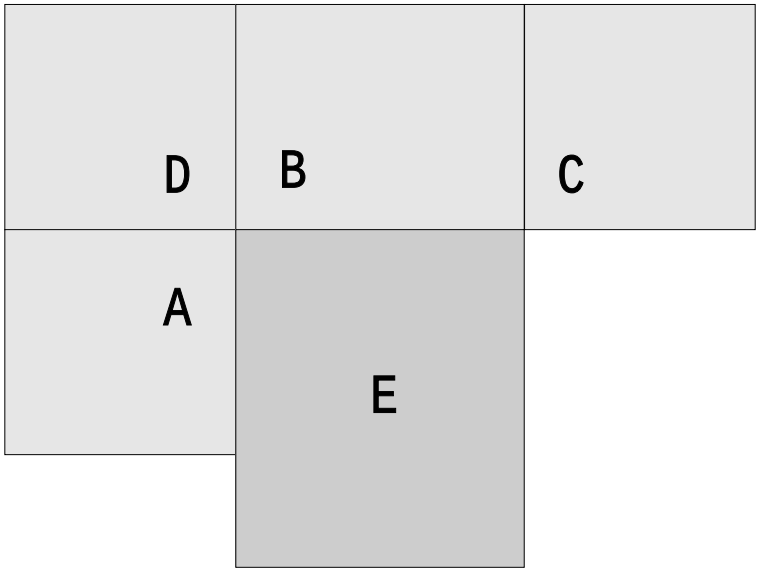
- P1, Q1:

  Only filtered when (Luma && Abs(p2-p0)< β))
  - Δ_p=[1, -3, 3, -1]/8
    [p2, p1, P0, Q0]
  - P1 = p1 + Δ_p

AVS Deblocking Compare with H.264

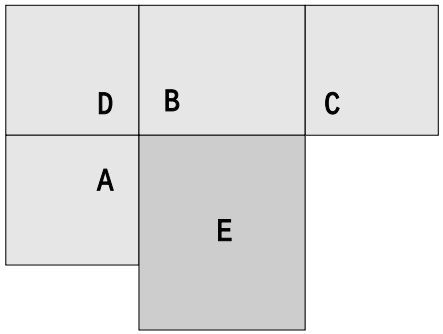
- AVS
  - 8x8 base
    - Less boundaries
  - Less BS-levels (0..2), simpler decision tree
  - Less pixels filtered (p0, p1,q0, q1)
    - Parallel
- H.264
  - 4x4 base
    - More boundaries
  - More BS-levels (0..4), more complex decision tree
  - More pixels filtered (p0..p3,q0..q3)
- Lower complexity and higher parallel

Motion Vector Prediction

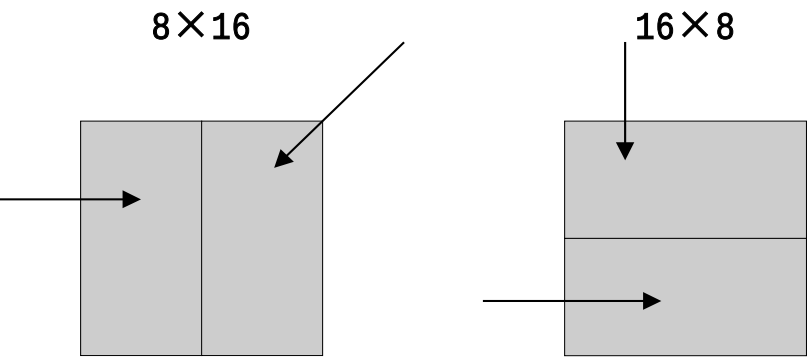


- Predict motion vector of E from A, B, C, D's motion vectors

Motion Vector Prediction (1)



- 8x16
- Left block : PreMVE=MVA
- Right block: PreMVE=MVC



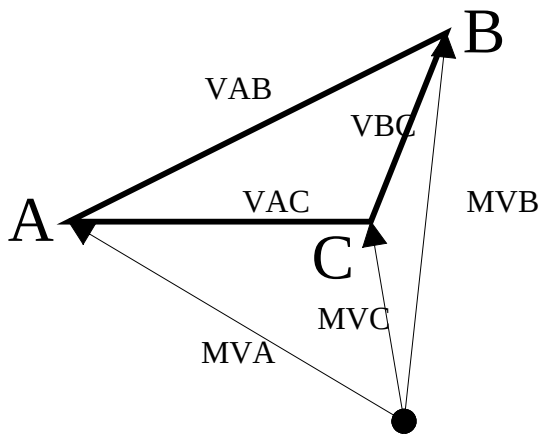
- 16x8
- Top block: PreMVE=MVB
- Bottom block: PreMVE=MVA

- Temporal distance scaling:
  - MVA_x = Sign(mvA_x) × (Abs(mvA_x) × BlockDistanceE × (512 / BlockDistanceA) + 256) >> 9

Motion Vector Prediction (2)

- PreMV$_E$:geometrical median of MV$_A$, MV$_B$, MV$_C$
  - V$_{AB}$ = Dist(MV$_A$, MV$_B$)
  - V$_{BC}$ = Dist(MV$_B$, MV$_C$)
  - V$_{CA}$ = Dist(MV$_C$, MV$_A$)
  - where,

    Dist(MV$_1$, MV$_2$) = Abs(x1 – x2) + Abs(y1 – y2)
  - FMV = Median(V$_{AB}$, V$_{BC}$, V$_{CA}$)



- If FMV=V$_{AB}$
  - PreMV$_E$ = MV$_C$
- If FMV=V$_{BC}$
  - PreMVE = MVA
- If FMV=V$_{CA}$
  - PreMV$_E$ = MV$_B$

AVS CBP Coding Compared to H.264

- AVS
  - Luma: 1 bit in CBP presence of one 8x8-block
  - Chroma: 1 bit in CBP presence of U 8x8- block and 1 bit V 8x8- block
  - Range: 0..63
  - Exp-Golomb coded

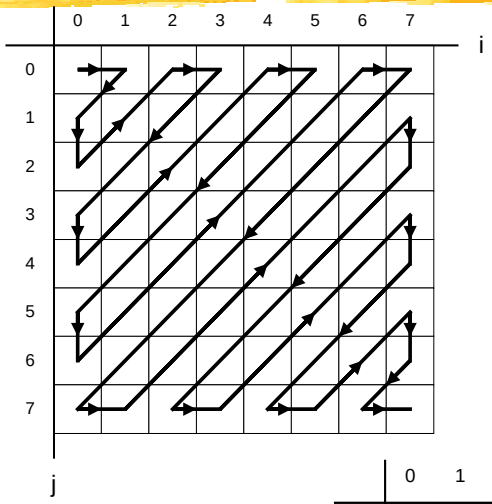
0	1
2	3

- H.264
  - Luma: 1 bit in CBP presence of one 8x8-block, i.e. four 4x4 blocks
  - Chroma: 2 bit in CBP
    - 1 bit for DC
    - 1 bit for AC
  - Range: 0..47
  - Exp-Golomb coded

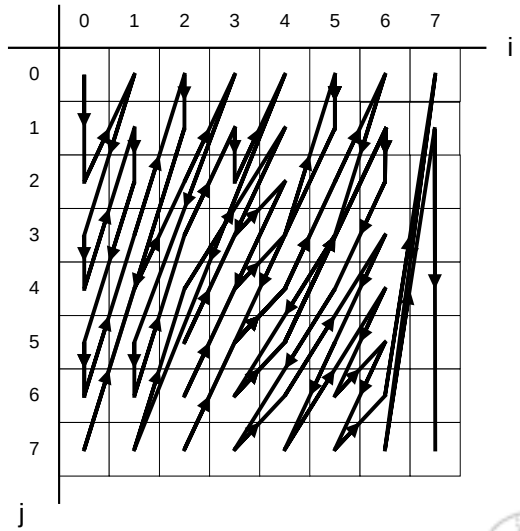
0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

Transform Coefficients Adaptive Scan

- Zigzag-scan (frame-coding)
  - Progressive
  - Interlace picture frame-coding



- Alternative-scan
  - Interlace picture field-coding

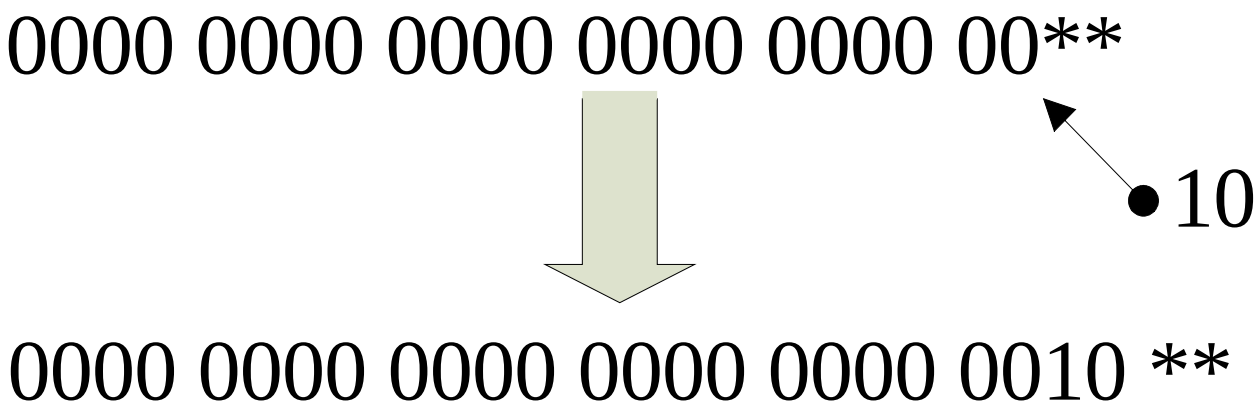


Start-Code Emulation

- Start-code of AVS:
  - Prefix '0000 0000 0000 0000 0000 0001'
- Start-code Emulation:
  - Some bits string output from entropy coding might same as start-code-prefix
  - make decoder confused

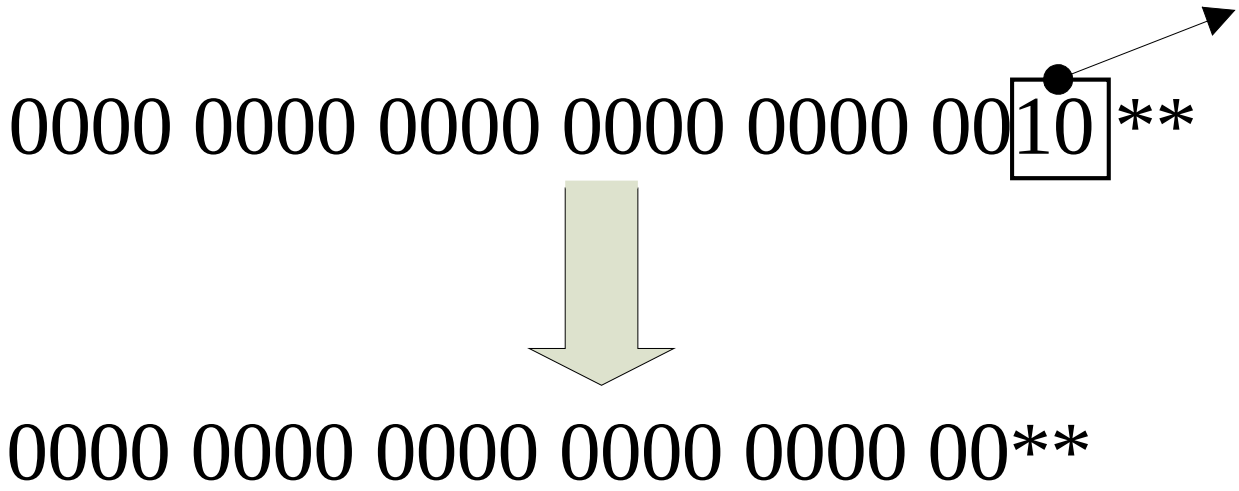
Prevent Start-Code Emulation

- Encoder :
  - Always insert '10' after 22 '0's
  - Except: sequence_header、sequence_display_extension、copyright_extension、user_data、camera_parameters_extension



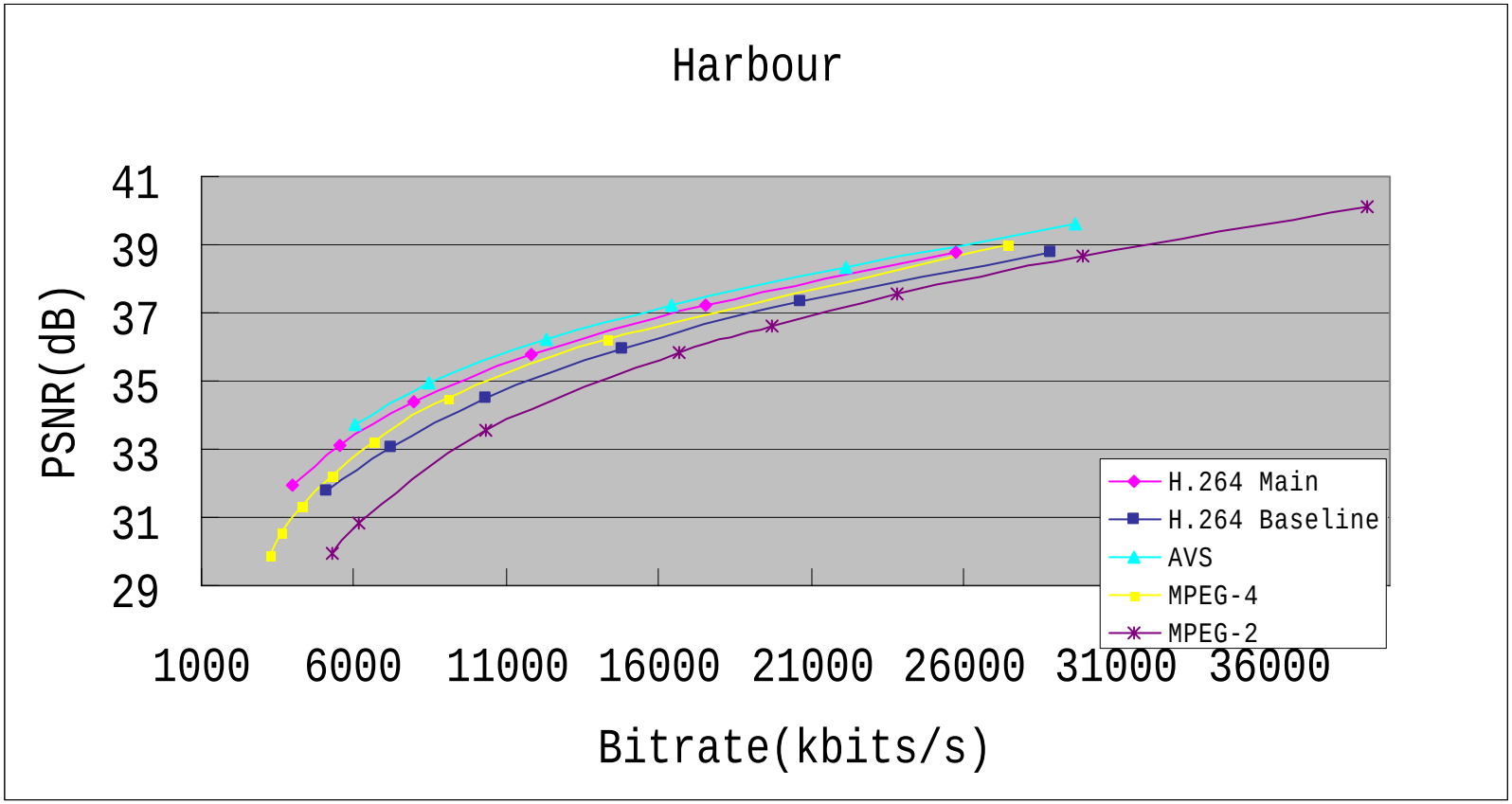
Prevent Start-Code Emulation (Cont.)

- Decoder:
  - Discard '10' after 22 '0's
  - Except: sequence_header、sequence_display_extension、copyright_extension、user_data、camera_parameters_extension

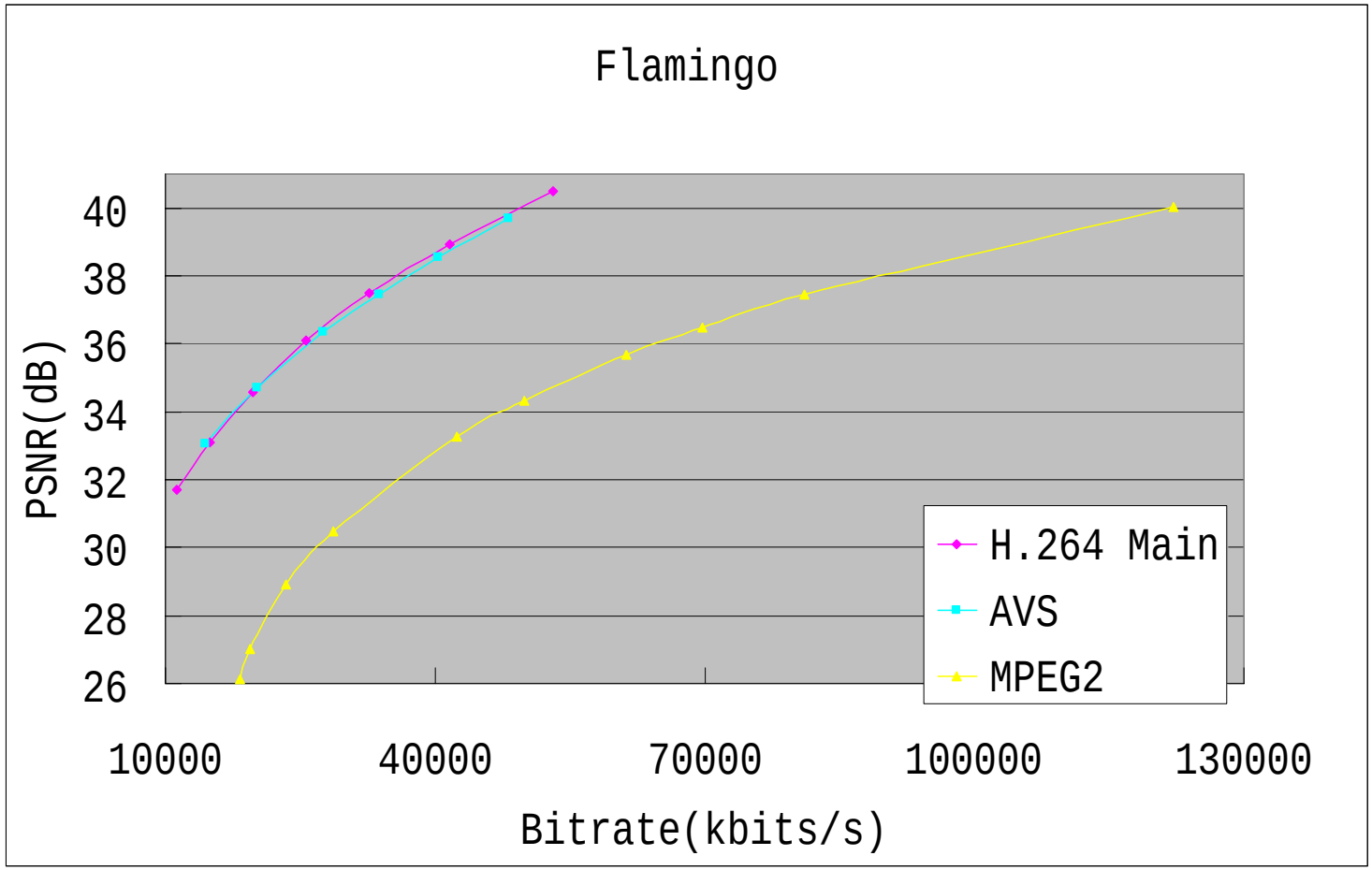


Performance

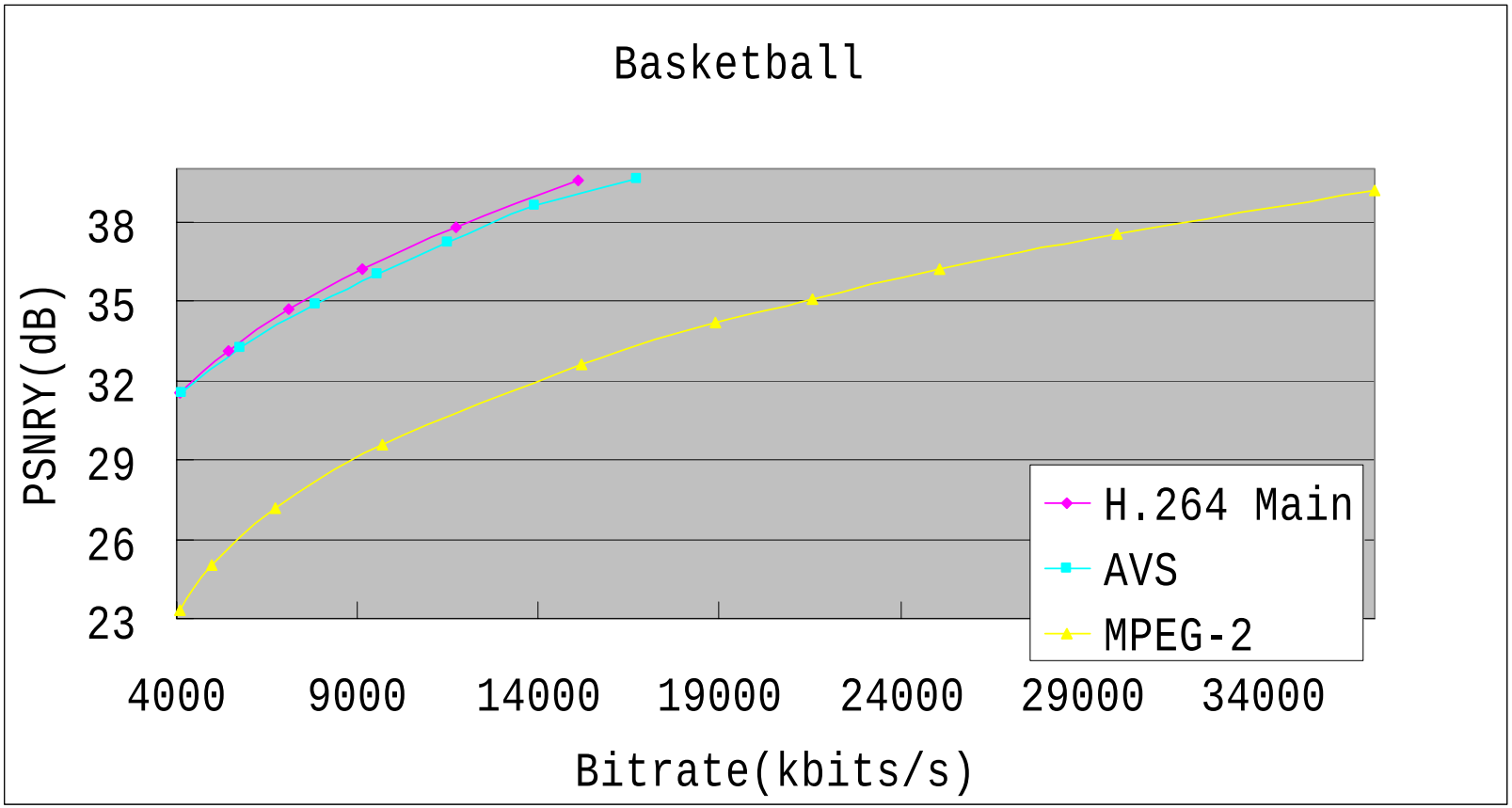
Performance- HD (1280 x720)



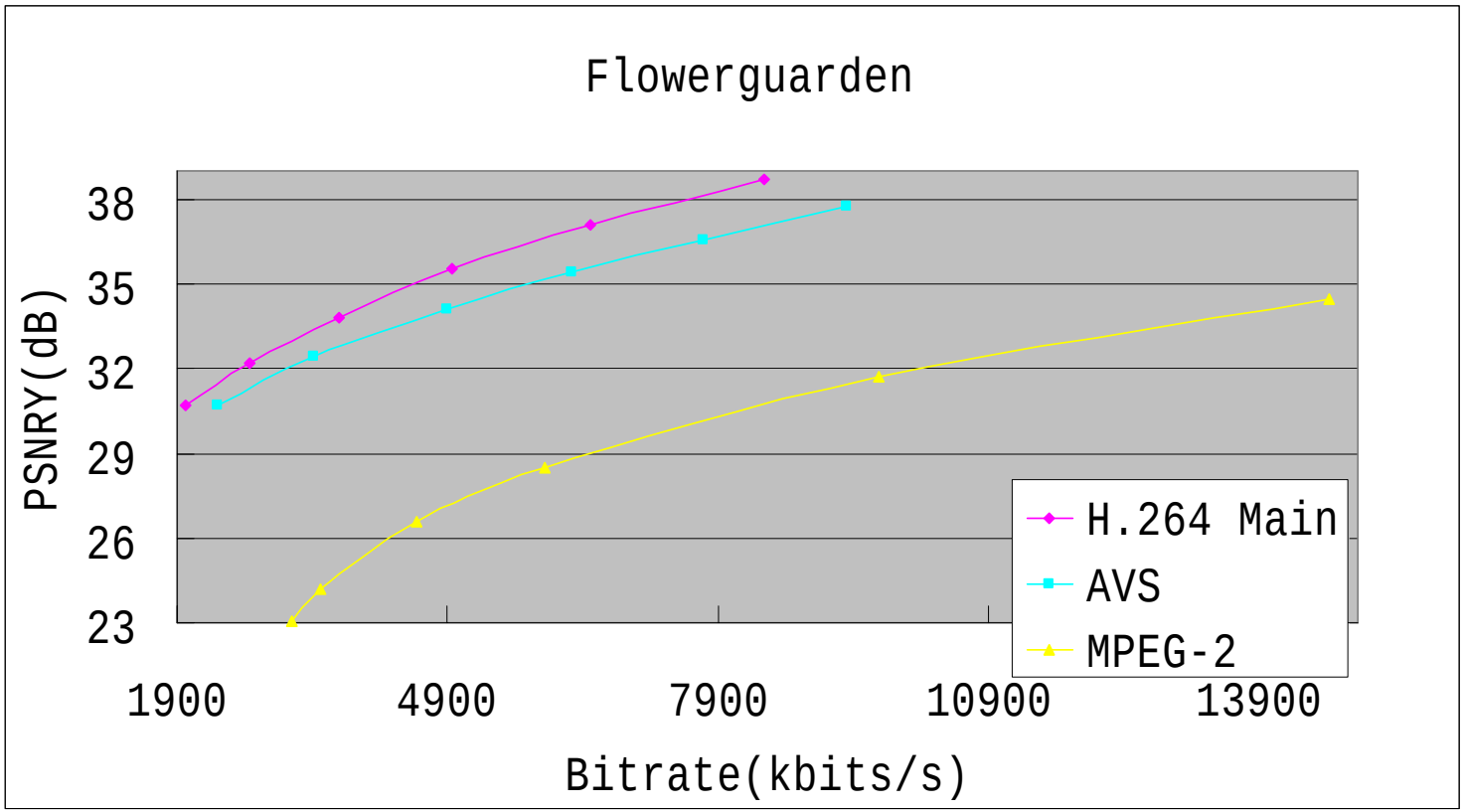
Performance- HD (1280 x720)



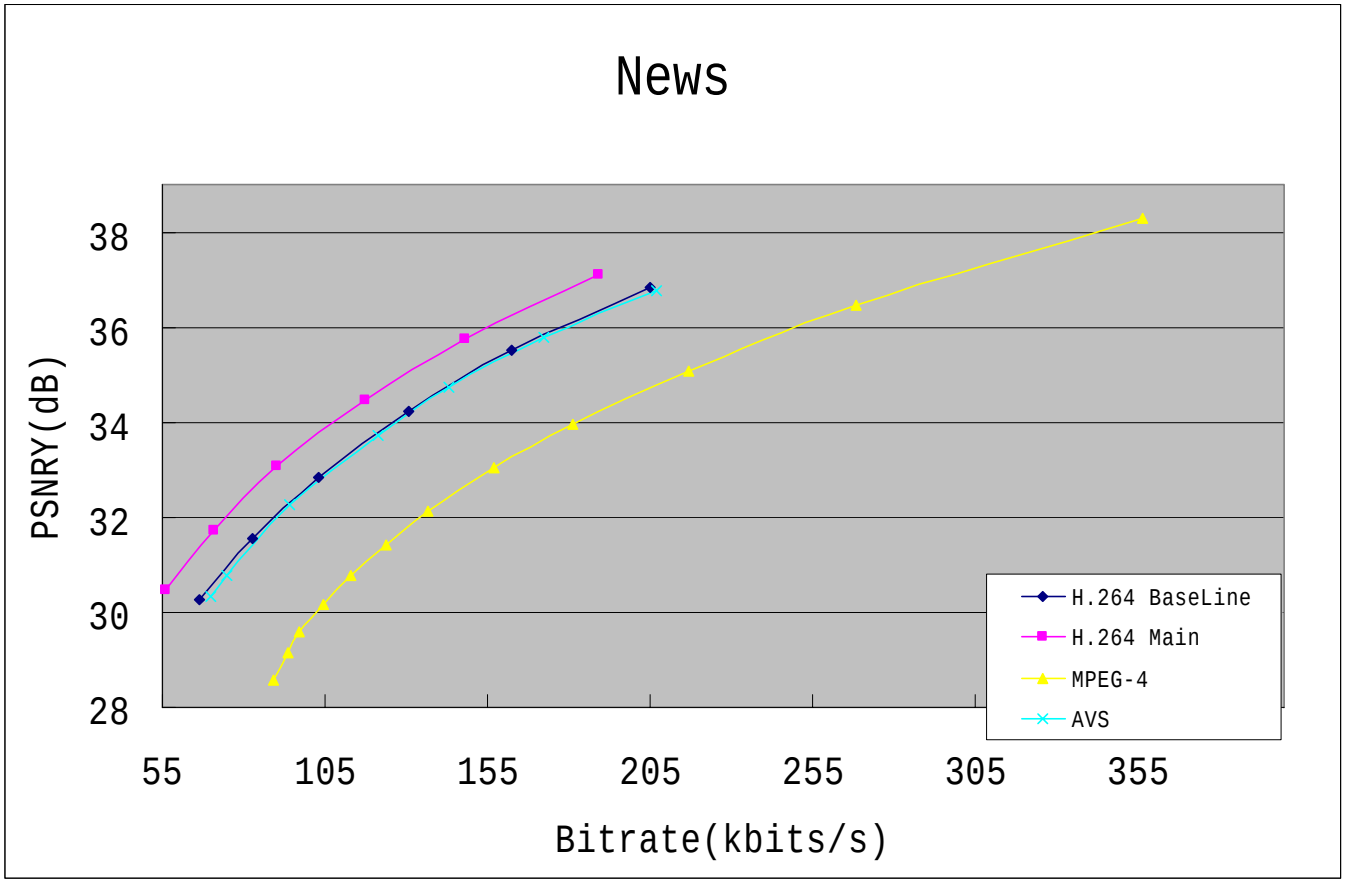
Performance – SD (720x576)



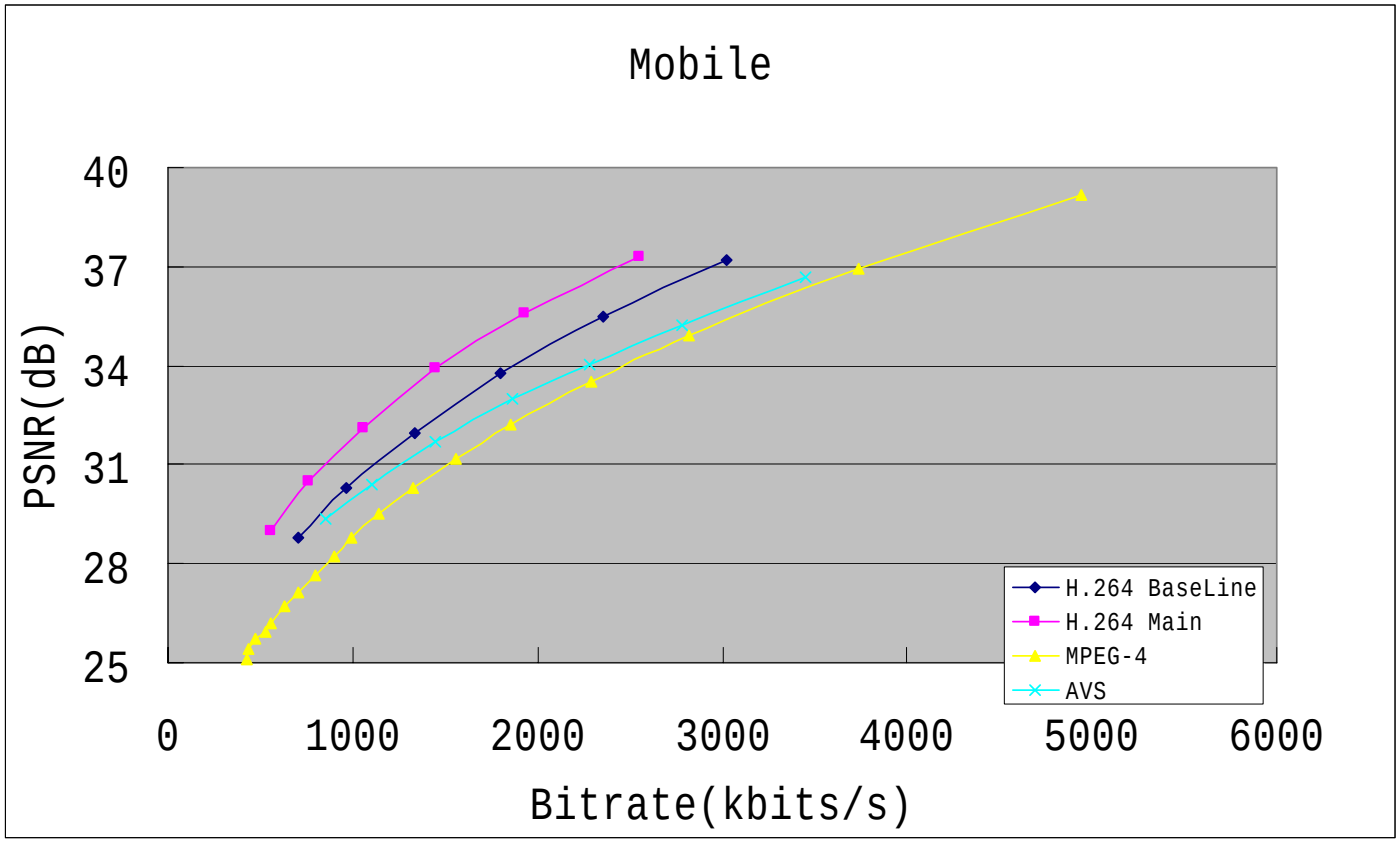
Performance – SD (720x576)



Performance – CIF (352x288)



Performance – CIF (352x288)

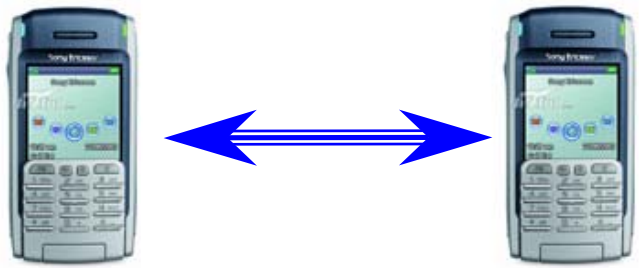


Next step?

AVS-M

- Stage 1
  - Dec. 2004
  - support CIF and below
  - 384kbps and below
  - Only simple error resilience tools
  - Suitable for local playback with non-realtime transmission
  - Compression performance and complexity of decoder

- Stage 2
  - Apr. 2006
  - Bi-direction communication
  - Low latency
  - Error resilience tools
  - Compression performance and complexity of both encoder and decoder



X Profile

- Requirements
  - Digital television broadcasting
    - Terrestrial
    - Cable
    - satellite
  - Storage media
    - Optical
    - hard disk
    - solid state memory (e.g. flash, SD card, etc.)
  - Video surveillance
  - Video on demand
  - Video conference

Conclusion

- Advantages of AVS
  - Based on self-owned Intellectual Property
    - Lower loyalty fee (1.0 ￥)
  - Advanced compression efficiency
    - Similar performance with H.264
    - 2 times to MPEG-2
  - Lower complexity
    - Computation
    - memory
    - Bandwidth
  - Well-defined applications
    - Simplify and clear
- Chances
  - 300-500 million Chips and end-user products
  - Early join, early benefit from

Thanks for your interest in

AVS !