H.264/AVC Video Coding Standard

- **Standardization, History, Goals, and Applications**
- **Codec Overview**
- **Video Coding Layer (VCL)**
  - Picture Partitioning and Interlace Processing
  - Codec Structure
  - Motion-Compensated Prediction
  - Intra Prediction
  - Prediction Residual Coding
  - Deblocking Filter
  - Encoder Test Model
- **Performance**
- **Network Abstraction Layer (NAL)**
  - NAL Units and Types
  - RTP Carriage and Byte Stream Format

The JVT Project

- ITU-T SG16 H.26P and H.26L plans in 1993 (H.26P became H.263)
- ITU-T Q.6/SG16 (**VCEG - Video Coding Experts Group**) formed for ITU-T standardization activity for video compression since 1997
- **August 1999**: 1st test model (TML-1) of H.26L
- **December 2001**: Formation of the **Joint Video Team (JVT)** between **VCEG** and ISO/IEC JTC 1/SC 29/WG 11 (**MPEG - Moving Pictures Experts Group)** to establish a joint standard project - **H.264 / MPEG4-AVC** (similar to H.262 / MPEG-2 Video);
- **JVT Chairs**: G. J. Sullivan, A. Luthra, and T. Wiegand
- **ITU-T Approval**: **May 2003** – ITU-T SG16 Final Standard Approved
- **ISO/IEC Approval**: **March 2003** - Final Draft International Standard – currently balloting
- **Extensions Project: Professional Extensions until April 2004**

Goals

- **Improved Coding Efficiency**
  - Average bit rate reduction of 50% given fixed fidelity compared to any other standard
  - Complexity vs. coding efficiency scalability
- **Improved Network Friendliness**
  - Issues examined in H.263 and MPEG-4 are further improved
  - Anticipate error-prone transport over mobile networks and the wired and wireless Internet
- **Simple syntax specification**
  - Targeting simple and clean solutions
  - Avoiding any excessive quantity of optional features or profile configurations

Applications

- **Entertainment Video (1-8+ Mbps, higher latency)**
  - Broadcast / Satellite / Cable / DVD / VoD / FS-VDSL / …
  - DVB/ATSC/SCTE, DVD Forum, DSL Forum
- **Conversational Services (usu. <1Mbps, low latency)**
  - H.320 Conversational
  - 3GPP Conversational H.324/M

  circuit-switched

  - H.323 Conversational Internet/best effort IP/RTP
  - 3GPP Conversational IP/RTP/SIP

  packet-switched
- **Streaming Services (usu. lower bit rate, higher latency)**
  - 3GPP Streaming IP/RTP/RTSP
  - Streaming IP/RTP/RTSP (without TCP fallback)
- **Other Services**
  - 3GPP Multimedia Messaging Services

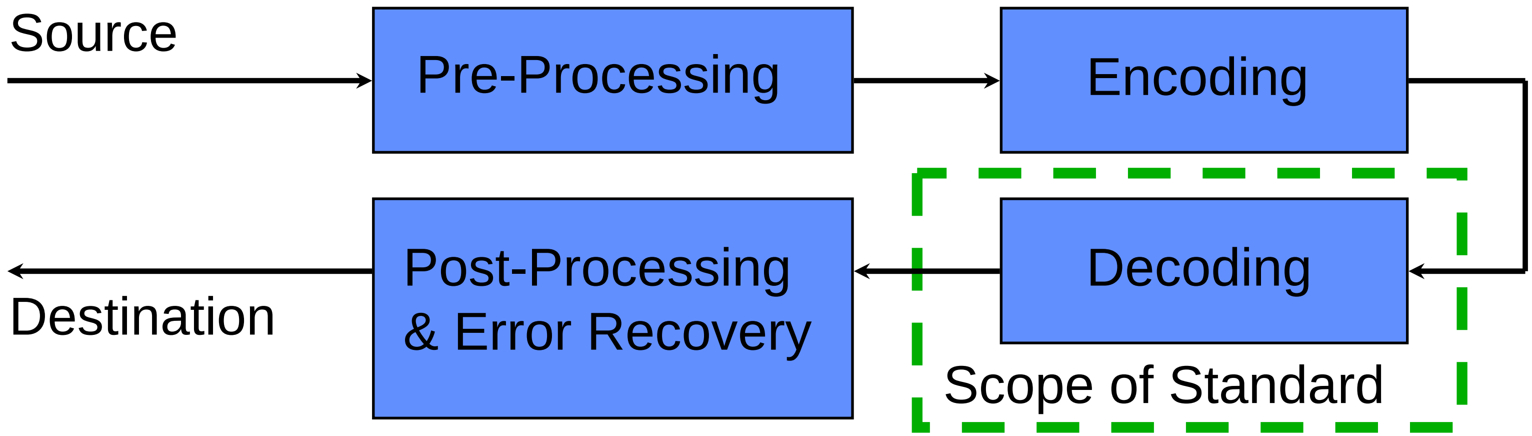
Relationship to Other Standards

- **Identical specifications have been approved in both ITU-T / VCEG and ISO/IEC / MPEG**
- **In ITU-T / VCEG this is a new & separate standard**
  - ITU-T Recommendation H.264
  - ITU-T Systems (H.32x) will be modified to support it
- **In ISO/IEC / MPEG this is a new “part” in the MPEG-4 suite**
  - Separate codec design from prior MPEG-4 visual
  - New part 10 called “Advanced Video Coding” (AVC – similar to “AAC” position in MPEG-2 as separate codec)
  - MPEG-4 Systems / File Format has been modified to support it
  - **H.222.0 | MPEG-2 Systems also modified to support it**
- **IETF finalizing RTP payload packetization**

The *Scope* of Picture and Video Coding Standardization

Only Restrictions on the *Bitstream*, *Syntax,* and *Decoder* are standardized:

- Permits optimization beyond the obvious
- Permits complexity reduction for implementability
- Provides *no* guarantees of quality

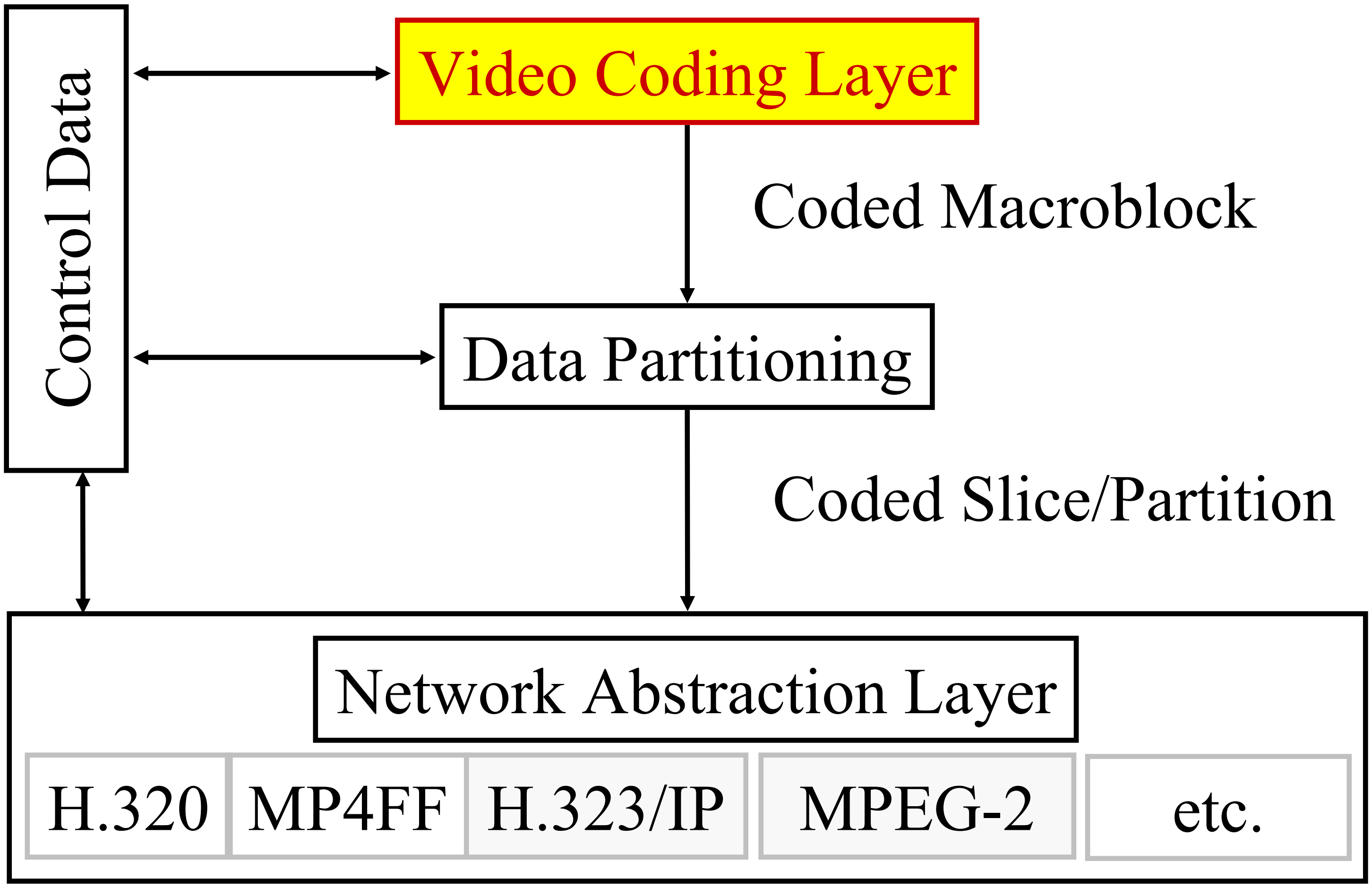


Profiles & Levels Concepts

- **Many standards contain different configurations of capabilities – often based in “profiles” & “levels”**
  - A profile is usually a set of algorithmic features
  - A level is usually a degree of capability (e.g. resolution or speed of decoding)

- **H.264/AVC has three profiles**
  - Baseline (lower capability plus error resilience, e.g., videoconferencing, mobile video)
  - Main (high compression quality, e.g., broadcast)
  - Extended (added features for efficient streaming)

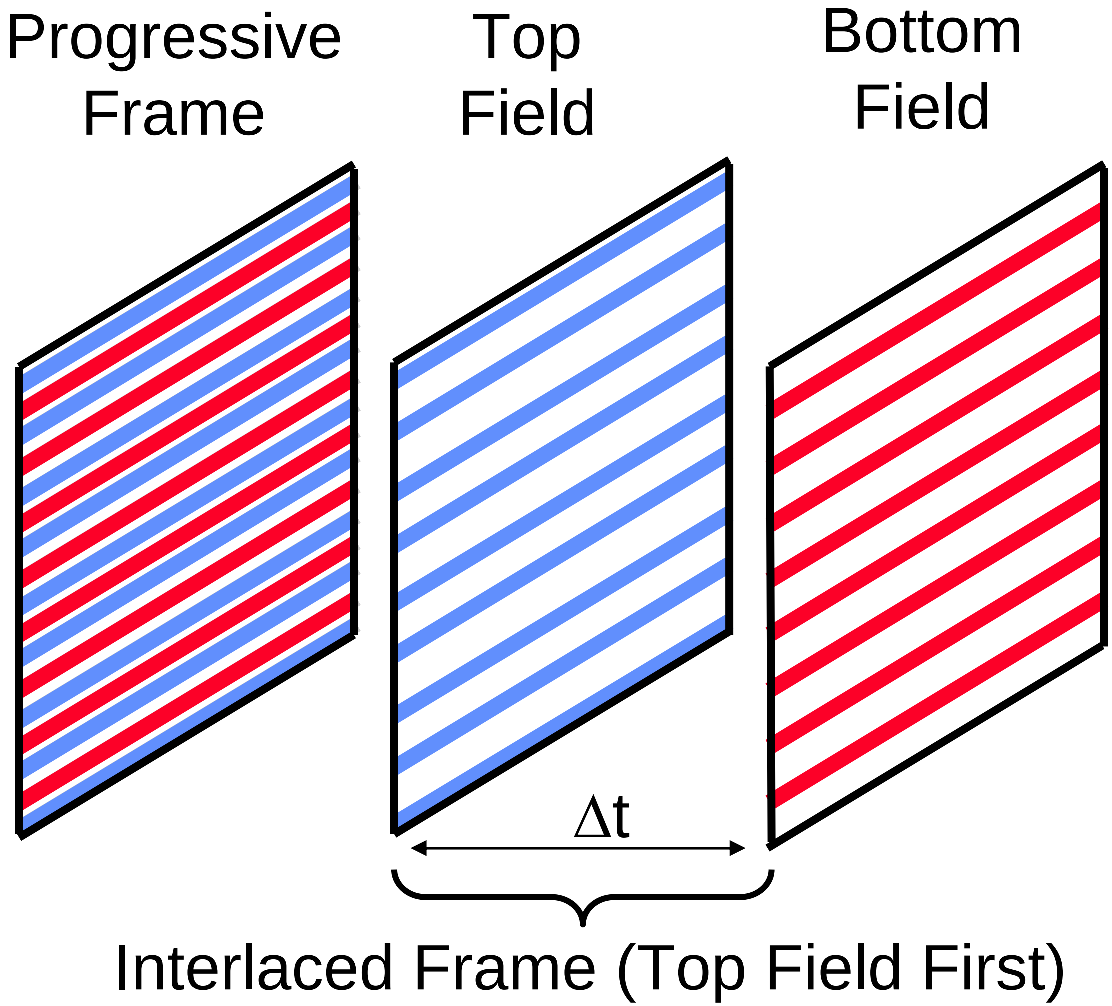
H.264|AVC Layer Structure



High-Level VCL Summary

- Video coding layer is based on hybrid video coding and similar in spirit to other standards but with important differences
- Some new key aspects are:
  - Enhanced motion compensation
  - Small blocks for transform coding
  - Improved de-blocking filter
  - Enhanced entropy coding
- Substantial bit-rate savings relative to other standards for the same quality

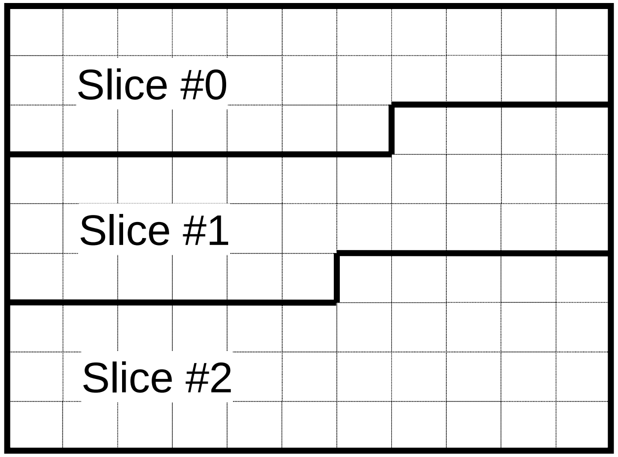
Input Video Signal

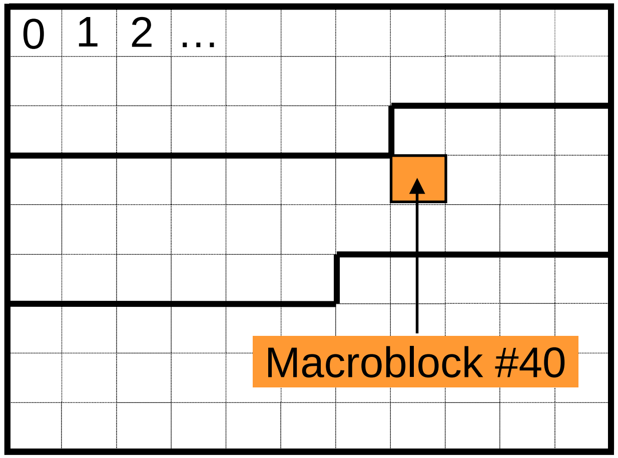


- Progressive and interlaced frames can be coded as one unit
- Progressive vs. interlace frame is signaled but has no impact on decoding
- Each field can be coded separately
- Dangling fields

Partitioning of the Picture

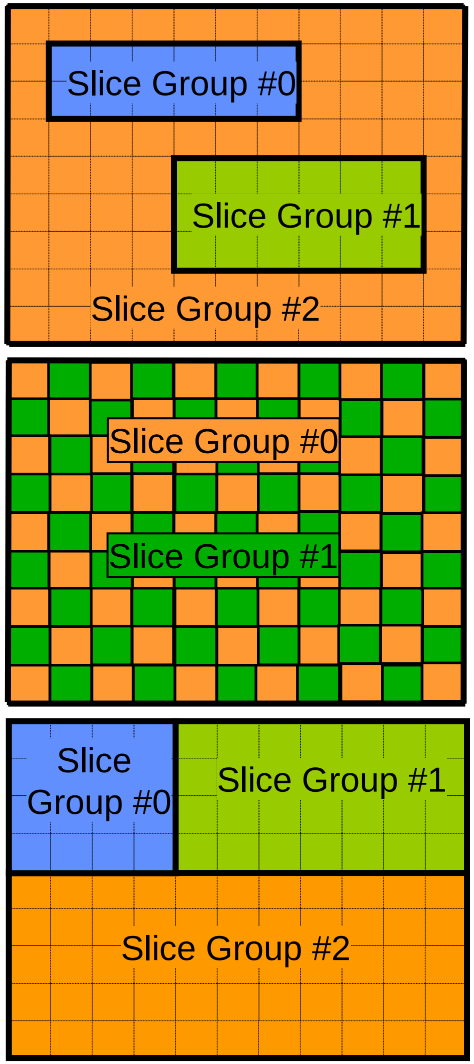
- **Slices**:
  - A picture is split into 1 or several slices
  - Slices are self-contained
  - Slices are a sequence of macroblocks
- **Macroblocks**:
  - Basic syntax & processing unit
  - Contains 16x16 luma samples and 2 x 8x8 chroma samples
  - Macroblocks within a slice depend on each other
  - Macroblocks can be further partitioned





Flexible Macroblock Ordering (FMO)

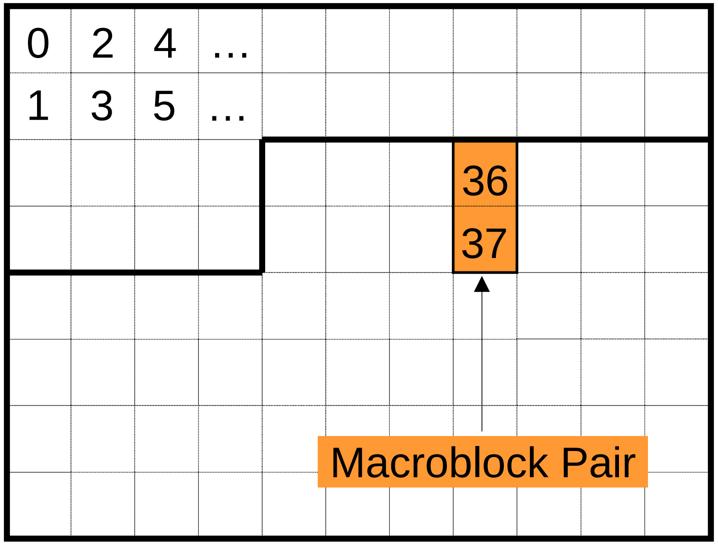
- **Slice Group**:
  - Pattern of macroblocks defined by a Macroblock allocation map
  - A slice group may contain 1 to several slices
- **Macroblock allocation map types**:
  - Interleaved slices
  - Dispersed macroblock allocation
  - Explicitly assign a slice group to each macroblock location in raster scan order
  - One or more “foreground” slice groups and a “leftover” slice group



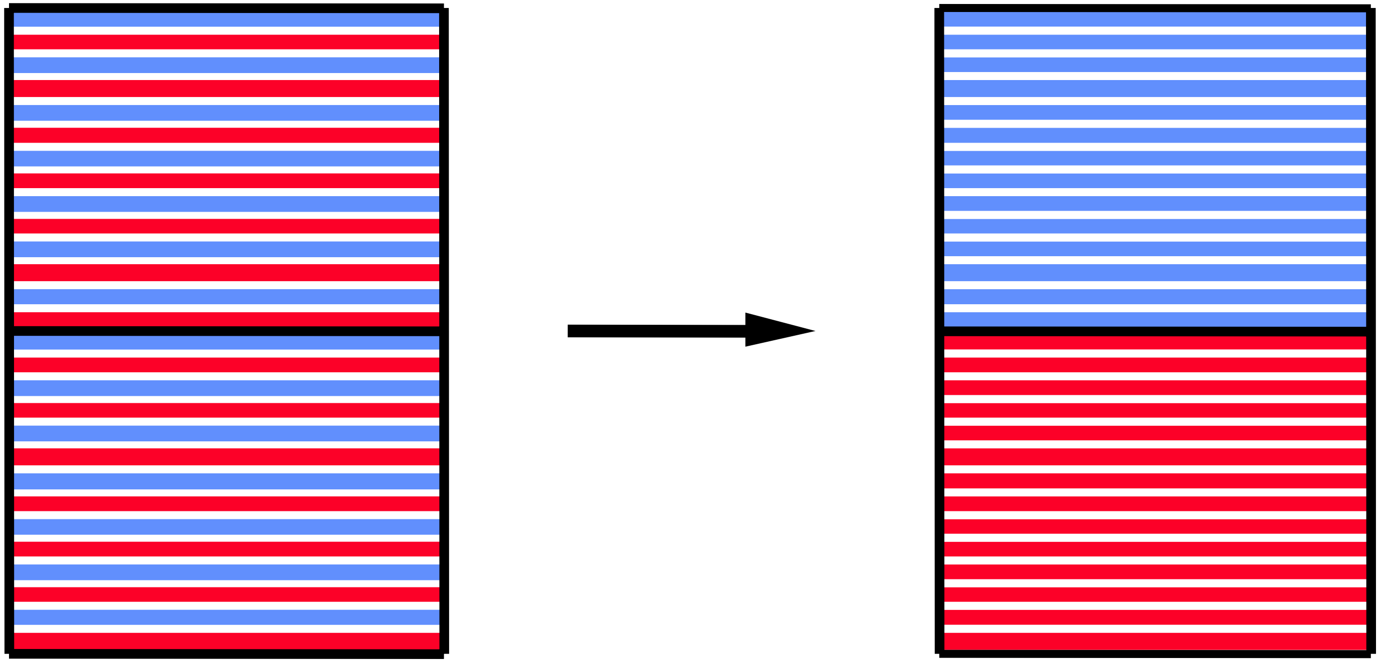
Interlaced Processing

- **Field coding**: each field is coded as a separate picture using fields for motion compensation

- **Frame coding**:
  - *Type 1*: the complete frame is coded as a separate picture
  - *Type 2*: the frame is scanned as macroblock pairs, for each macroblock pair: switch between frame and field coding



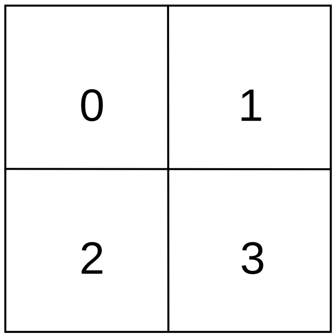
Macroblock-Based Frame/Field Adaptive Coding



A Pair of Macroblocks
in Frame Mode

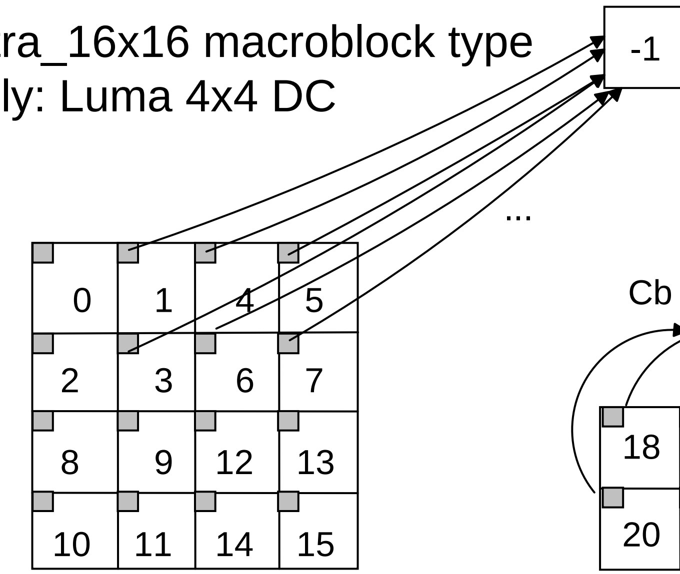
Top/Bottom Macroblocks
in Field Mode

Scanning of a Macroblock

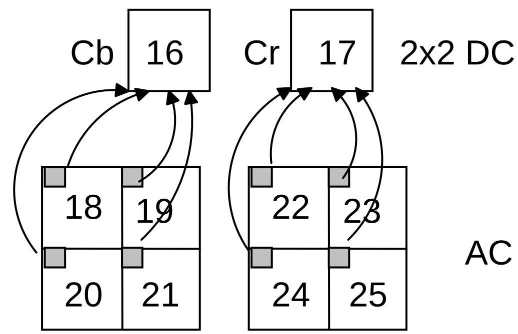


Coded Block Pattern for Luma in 8x8 block order: signals which of the 8x8 blocks contains at least one 4x4 block with non-zero transform coefficients

Intra_16x16 macroblock type only: Luma 4x4 DC

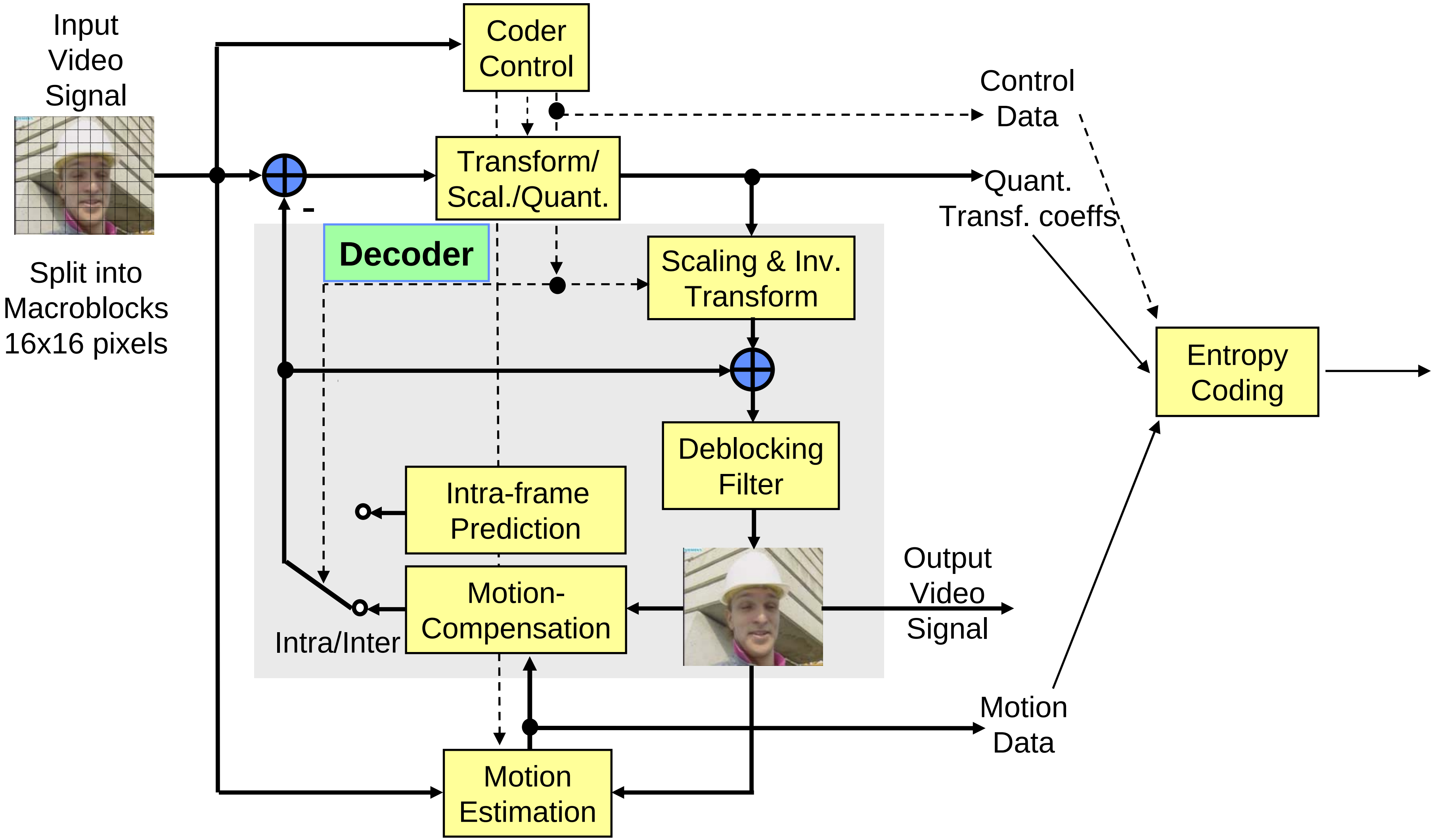


Luma 4x4 block order for 4x4 intra prediction and 4x4 residual coding

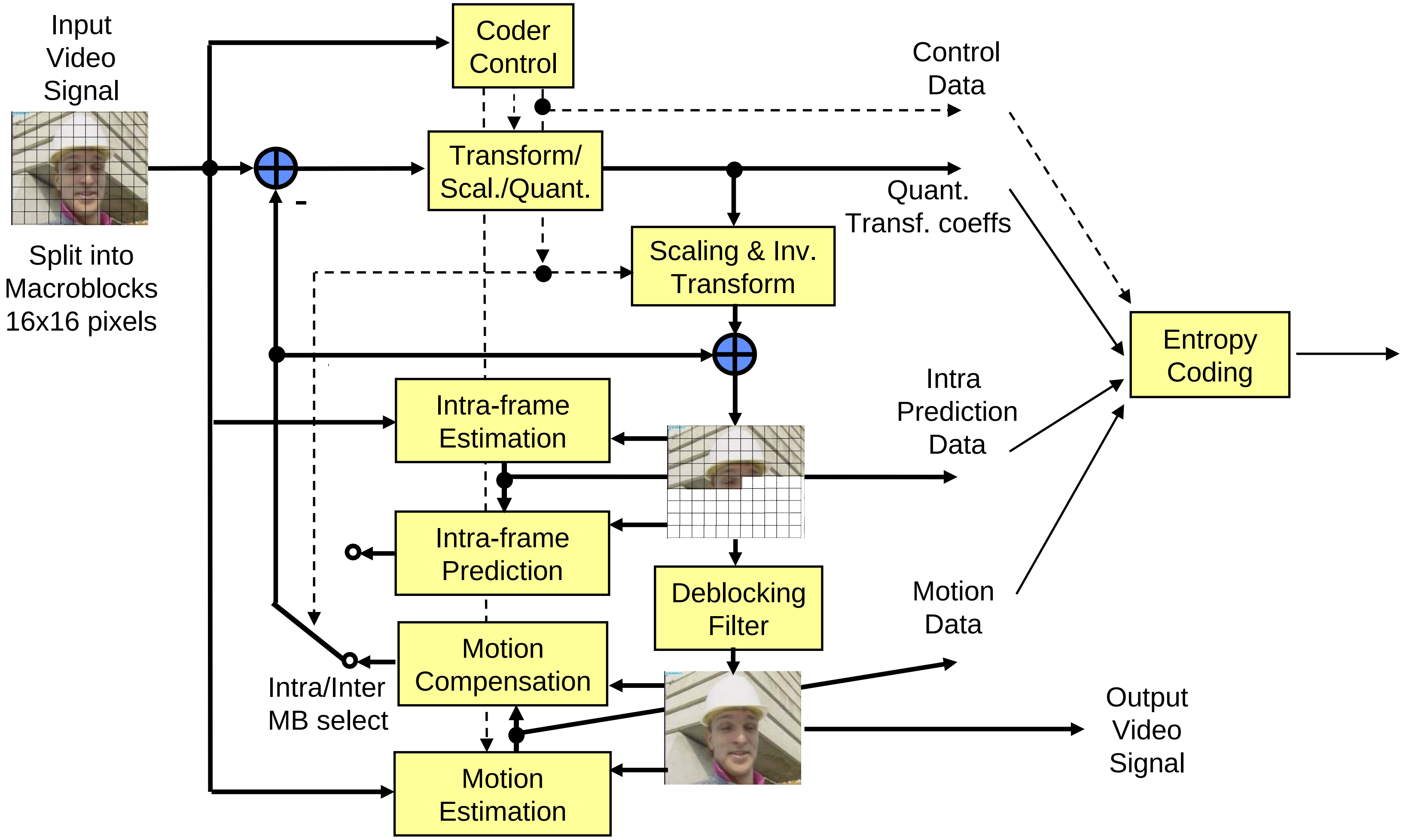


Chroma 4x4 block order for 4x4 residual coding, shown as 16-25, and intra 4x4 prediction, shown as 18-21 and 22-25

Basic Coding Structure



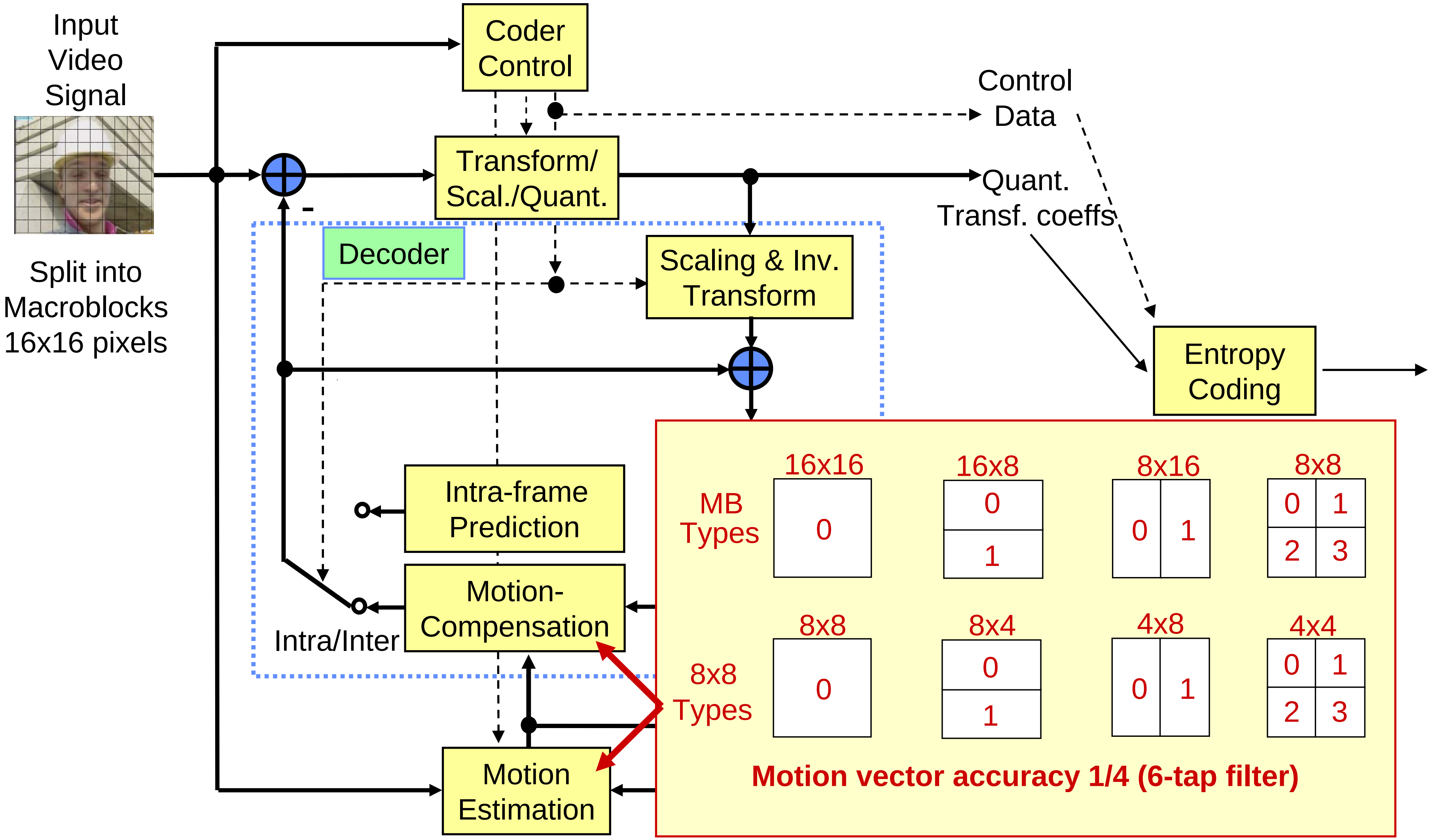
Basic Coding Structure



Common Elements with other Standards

- Macroblocks: 16x16 luma + 2 x 8x8 chroma samples
- Input: Association of luma and chroma and conventional sub-sampling of chroma (4:2:0)
- Block motion displacement
- Motion vectors over picture boundaries
- Variable block-size motion
- Block transforms
- Scalar quantization
- I, P, and B coding types

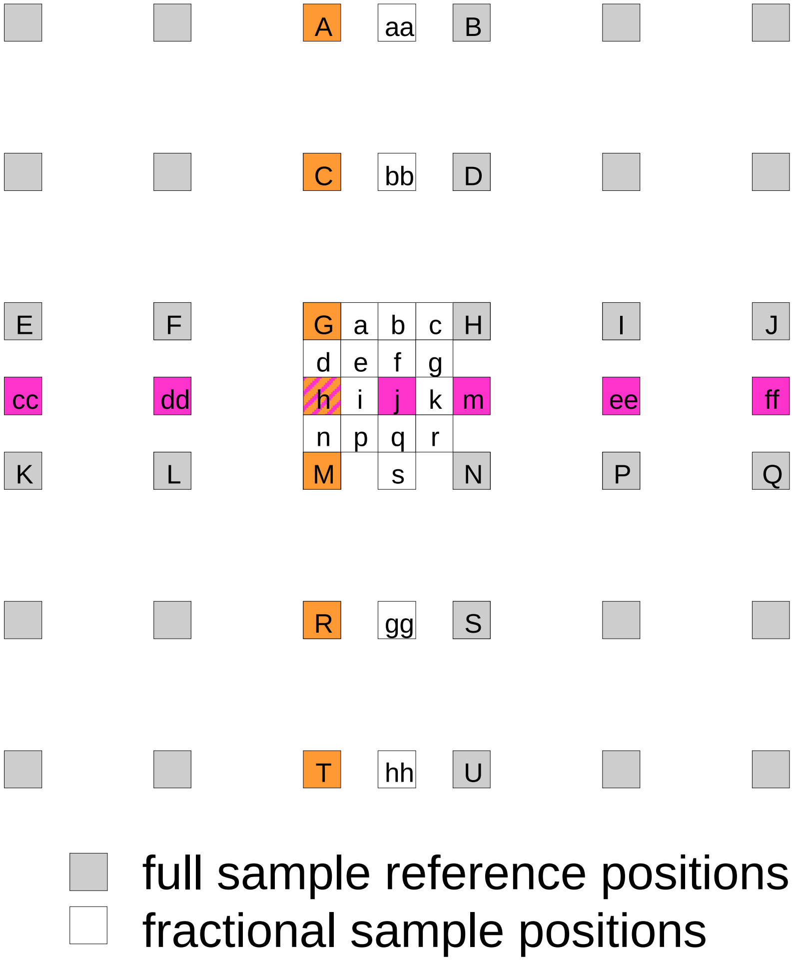
Motion Compensation Accuracy



Quarter Sample Luma Interpolation

- Half sample positions are obtained by applying a 6-tap filter with tap values: (1, -5, 20, 20, -5, 1)

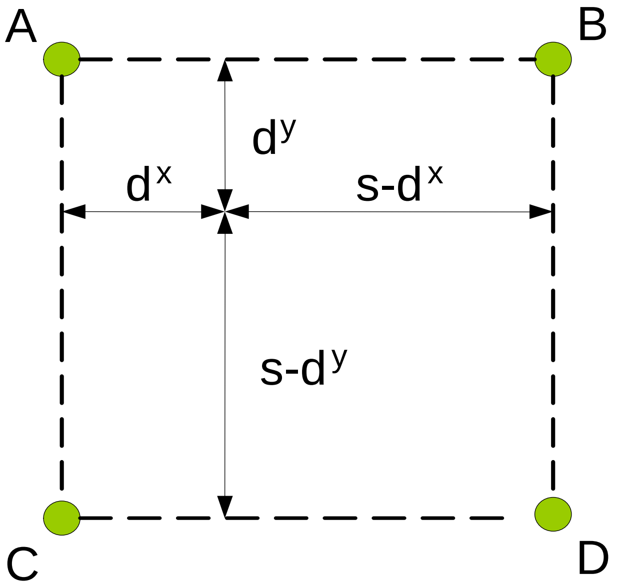
- Quarter sample positions are obtained by averaging samples at integer and half sample positions



Chroma Sample Interpolation

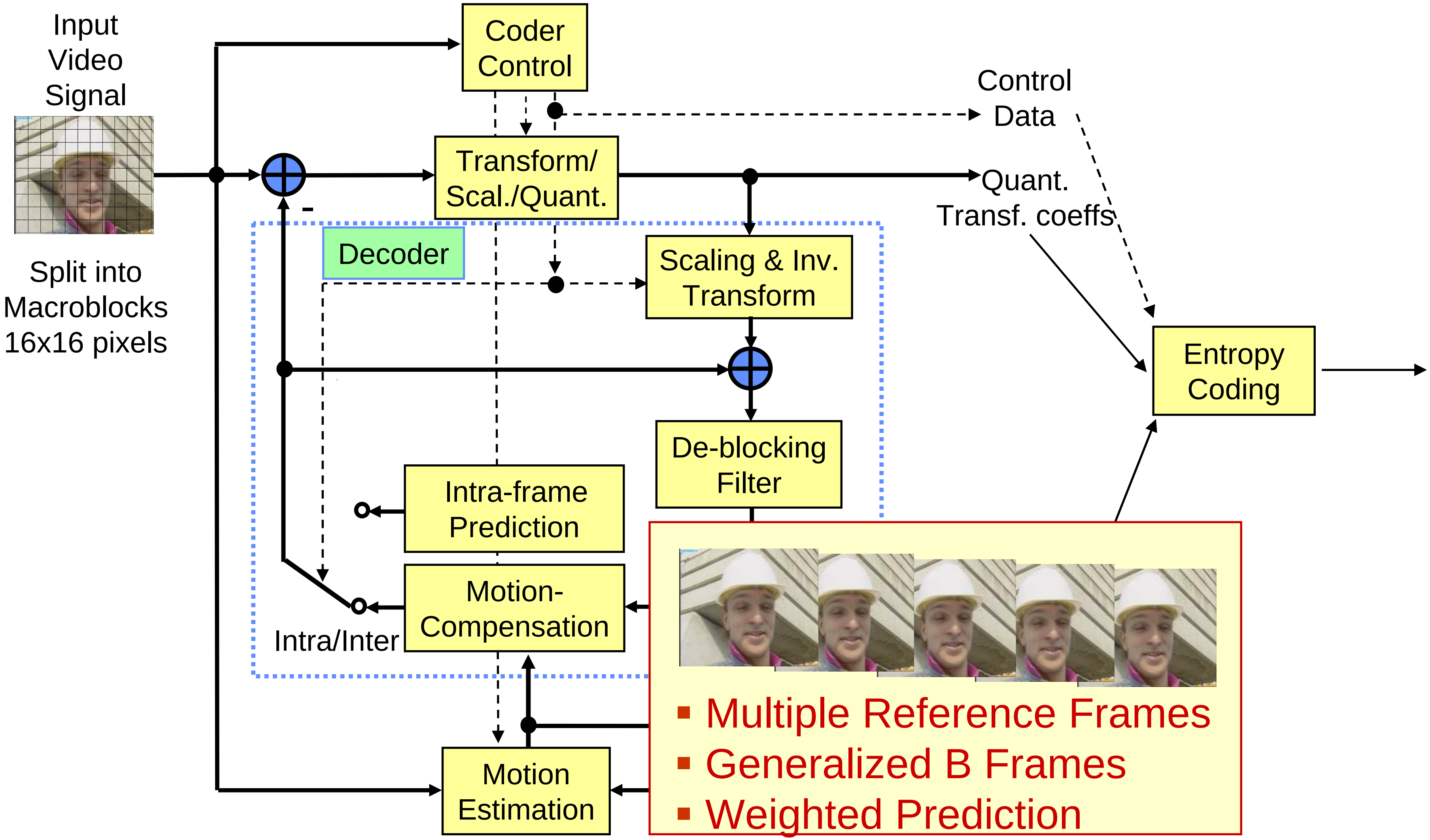
Chroma interpolation is 1/8-sample accurate since luma motion is 1/4-sample accurate

Fractional chroma sample positions are obtained using the equation:

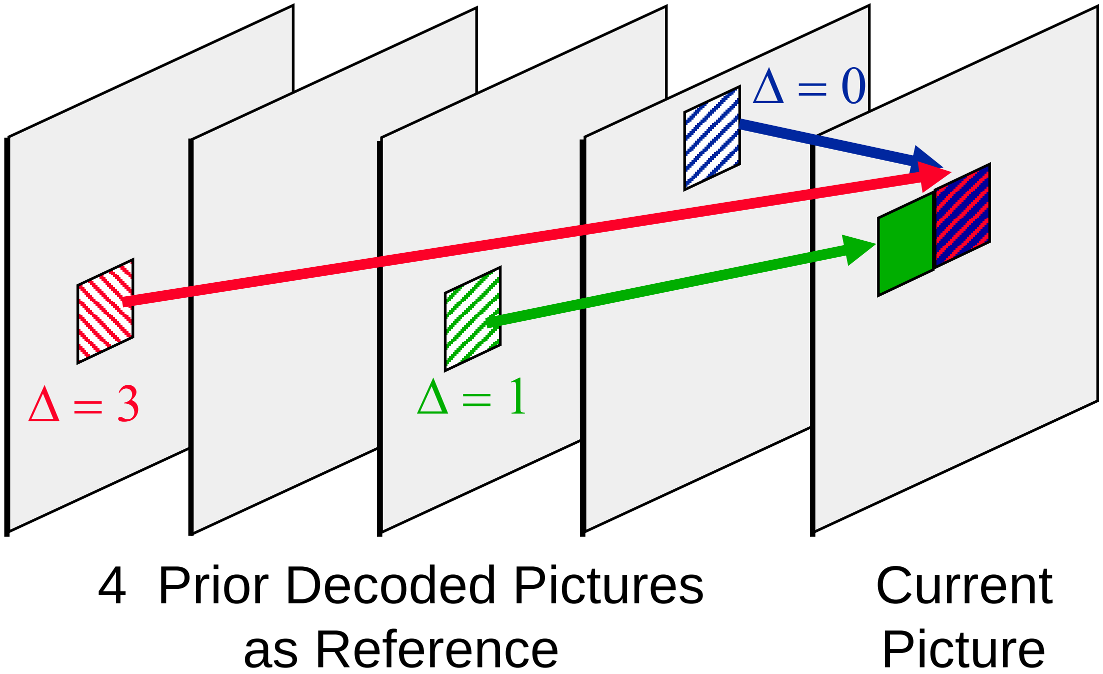


$$v = ((s-d^x)(s-d^y)A + d^x(s-d^y)B + (s-d^x)d^yC + d^xd^yD + s^2/2)/s^2$$

Multiple Reference Frames



Multiple Reference Frames and Generalized Bi-Predictive Frames

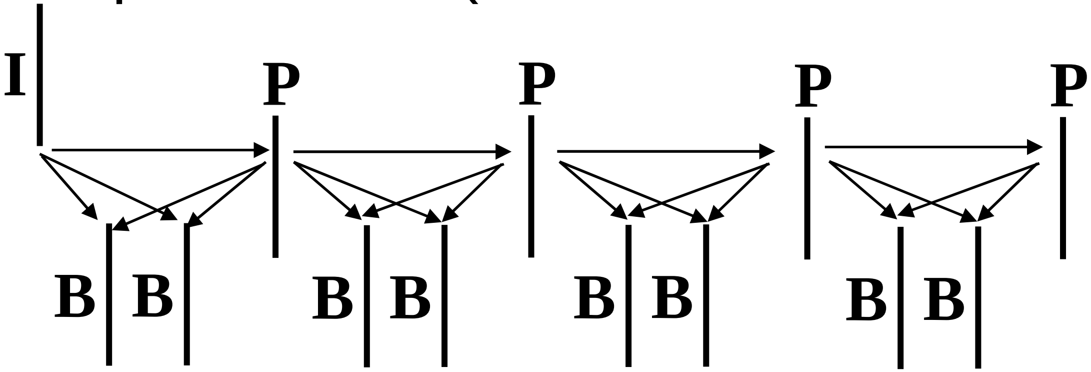


1. Extend motion vector by reference picture index Δ
2. Provide reference pictures at decoder side
3. In case of bi-predictive pictures: decode 2 sets of motion parameters

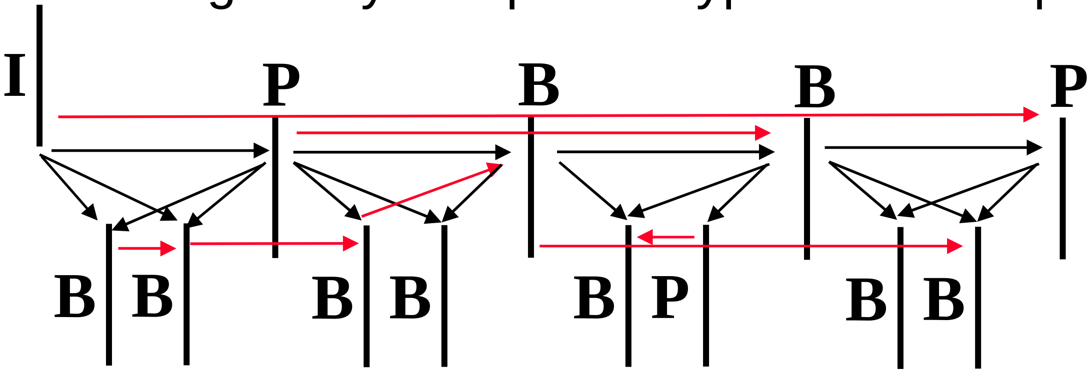
Can jointly exploit scene cuts, aliasing, uncovered background and other effects with one approach

New Types of Temporal Referencing

- Known dependencies (MPEG-1, MPEG-2, etc.)



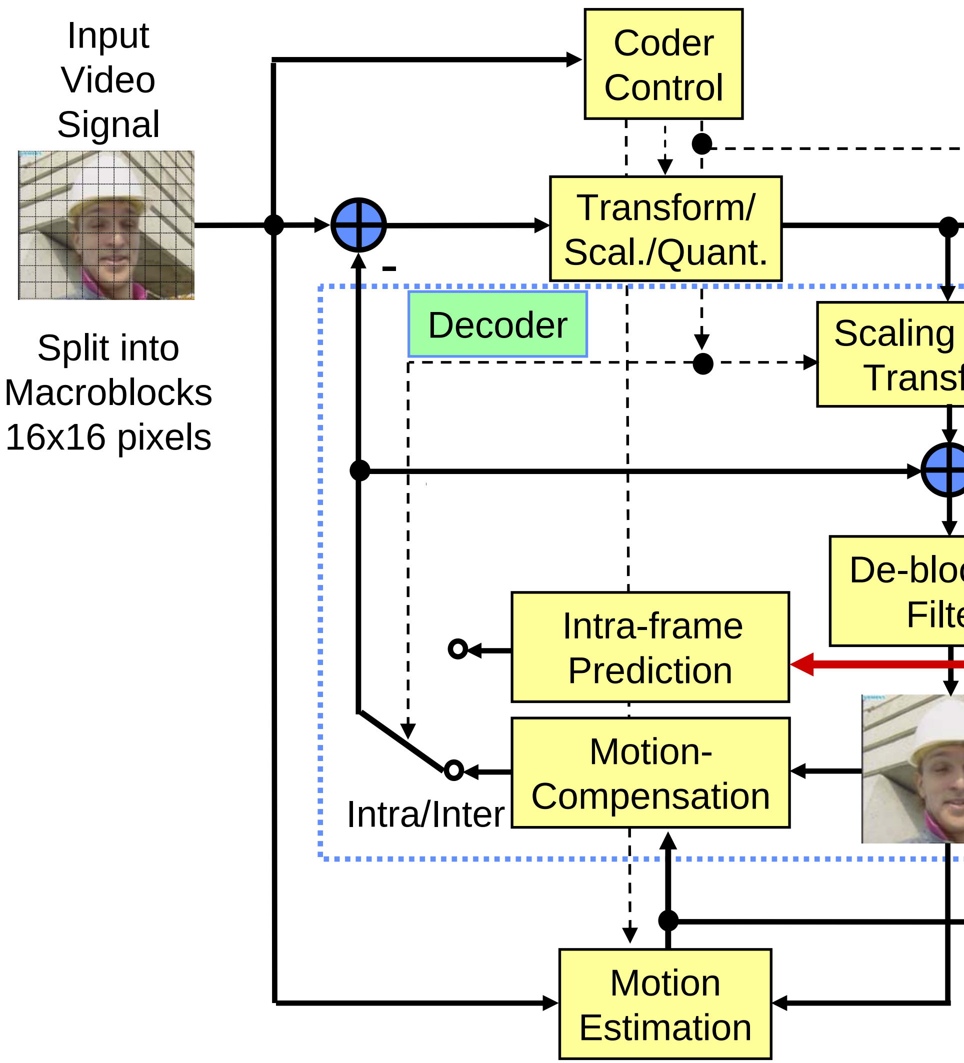
- New types of dependencies:
  - Referencing order and display order are decoupled
  - Referencing ability and picture type are decoupled



Weighted Prediction

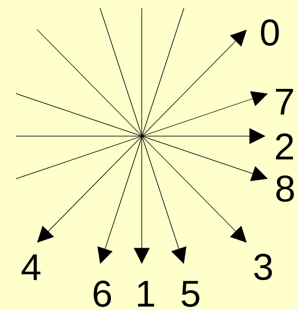
- In addition to shifting in spatial position, and selecting from among multiple reference pictures, each region's prediction sample values can be
  - multiplied by a weight, and
  - given an additive offset
- Some key uses:
  - Improved efficiency for B coding, e.g.,
    - accelerating motion,
    - multiple non-reference B temporally between reference pics
  - Excels at representation of fades:
    - fade-in
    - fade-out
    - cross-fade from scene-to-scene
- Encoder can apply this to both P and B prediction types

Intra Prediction



- Directional spatial prediction (9 types for luma, 1 chroma)

Q	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				



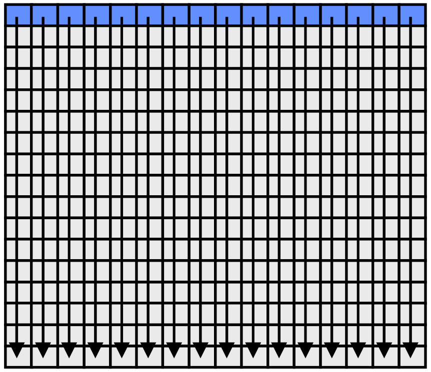
- e.g., Mode 3: diagonal down/right prediction a, f, k, p are predicted by (A + 2Q + I + 2) >> 2

Spatial prediction using surrounding "available" samples

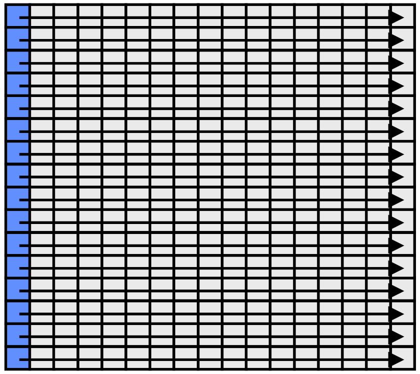
- **Available samples are…**
  - Previously reconstructed within the same slice at the decoder
  - Inside the same slice
- **Luma intra prediction either:**
  - Single prediction for entire 16x16 macroblock
    - 4 modes (vertical, horizontal, DC, planar)
  - 16 individual predictions of 4x4 blocks
    - 9 modes (DC, 8 directional)
- **Chroma intra prediction:**
  - Single prediction type for both 8x8 regions
    - 4 modes (vertical, horizontal, DC, planar)

16x16 Intra Prediction Directions

Mode 0 - Vertical



Mode 1 - Horizontal



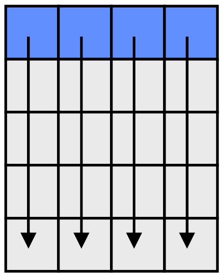
$$\text{Pred(x, y)} = \left[\sum_{x'=0}^{15} P(x',-1) + \sum_{y'=0}^{15} P(-1,y') + 16\right] >> 5$$ x, y = 0,…,15 (above and left available)

$$\text{Pred(x, y)} = \left[\sum_{y'=0}^{15} P(-1,y') + 8\right] >> 4$$ x, y = 0,…,15 (only left available)

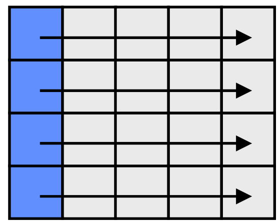
$$\text{Pred(x, y)} = \left[\sum_{x'=0}^{15} P(x',-1) + 8\right] >> 4$$ x, y = 0,…,15 (only above available)

4x4 Intra Prediction Directions

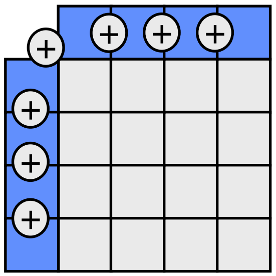
Mode 0 - Vertical



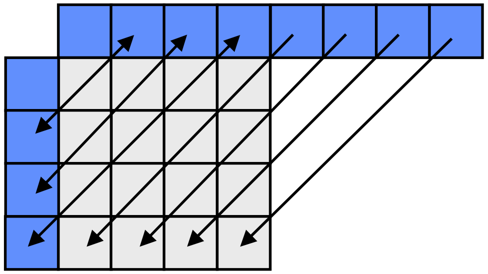
Mode 1 - Horizontal



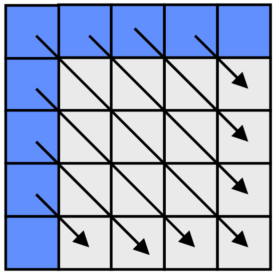
Mode 2 - DC



Mode 3 – Diagonal Down/Left

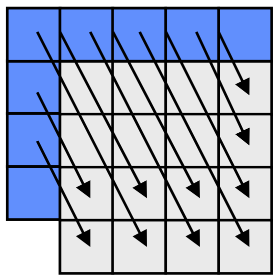


Mode 4 – Diagonal Down/Right

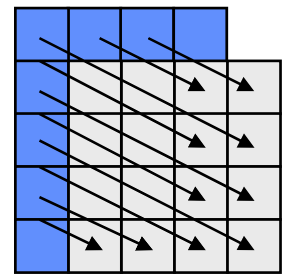


4x4 Intra Prediction Directions

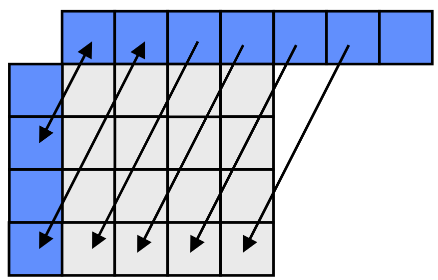
Mode 5 – Vertical-Right



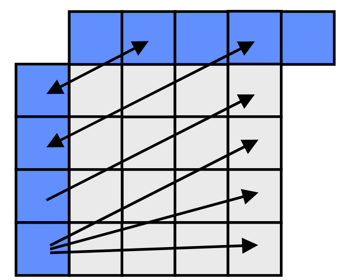
Mode 6 – Horizontal-Down



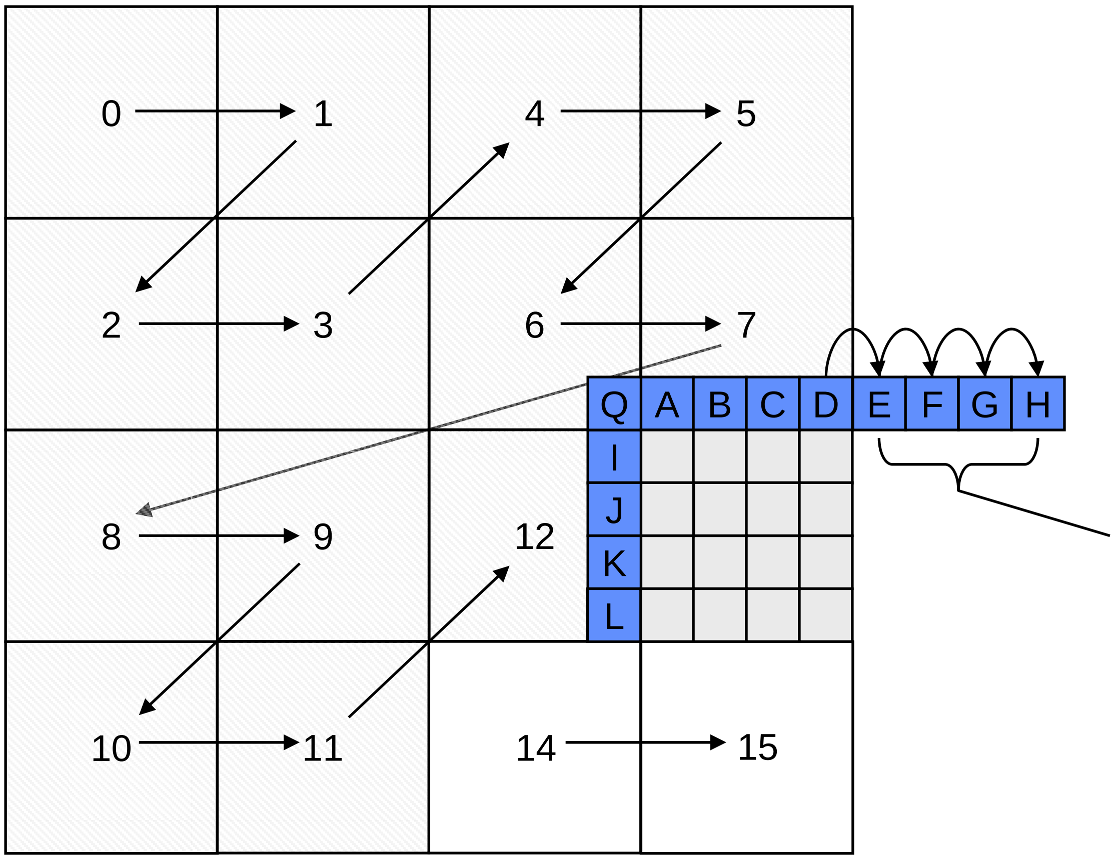
Mode 7 – Vertical-Left



Mode 8 – Horizontal-Up

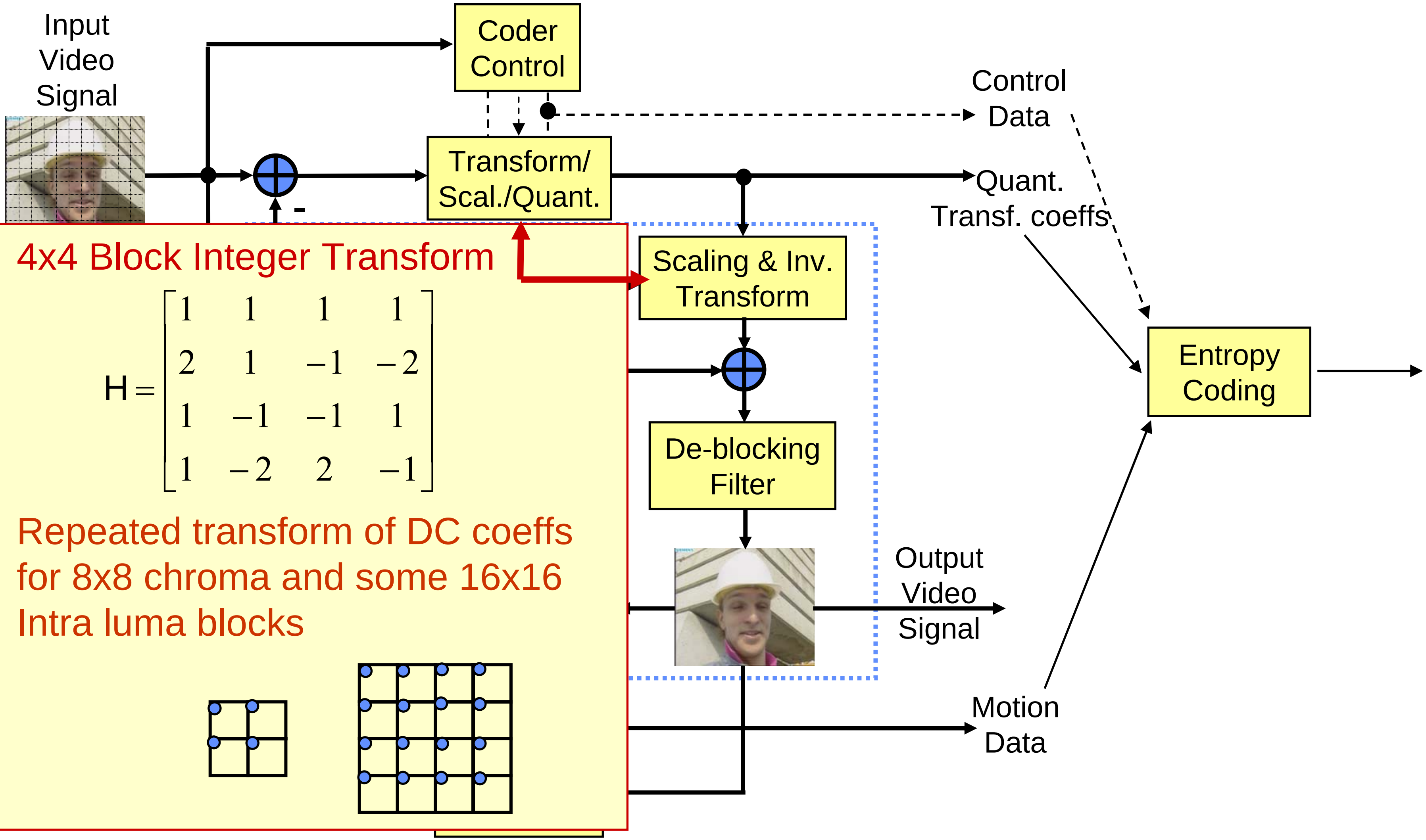


4x4 Boundary Conditions



EFGH not available since this 4x4 block is outside the macroblock – replace EFGH with value of D

Transform Coding



- 4x4 Block Integer Transform

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

- Repeated transform of DC coeffs for 8x8 chroma and some 16x16 Intra luma blocks

Integer Transforms (1)

- Separable transform of a block B_{4x4} of size *4x4*

$$\mathbf{C}_{4x4} = \mathbf{T}_v \cdot \mathbf{B}_{4x4} \cdot \mathbf{T}_h^T$$

- T_h, T_v: horizontal and vertical transform matrix

$$\mathbf{T}_v = \mathbf{T}_h = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

- 4x4 transform matrix:
    - Easy implementation (adds and shifts)
    - Different norms for even and odd rows of the matrix

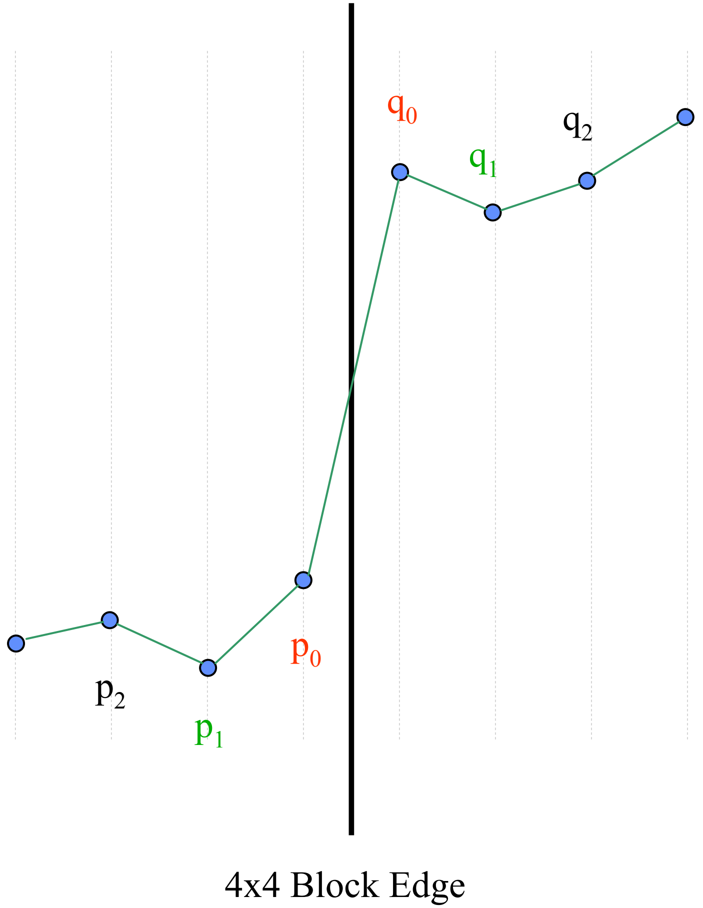
Quantization of Transform Coefficients

- Logarithmic step size control
- Smaller step size for chroma (per H.263 Annex T)
- Extended range of step sizes
- Can change to any step size at macroblock level
- Quantization reconstruction is one multiply, one add, one shift

Deblocking Filter

- Improves subjective visual *and* objective quality of the decoded picture
- Significantly superior to post filtering
- Filtering affects the edges of the 4x4 block structure
- Highly content adaptive filtering procedure mainly removes blocking artifacts and does not unnecessarily blur the visual content
  - On slice level, the global filtering strength can be adjusted to the individual characteristics of the video sequence
  - On edge level, filtering strength is made dependent on inter/intra, motion, and coded residuals
  - On sample level, quantizer dependent thresholds can turn off filtering for every individual sample
  - Specially strong filter for macroblocks with very flat characteristics almost removes “tiling artifacts”

Principle of Deblocking Filter



One dimensional visualization of an edge position

Filtering of p_0 *and* q_0 only takes place if:

1. $|p_0 - q_0| < \alpha(QP)$
2. $|p_1 - p_0| < \beta(QP)$
3. $|q_1 - q_0| < \beta(QP)$

Where $\beta(QP)$ is considerably smaller than $\alpha(QP)$

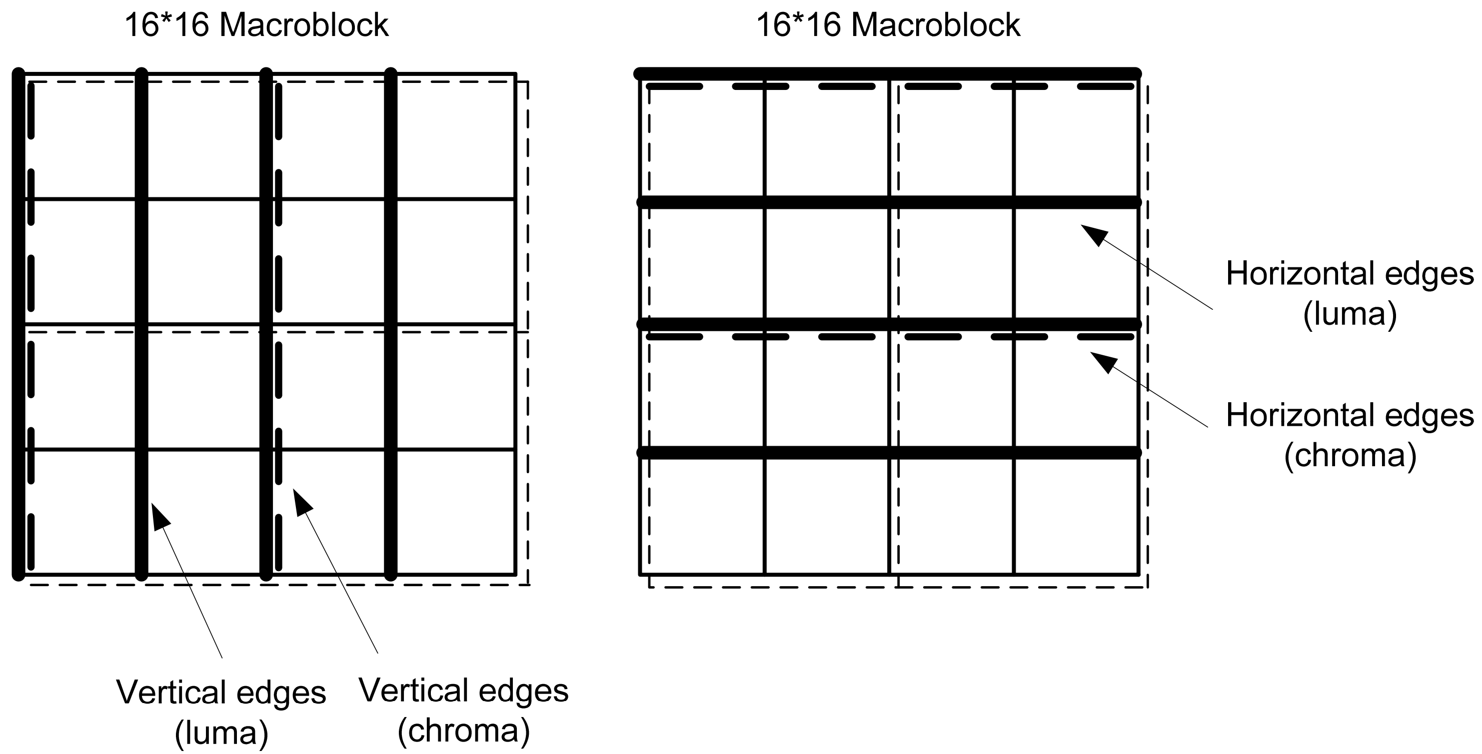
Filtering of p_1 *or* q_1 takes place if additionally :

1. $|p_2 - p_0| < \beta(QP)$ *or* $|q_2 - q_0| < \beta(QP)$

(QP = quantization parameter)

Order of Filtering

- Filtering can be done on a macroblock basis that is, immediately after a macroblock is decoded
- First, the vertical edges are filtered then the horizontal edges
- The bottom row and right column of a macroblock are filtered when decoding the corresponding adjacent macroblocks



Deblocking: Subjective Result for Intra

Highly compressed first decoded intra picture
at a data rate of 0.28 bit/sample

1) Without Filter

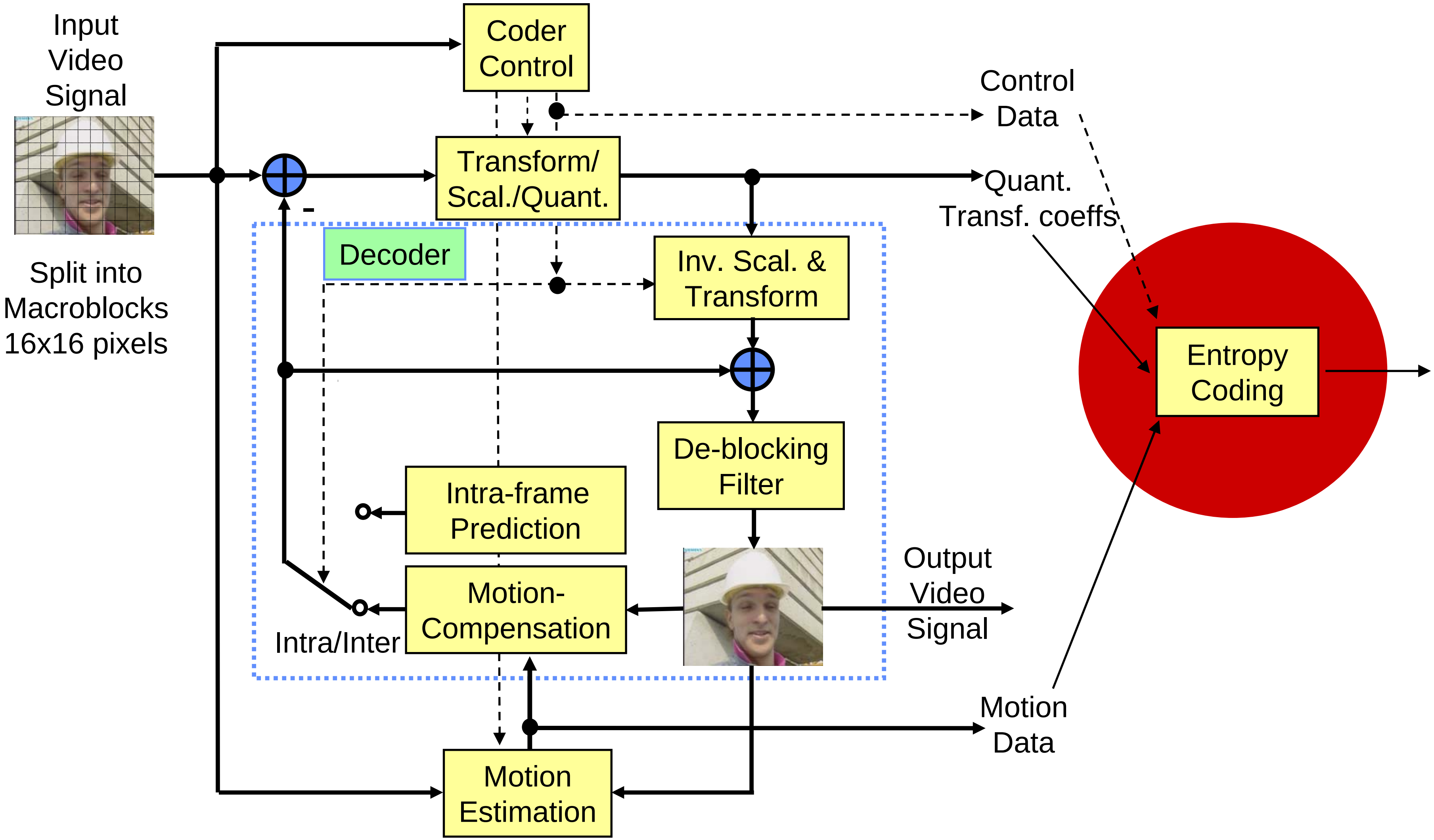
2) with H264/AVC Deblocking

Deblocking: Subjective Result for Inter

Highly compressed decoded inter picture

1) Without Filter 2) with H264/AVC Deblocking

Entropy Coding



Variable Length Coding

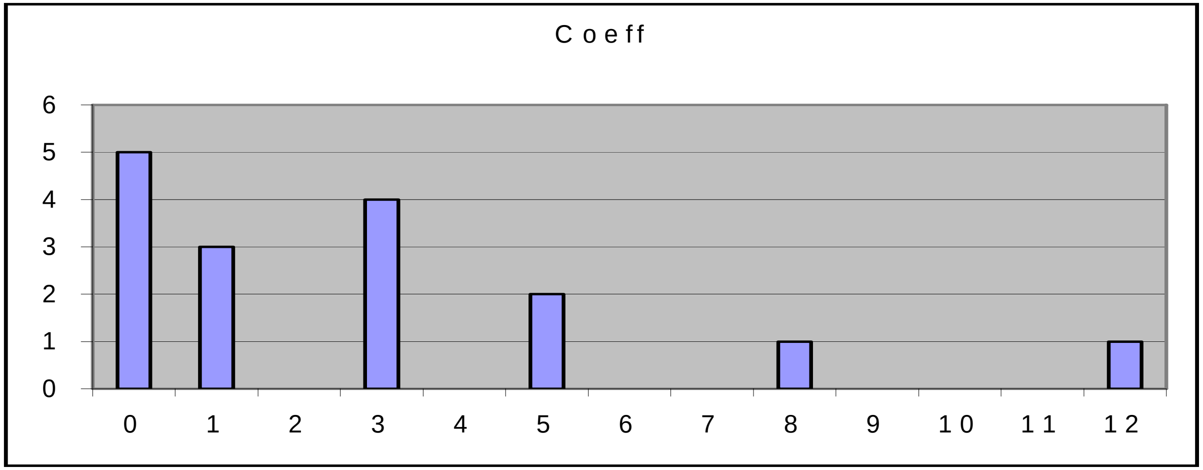
- Exp-Golomb code is used universally for almost all symbols except for transform coefficients
- Context adaptive VLCs for coding of transform coefficients
  - No end-of-block, but number of coefficients is decoded
  - Coefficients are scanned backwards
  - Contexts are built dependent on transform coefficients

Context Adaptive VLC (CAVLC)

- Transform coefficients are coded with the following elements:
  - Number of non-zero coefficients.
  - Levels and signs for all non-zero coefficients.
  - Total number of zeros before last non-zero coefficient.
  - Run before each non-zero coefficient

Number of Coefficients/Trailing ”1s”

- Typically the last non-zero coefficients have |Level | = 1
- The number of non-zero coefficients (example: N=6) and number of ”Trailing 1s” (T1s=2) are coded in a combined symbol
  - In this way typically > 50% of the coefficients are signalled as T1s and no other level information than sign is then needed for these coefficients.
- The VLC table to use is adaptively chosen based on the number of coefficients in neighboring blocks.



Reverse Scanning and Level Coding

- In a forward scan coefficients levels typically start with high values and decrease towards 1 (Trailing ”1s”)
- Therefore the value of the last nonzero coefficient is more accurately predictable than for the first one.
- Efficient adaptation is obtained by
  - Start with a default VLC table for the first coefficient in the reverse scan
  - The table to use for the next coefficient is then selected based on the context as adapted by previously coded levels in the reverse scan.
  - To adapt to a wide variety of input statistics there are 7 structured VLC tables to choose between.

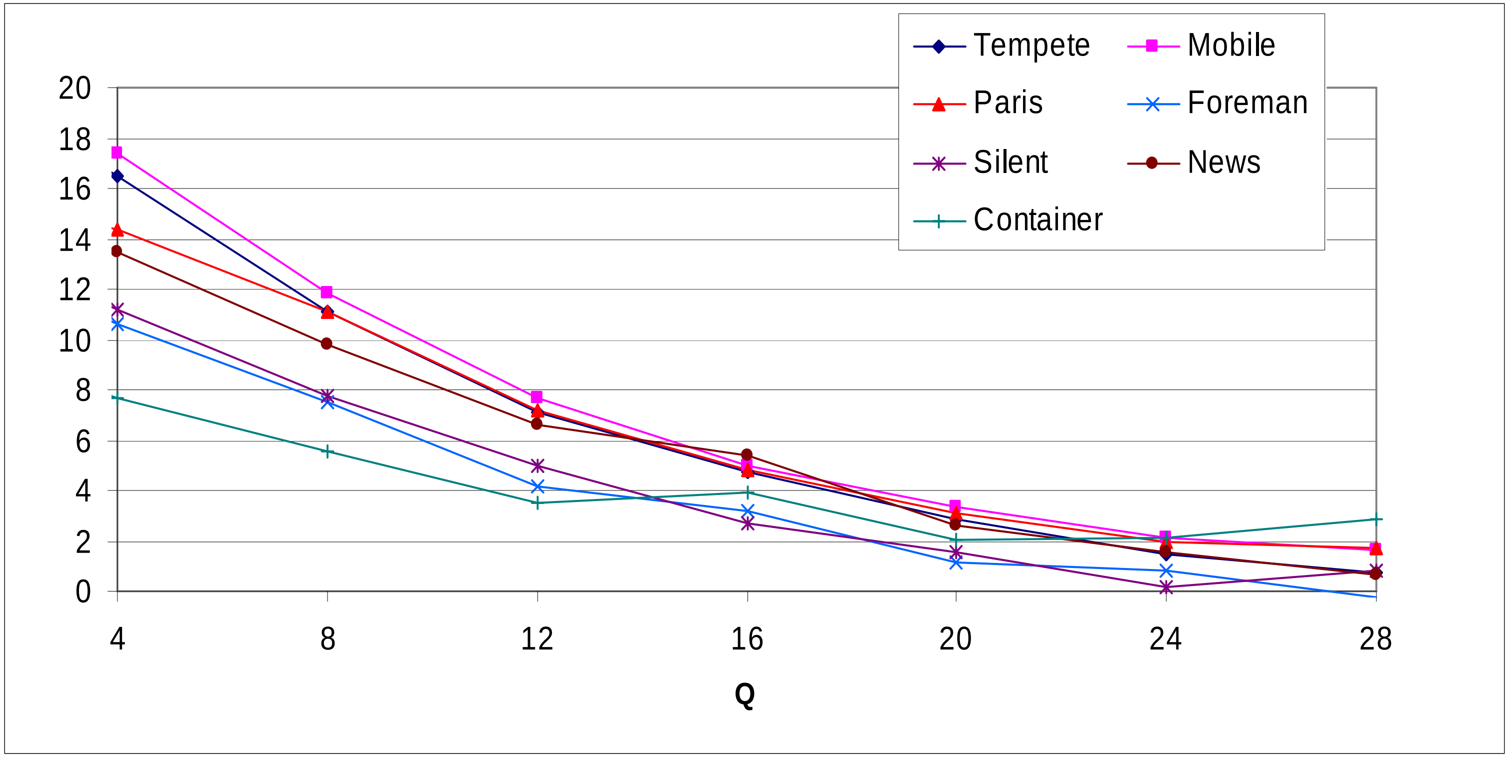
Run Information: TotalZeros and RunBefore

- TotalZeros
  - This is the total number of zeros before the last nonzero coefficient in a forward scan.
  - Since the number of non-zero coefficients (N) is already known, the maximum value of TotalZeros is: 16 – N, and a VLC of appropriate length can be used.
- RunBefore
  - Finally, in a reverse scan order, the run before each non-zero coefficient is coded.
  - Since this run can take on only a certain set of values, depending on TotalZeros and runs coded so far, a VLC with optimal length and statistics can always be used.

Bit-Rate Savings for CAVLC

Bit-rate Reduction
Relative to Run-Level UVLC [%]

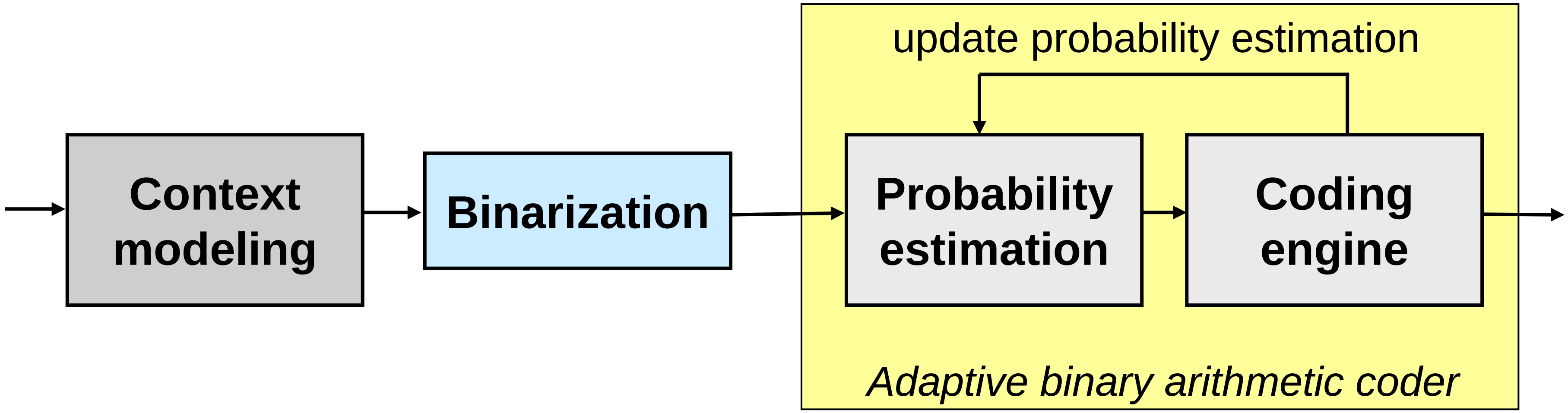
Inter-Picture Coding



Context-based Adaptive Binary Arithmetic Codes (CABAC)

- Usage of adaptive probability models for most symbols
- Exploiting symbol correlations by using contexts
- Restriction to binary arithmetic coding
  - Simple and fast adaptation mechanism
  - Fast binary arithmetic codec based on table look-ups and shifts only

CABAC: Technical Overview



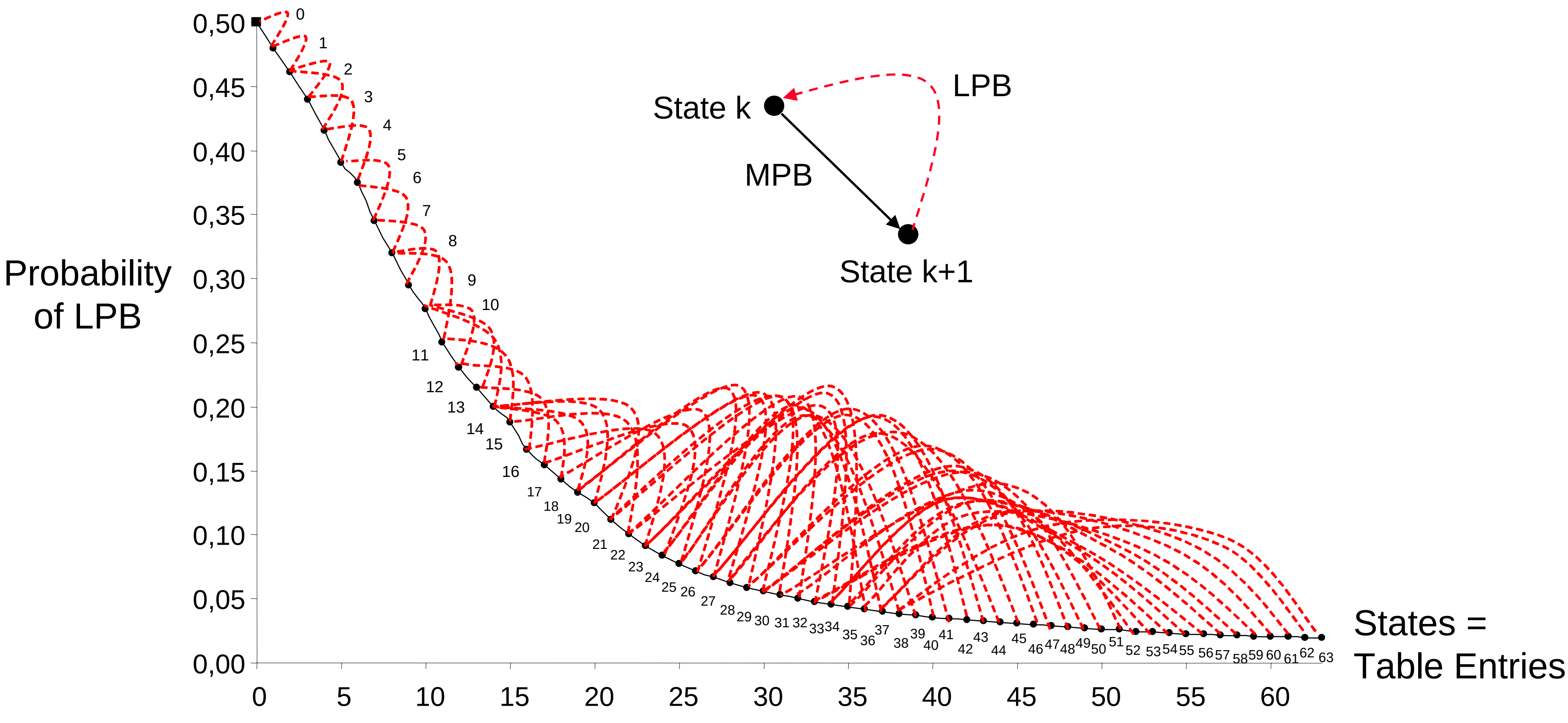
Chooses a model conditioned on past observations

Maps non-binary symbols to a binary sequence

Uses the provided model for the actual encoding and updates the model

Probability Estimation

- Probability estimation is realized via table look-up
- Table contains states and transition rules upon receipt of MPB or LPB



Binarization

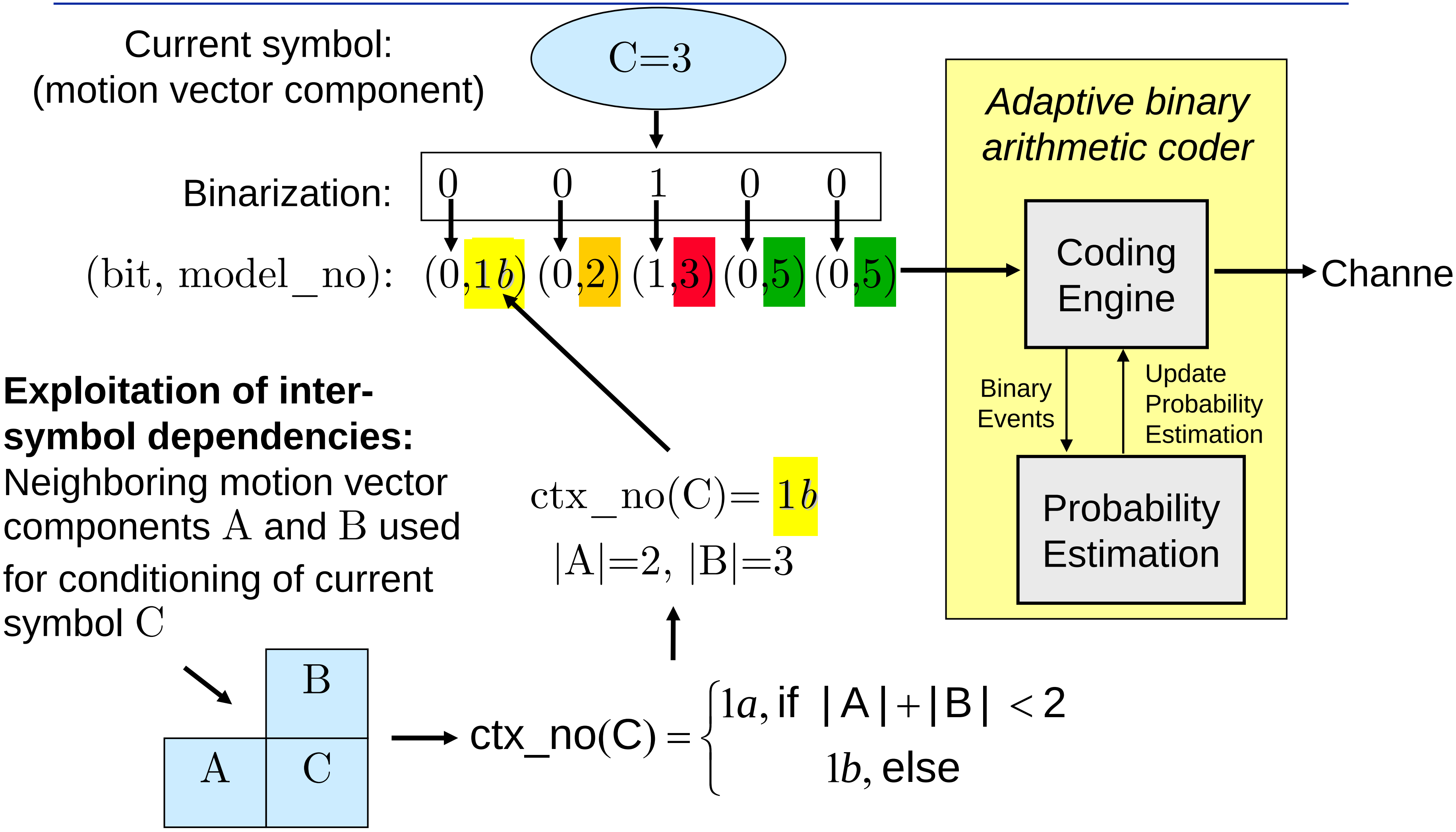
Symbol	Binarization
0	1
1	0 1
2	0 0 1
3	0 0 0 1
4	0 0 0 0 1
5	0 0 0 0 0 1
6	0 0 0 0 0 0 1
.	...
Bin_num	1 2 3 4 5 6 7 ...

Mapping to a binary sequence, e.g., using the unary code tree:

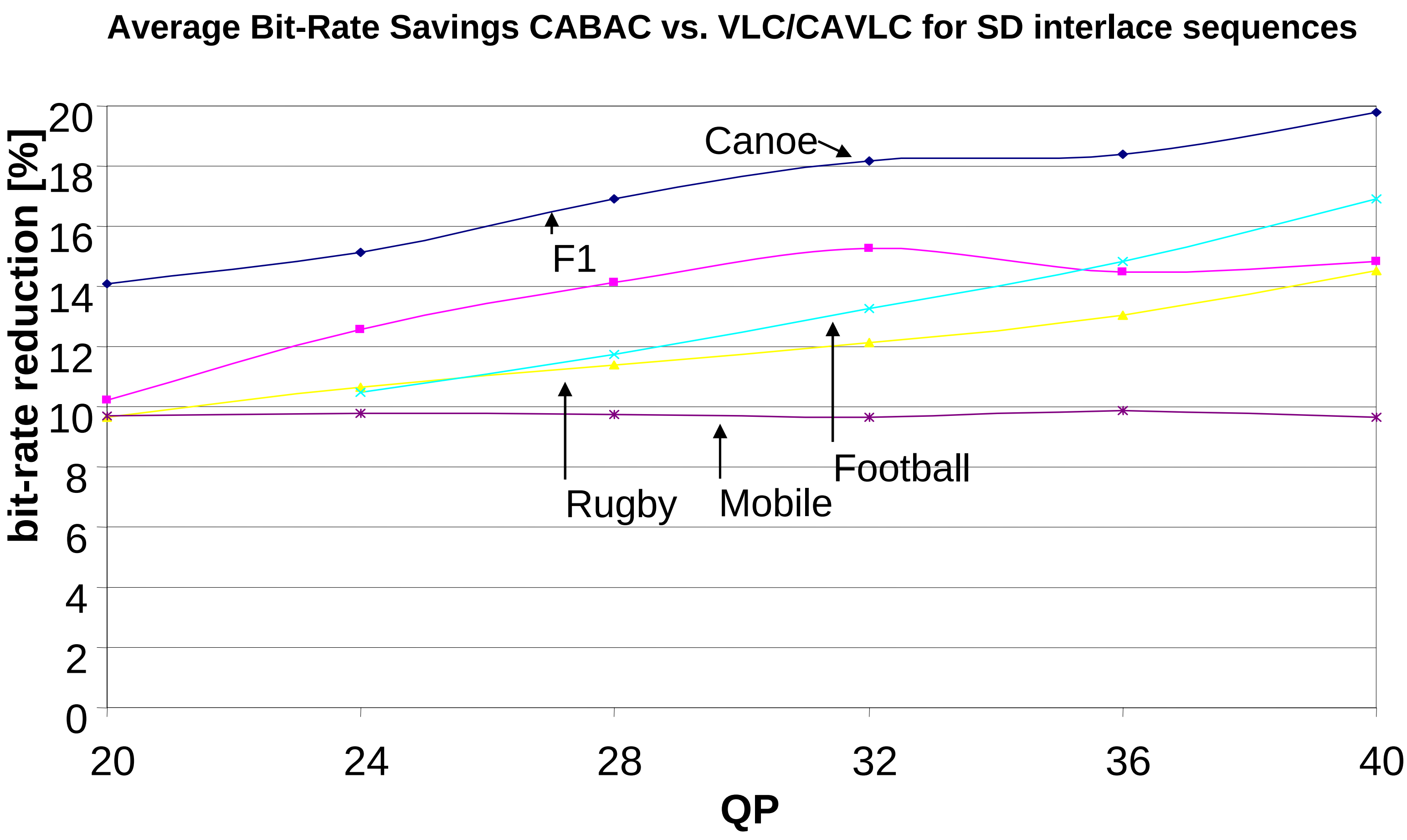
- Applies to all non-binary syntax elements except for macroblock type
- Ease of implementation
- Discriminate between binary decisions (bins) by their position in the binary sequence

⇒ Usage of different models for different bin_num in the table-based arithmetic coder

Context Modeling Example: Coding of MV



Bit-Rate Savings for CABAC



Coder Control

- Coder control is a non-normative part of H.264/AVC
- Goal within standardization process: demonstrate H.264/AVC performance and make design decisions using common conditions
- Choose coding parameters at encoder side
  „What part of the video signal should be coded using what method and parameter settings?"
- Constrained problem:

$$\min_{\mathbf{p}} D(\mathbf{p}) \quad \text{s.t.} \quad R(\mathbf{p}) \le R_T$$

D - Distortion
R - Rate
R_T - Target rate
$\mathbf{p}$ - Parameter Vector

- Unconstrained Lagrangian formulation:

$$\mathbf{p}_{opt} = \arg\min_{\mathbf{p}} \left\{ D(\mathbf{p}) + \lambda \cdot R(\mathbf{p}) \right\}$$

with λ controlling the rate-distortion trade-off

Rate-Constrained Mode Decision

- For given values of Q and λ_M, minimize

$$D_2(M \mid Q) + \lambda_M \cdot R(M \mid Q)$$

M - Evaluated macroblock mode out of a set of possible modes
Q - Value of quantizer control parameter for transform coefficients
λ_M - Lagrange parameter for mode decision
D_2 - Sum of squared differences (luma & chroma)
R - Number of bits associated with header, motion, transform coefficients

- Set of possible macroblock modes
  - Dependent on frame type (e.g. I, P, B)
  - For instance, P frame in H.264|AVC:

$$M \in \{\mathrm{SKIP},\ \mathrm{INTER\_16x16},\ \mathrm{INTER\_16x8},\ \mathrm{INTER\_8x16},\ \mathrm{INTER\_8x8},\ \mathrm{INTRA\_4x4},\ \mathrm{INTRA\_16x16}\}$$

- Prior to macroblock mode decision: sub macroblock (8x8) mode decision

Rate-Constrained Motion Estimation

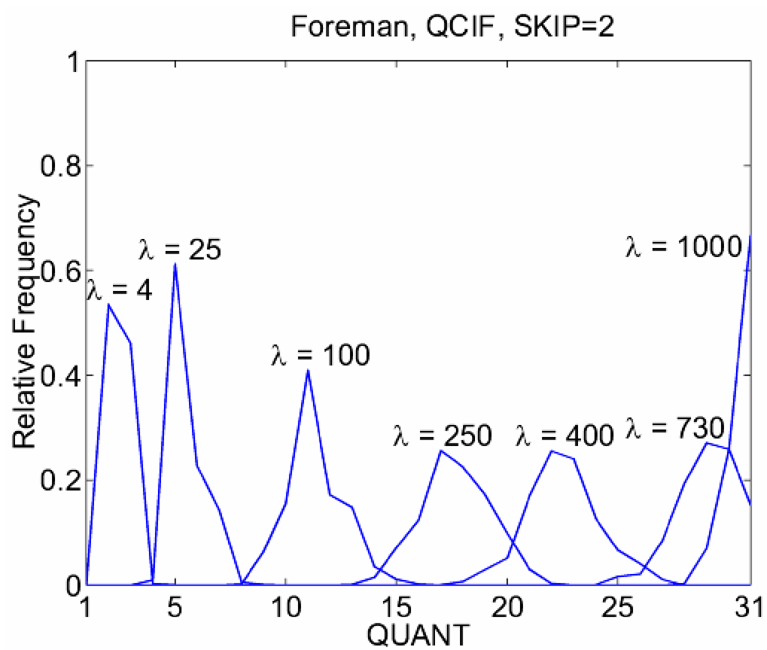
- Integer-pixel motion search as well as fractional sample search is performed by minimizing

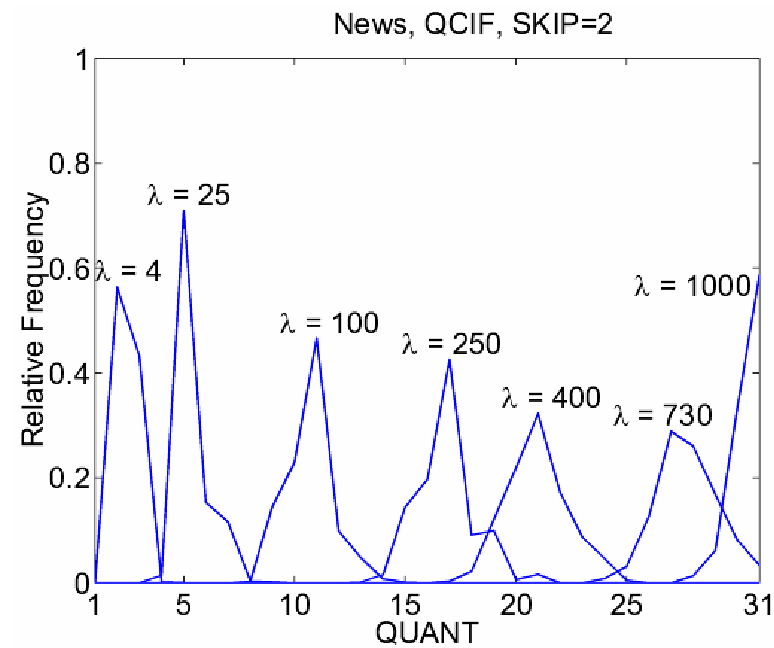
$$D_1(\mathbf{m}) + \lambda_D \cdot R(\mathbf{m} \mid \mathbf{p}_m)$$

$\mathbf{m}$	- Motion vector containing spatial displacement and picture reference parameter Δ
$\mathbf{p}_m$	- Predictor for motion vector
λ_D	- Lagrange parameter for motion estimation
D_1	- Sum of absolute differences (luminance)
R	- Number of bits associated with motion information

Relationship between λ and *QP*

- <u>Experiment:</u>
  - Fix Lagrangian multiplier λ_M and $\lambda_D = \sqrt{\lambda_M}$
  - Add modes with quantizer changing (DQUANT)
  - Perform rate-constrained mode decision
  - See [Wiegand and Girod, ICIP 2001]





Relationship between λ and *QP*

- H.263 / MPEG-4p2:

$$\lambda_M = 0.85 \cdot QP_{H.263}^2$$

$$\lambda_D = \sqrt{\lambda_M}$$

- H.264/AVC:

$$QP_{H.263} \approx 2^{(QP-12)/6}$$

$$\Rightarrow \quad \lambda_M = 0.85 \cdot 2^{(QP-12)/3}$$

$$\lambda_D = \sqrt{\lambda_M}$$

A Comparison of Performance

- Test of different standards (Trans. on Circuits and Systems for Video Technology, July 2003, Wiegand *et al*)
- Using same rate-distortion optimization techniques for all codecs
- “Streaming” test: High-latency (included B frames)
- “Real-time conversation” test: No B frames
- “Entertainment-quality application“ test: SD & HD resolutions
- Several video sequences for each test
- Compare four codecs:
  - MPEG-2 (in Main profile high-latency/streaming test only)
  - H.263 (High-Latency profile, Conversational High-Compression profile, Baseline profile)
  - MPEG-4 Visual (Simple profile and Advanced Simple profile with & without B pictures)
  - H.264/AVC (Main profile and Baseline profile)

Caution: Your Mileage Will Vary

- Theoretical performance versus actual implementation quality is a serious consideration
- Need tests on larger body of material for strong statistical significance
- PSNR analysis and perceptual quality can differ

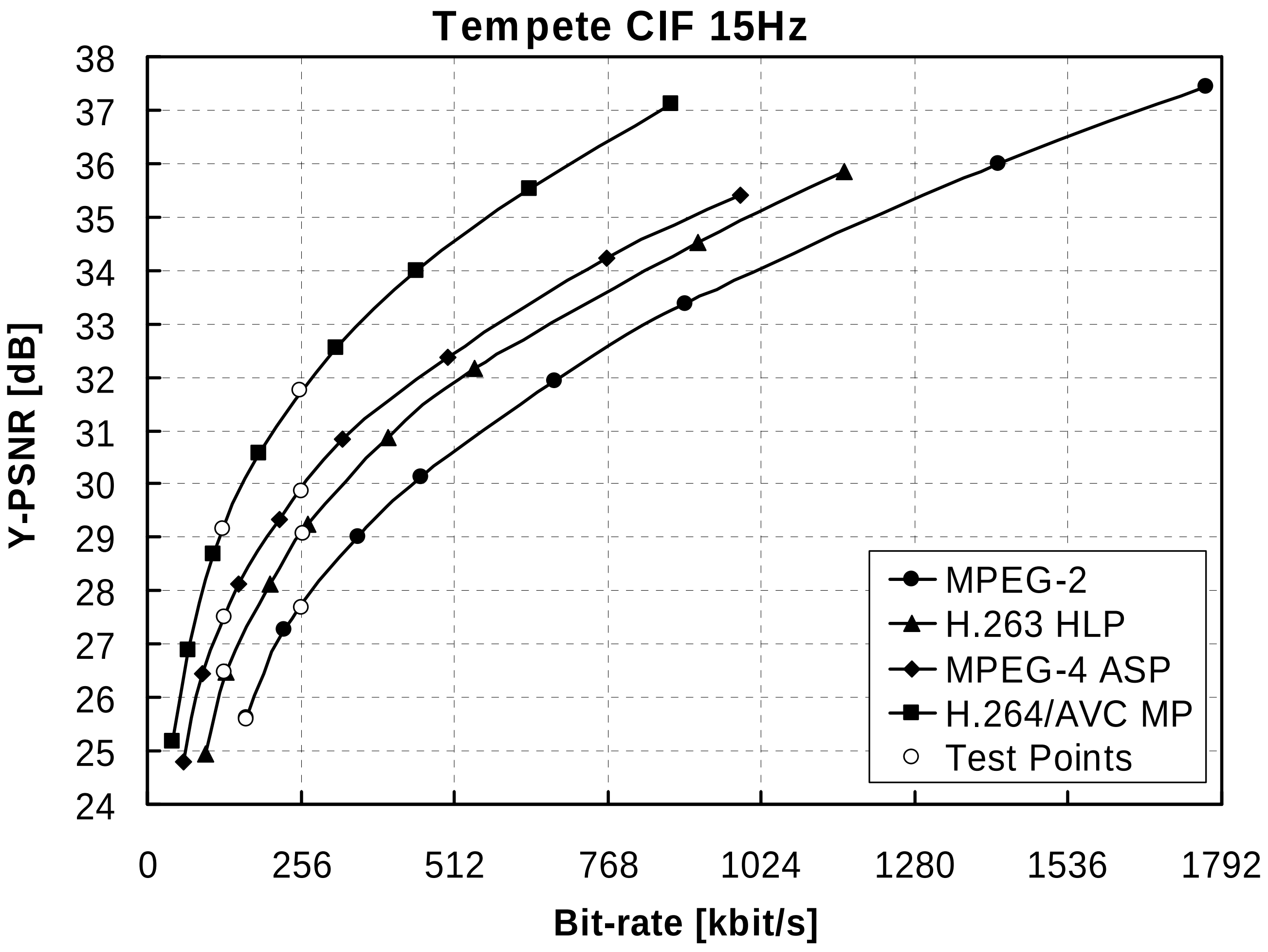
Test Set for Streaming Applications

Name	Resolution	Duration	Characteristics
Foreman	QCIF	10 sec.	Fast camera and content motion with pan at the end
Container Ship	QCIF	10 sec.	Still camera on slow moving scene
News	QCIF	10 sec.	Still camera on human subjects with synthetic background
Tempete	QCIF	8.67 sec.	Camera zoom; spatial detail; fast random motion
Bus	CIF	5 sec.	Fast translational motion and camera panning; moderate spatial detail
Flower Garden	CIF	8.33 sec.	Slow and steady camera panning over landscape; spatial and color detail
Mobile & Calendar	CIF	8.33 sec.	Slow panning and zooming; complex motion; high spatial and color detail
Tempete	CIF	8.67 sec.	Camera zoom; spatial detail; fast random motion

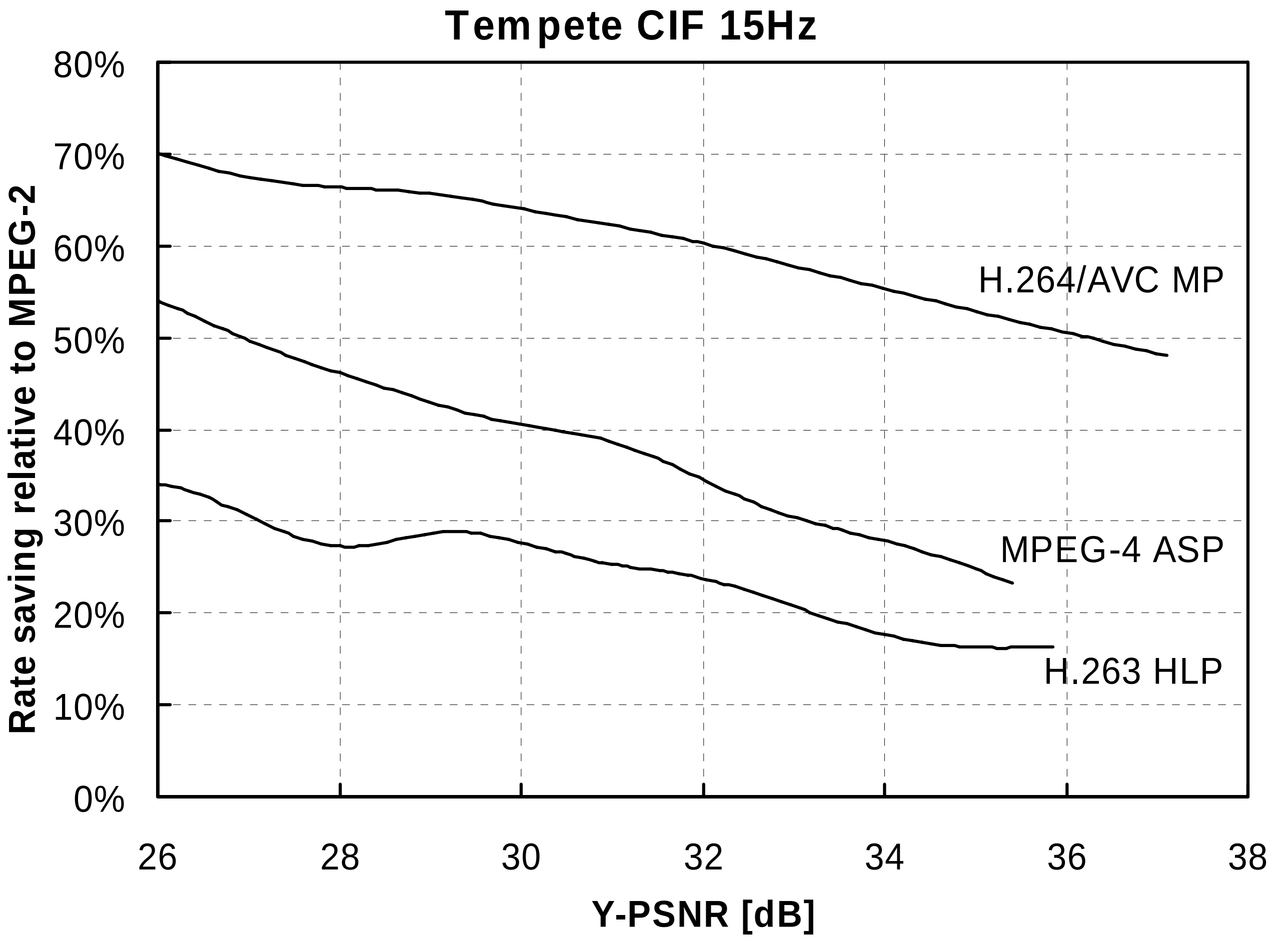
Test Results for Streaming Application

	Average bit-rate savings relative to:		
Coder	MPEG-4 ASP	H.263 HLP	MPEG-2
H.264/AVC MP	37.44%	47.58%	63.57%
MPEG-4 ASP	-	16.65%	42.95%
H.263 HLP	-	-	30.61%

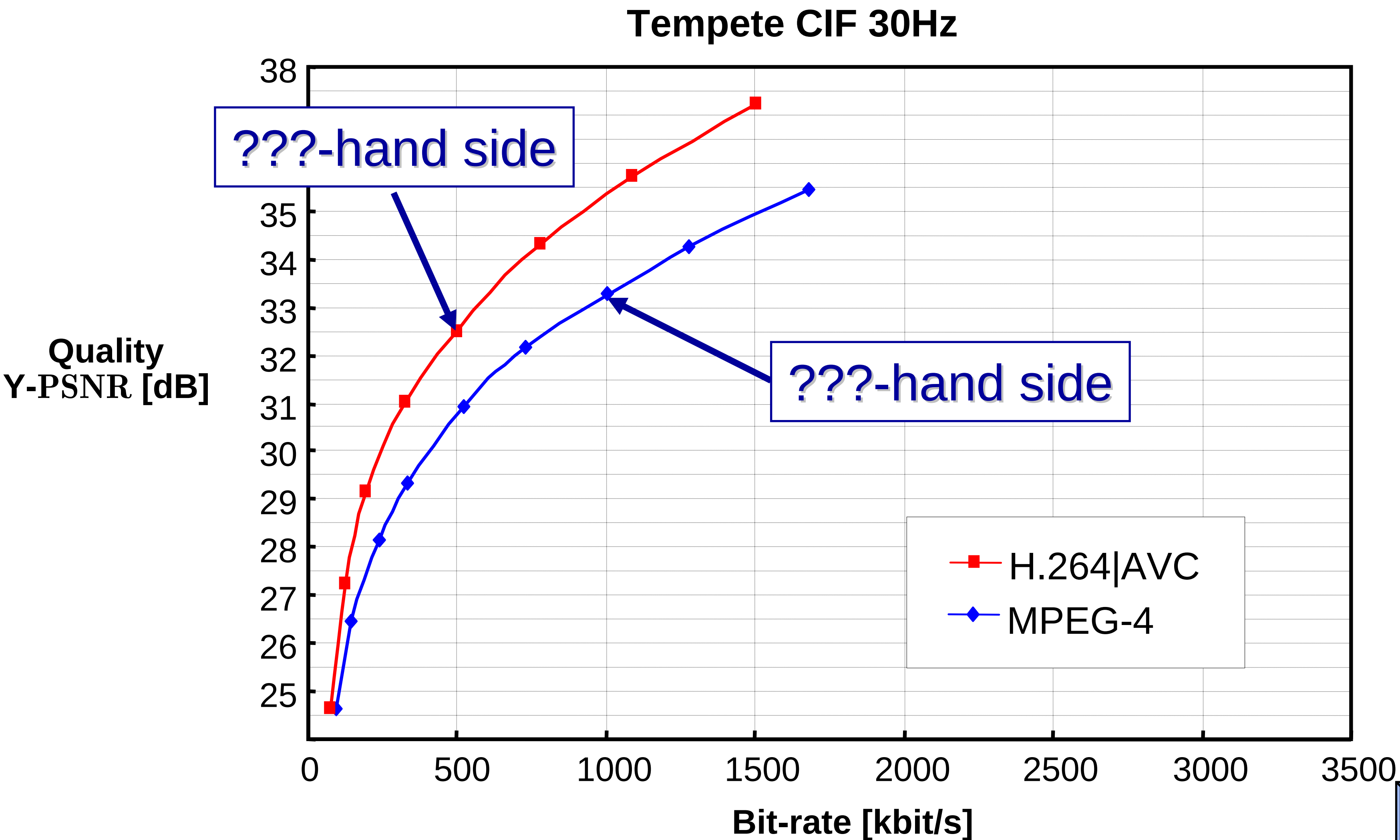
Example Streaming Test Result



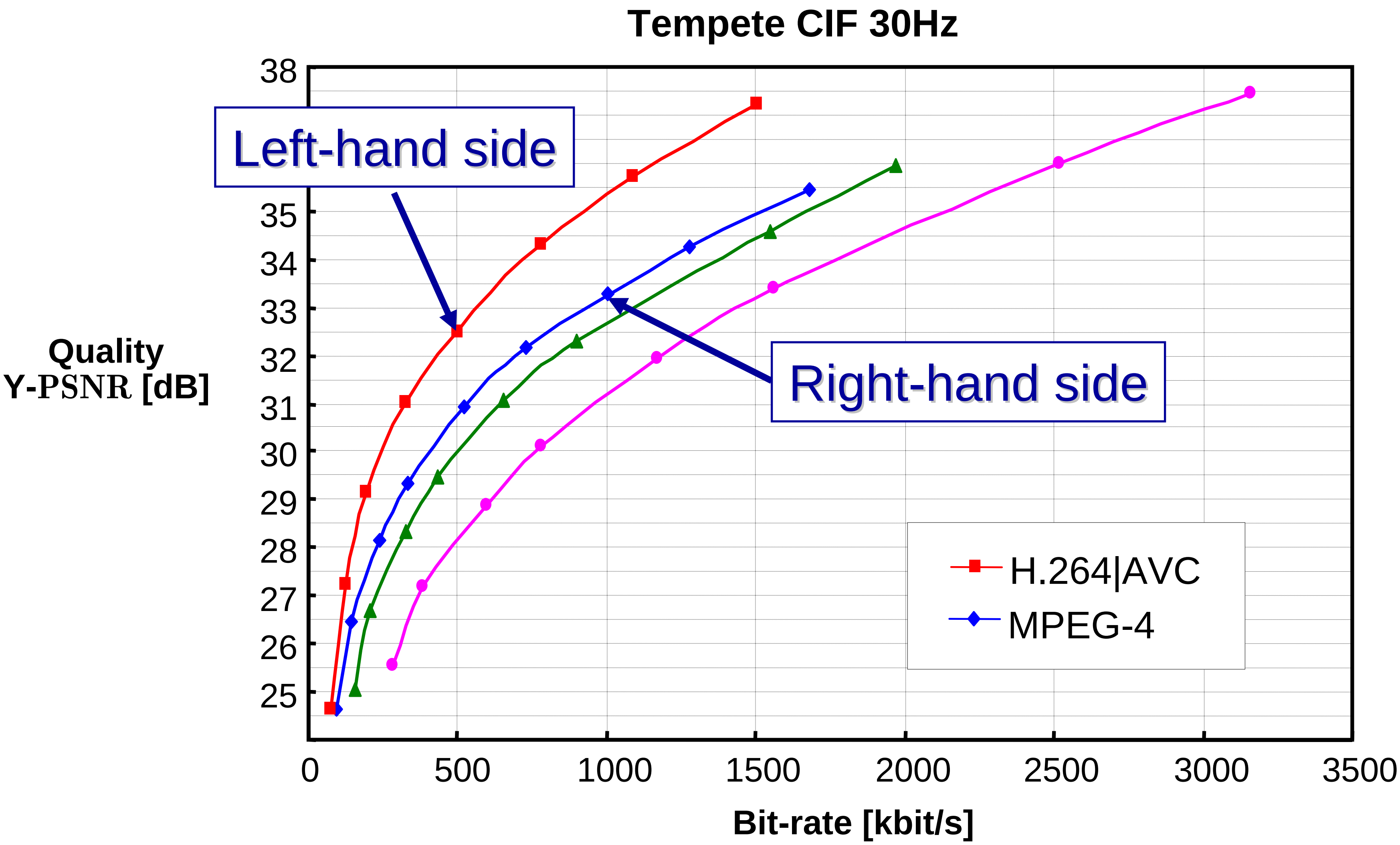
Example Streaming Test Result



Comparison to MPEG-4 ASP



Comparison to MPEG-2, H.263, MPEG-4



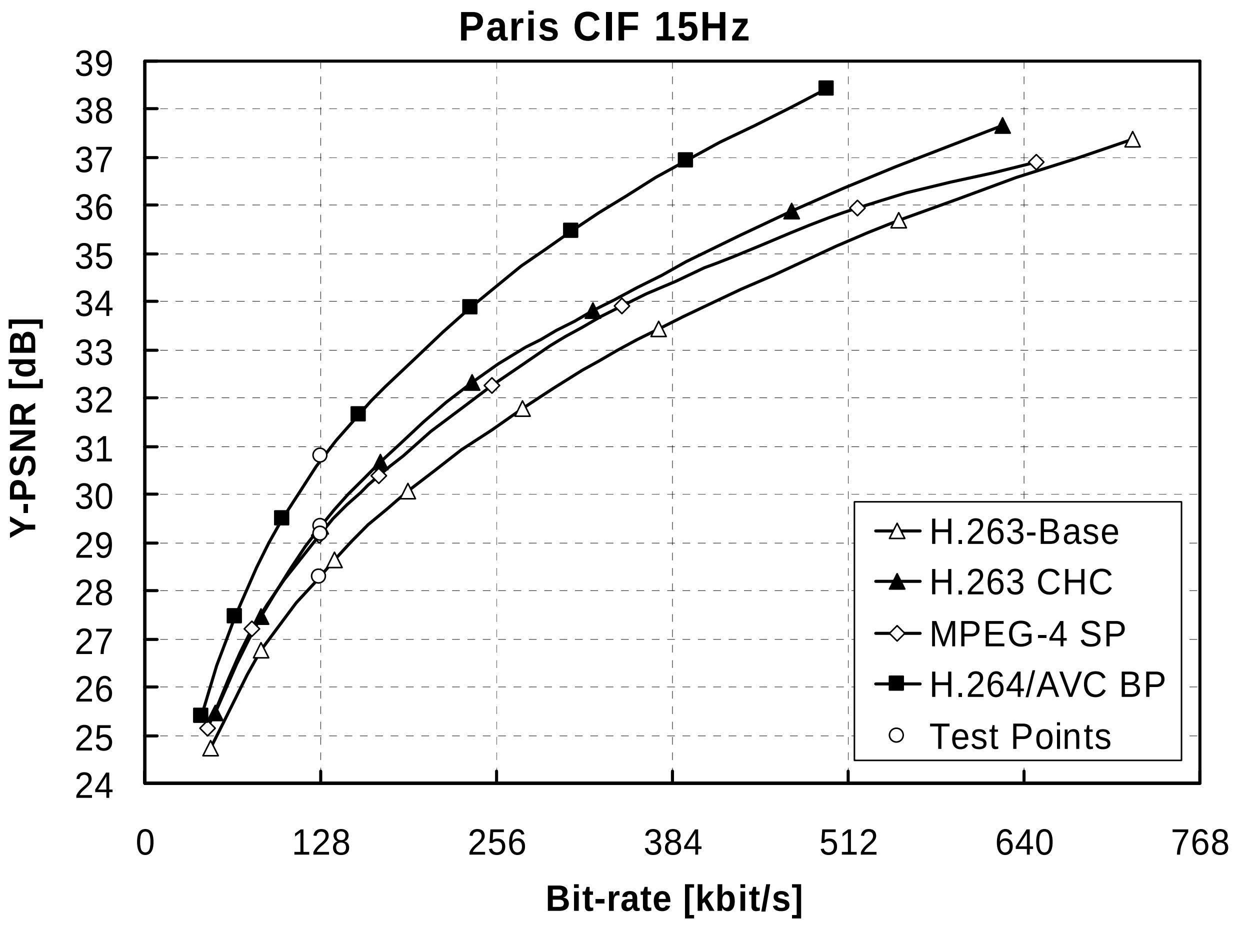
Test Set for Real-Time Conversation

Name	Resolution	Duration	Characteristics
Akiyo	QCIF	10 sec.	Still camera on human subject with synthetic background
Foreman	QCIF	10 sec.	Fast camera and content motion with pan at the end
Silent	QCIF	10 sec.	Still camera but fast moving subject
Mother & Daughter	QCIF	10 sec.	Still camera on human subjects
Carphone	CIF	10 sec.	Fast camera and content motion with landscape passing
Foreman	CIF	10 sec.	Fast camera and content motion with pan at the end
Paris	CIF	10 sec.	Still camera on human subjects; typical videoconferencing content
Sean	CIF	10 sec.	Still camera on human subject with synthetic background

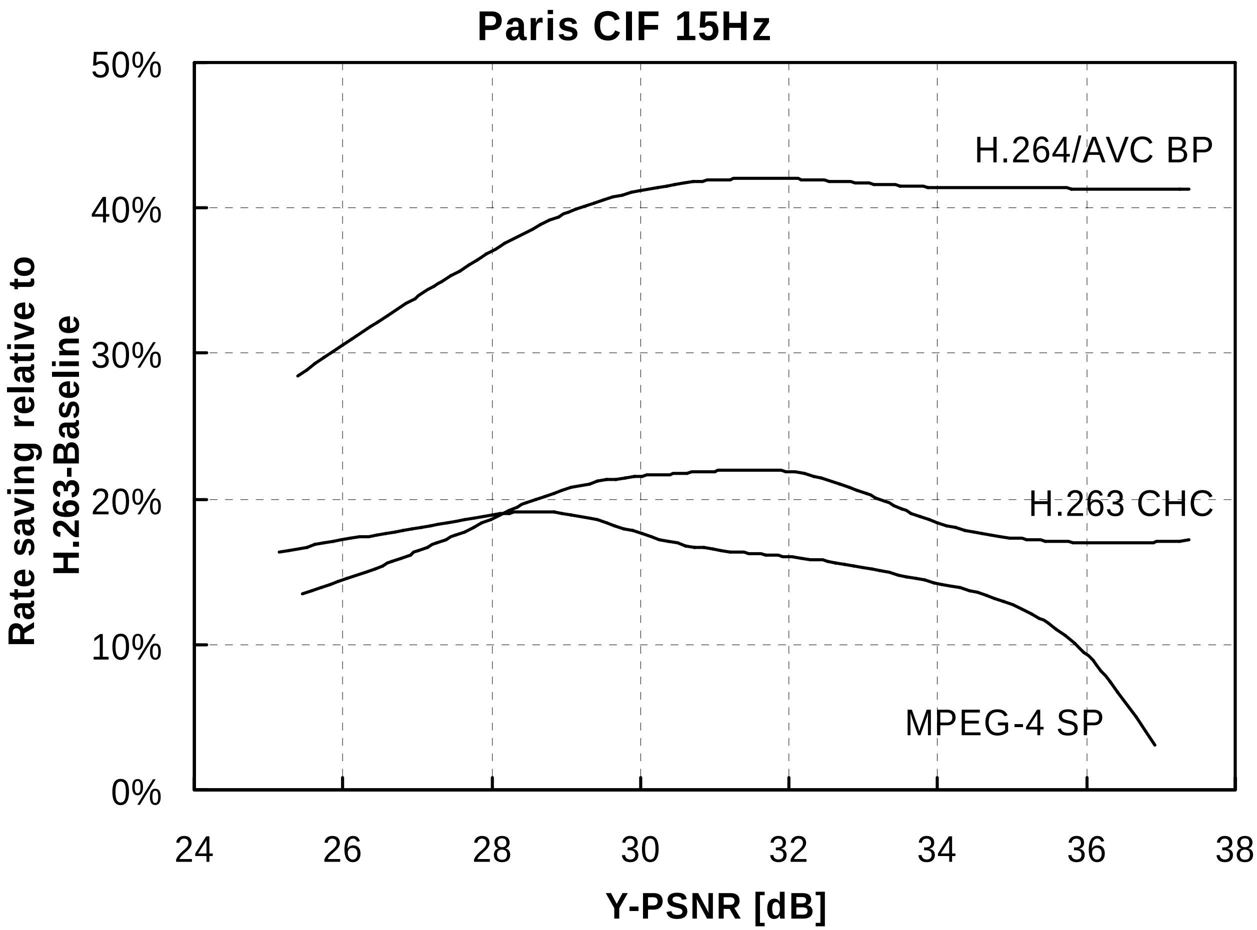
Test Results for Real-Time Conversation

	Average bit-rate savings relative to:		
Coder	H.263 CHC	MPEG-4 SP	H.263 Base
H.264/AVC BP	27.69%	29.37%	40.59%
H.263 CHC	-	2.04%	17.63%
MPEG-4 SP	-	-	15.69%

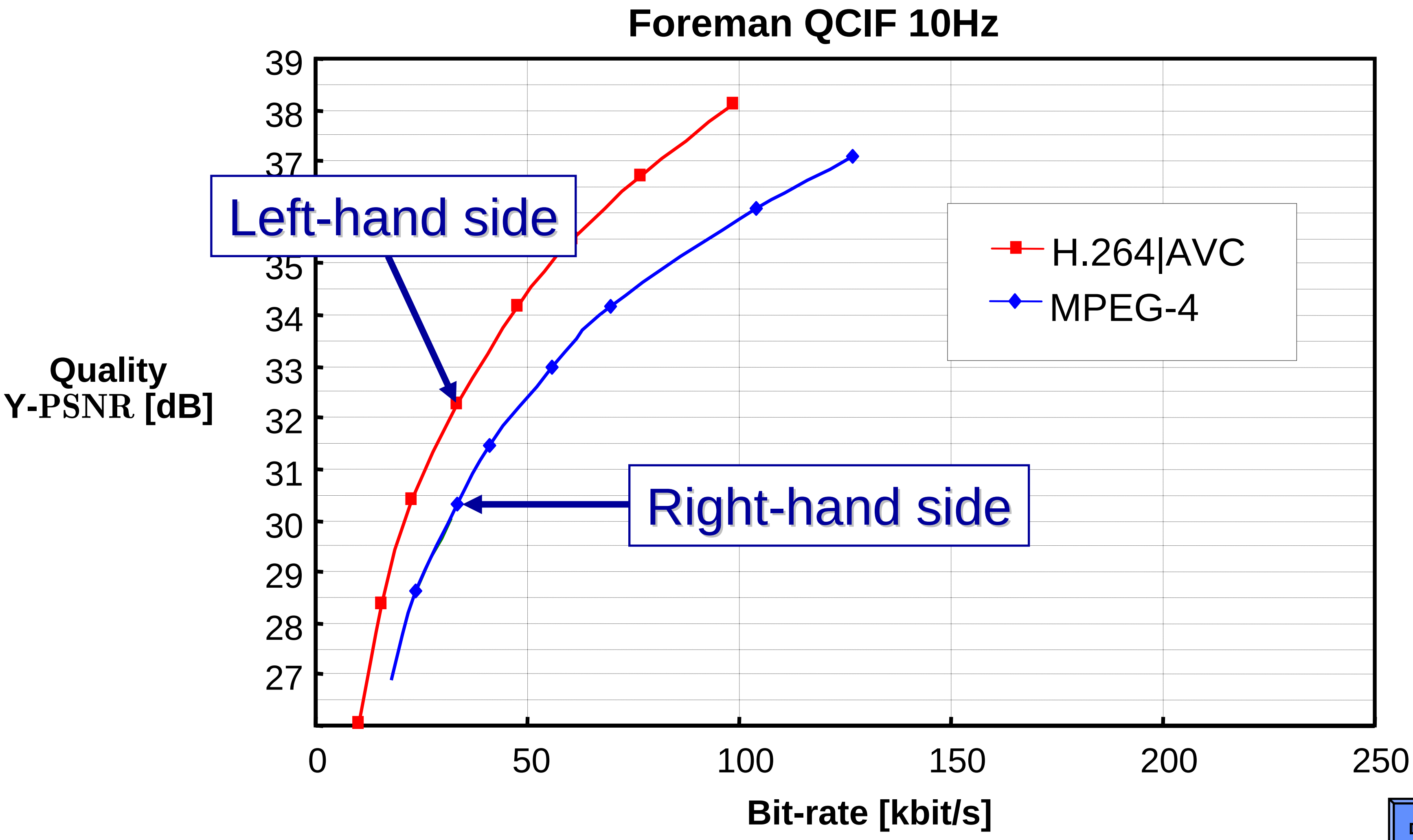
Example Real-Time Conversation Result



Example Real-Time Test Result



Comparison to MPEG-2, H.263, MPEG-4



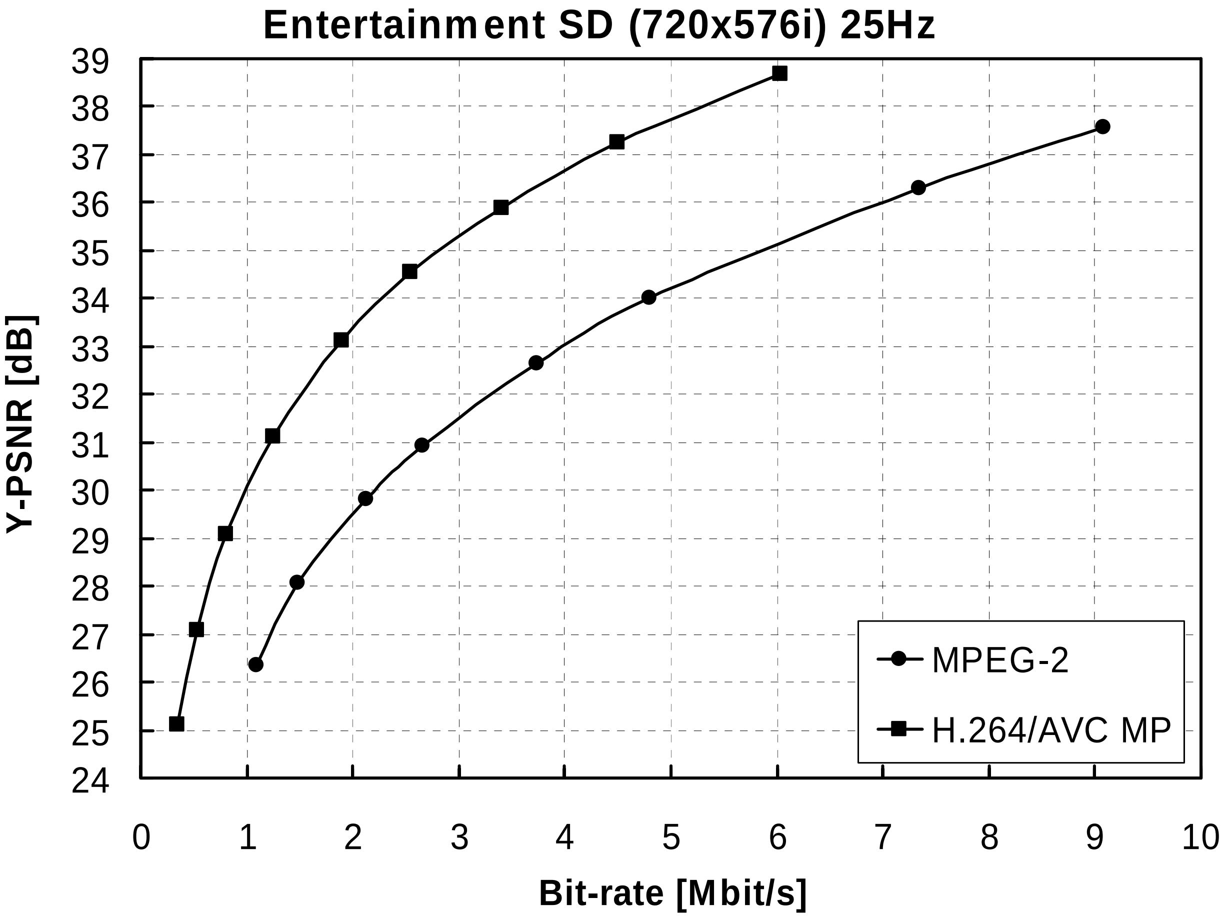
Test Set for Entertainment-Quality Applications

Name	Resolution	Duration	Characteristics
Harp & Piano	720×576i	8.8 sec.	Fast camera zoom; local motion
Basketball	720×576i	9.92 sec.	Fast camera and content motion; high spatial detail
Entertainment	720×576i	10 sec.	Camera and content motion; spatial detail
News	720×576i	10 sec.	Scene cut between slow and fast moving scene
Shuttle Start	1280×720p	10 sec.	Jiggling camera, low contrast, lighting change
Sailormen	1280×720p	10 sec.	Translational and random motion; high spatial detail
Night	1280×720p	7.67 sec.	Static camera, fast complex motion
Preakness	1280×720p	10 sec.	Camera zoom, highly complex motion, high spatial detail

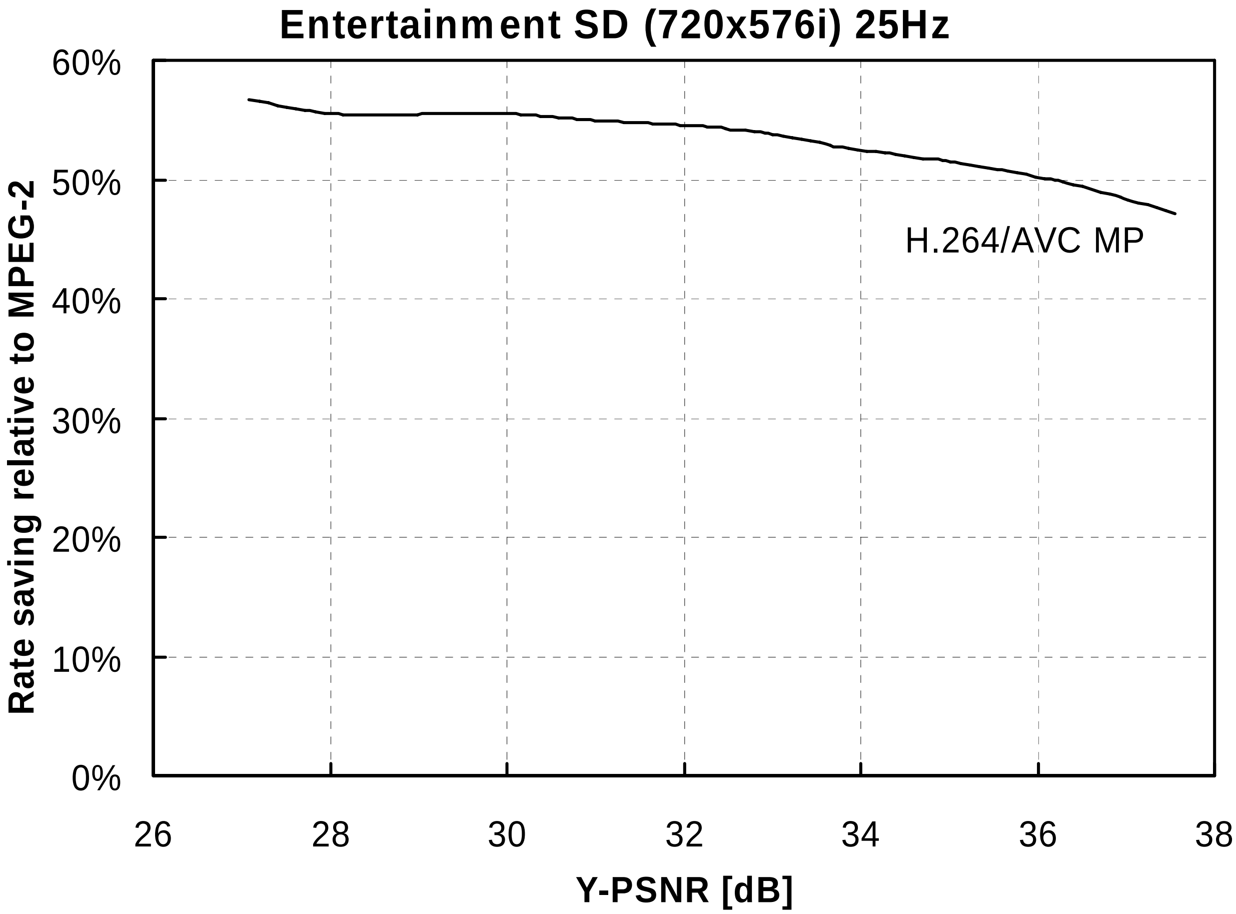
Test Results Entertainment-Quality Applications

	Average bit-rate savings relative to:
Coder	MPEG-2
H.264/AVC MP	45%

Example Entertainment-Quality Applications Result



Example Entertainment-Quality Applications Result



More Results ?

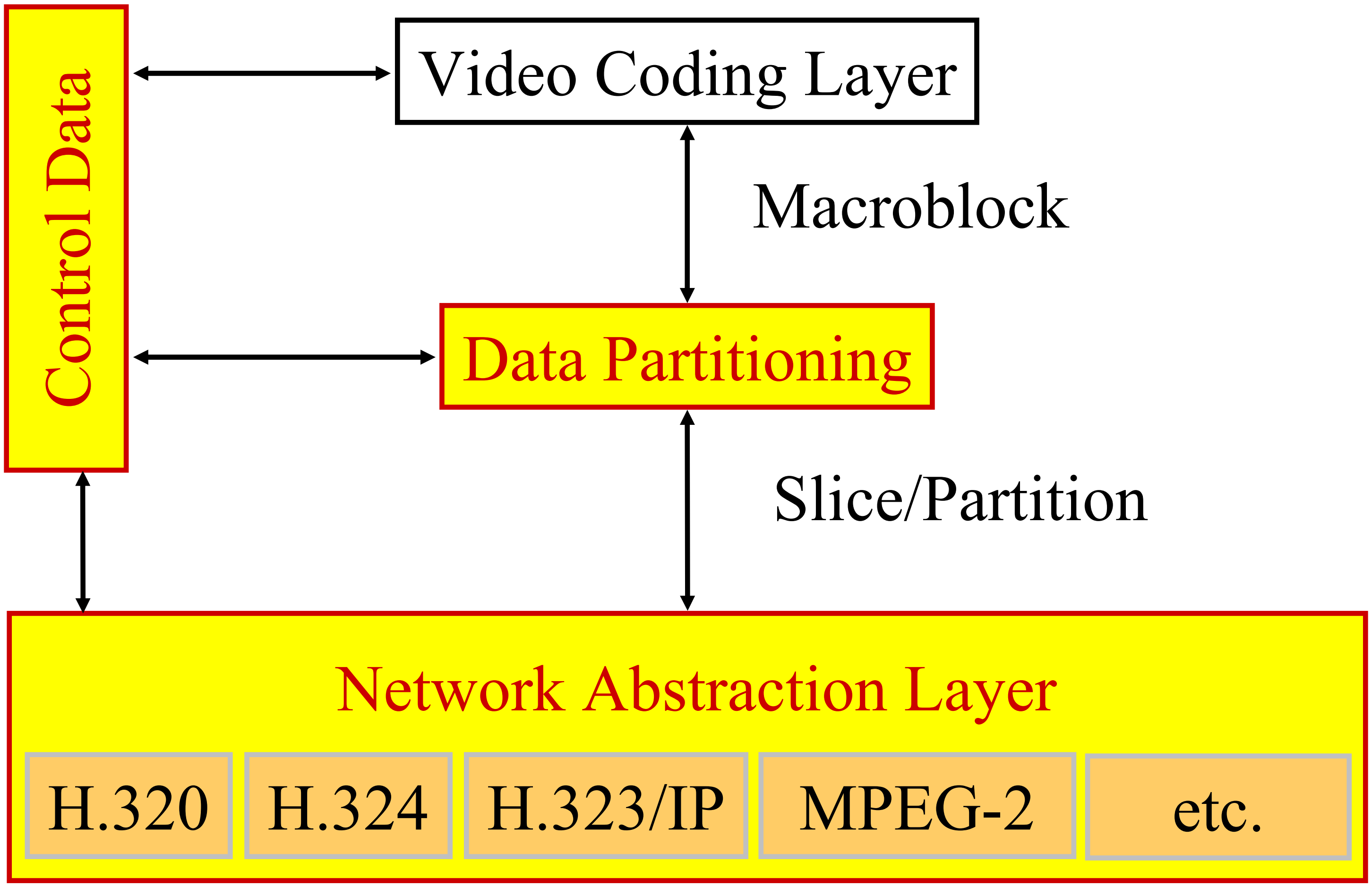
The various standard decoders together with bit-streams of all test cases presented in this paper can be down-loaded at

ftp://ftp.hhi.de/ieee-tcsvt/

More Details ?

T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan: "**Rate-Constrained Coder Control and Comparison of Video Coding Standards**," in *IEEE Transactions on Circuits and Systems for Video Technology*, July 2003.

H.264/AVC Layer Structure



Networks and Applications

- Broadcast over cable, satellite, DSL, terrestrial, etc.
- Interactive or serial storage on optical and magnetic devices, DVD, etc.
- Conversational services over ISDN, Ethernet, LAN, DSL Wireless Networks, modems, etc. or a mixture of several.
- Video-on-demand or multimedia streaming services over ISDN, DSL Ethernet, LAN, Wireless Networks, etc.
- Multimedia Messaging Services (MMS) over ISDN, DSL, Ethernet, LAN, Wireless Network, etc.
- New applications over existing and future networks!

How to handle this variety of applications and networks?

Network Abstraction Layer

Mapping of H.264/AVC video to transport layers like

- RTP/IP for any kind of real-time wireline and wireless Internet services (conversational and streaming)
- File formats, e.g. ISO MP4 for storage and MMS
- H.32X for wireline and wireless conversational services
- MPEG-2 systems for broadcasting services, etc.

Outside the scope the H.264/AVC standardization, but awareness!

Provision of appropriate mechanisms and interfaces

- Provide mapping to network and to facilitate gateway design
- Key Concepts: Parameter Sets, Network Abstraction Layer (NAL) Units, NAL unit and byte-stream formats

Completely within the scope of H.264/AVC standardization

Network Abstraction Layer (NAL) Units

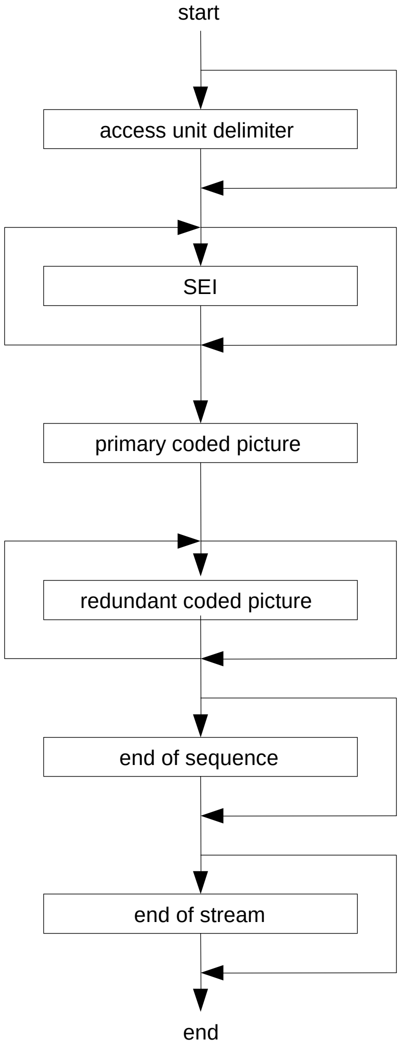
Constraints

- Many relevant networks are packet switched networks
- Mapping packets to streams is easier than vice versa
- Undetected bit-errors practically do not exist on the application layer

Architecture: NAL units as the transport entity

- NAL units may be mapped into a bit stream…
- … or forwarded directly by a packet network
- NAL units are self-contained (independently decodable)
- The decoding process assumes NAL units in decoding order
- The integrity of NAL units is signaled by the correct size (conveyed externally) and the *forbidden_bit* set to 0.

Access Units



NAL Unit Format and Types

NAL unit header	NAL unit payload

NAL unit header: 1 byte consisting of

- forbidden_bit (1 bit): may be used to signal that a NAL unit is corrupt (useful e.g. for decoders capable to handle bit errors)
- nal_storage_idc (2 bit): signals relative importance, and if the picture is stored in the reference picture buffer
- nal_unit_type (5 bit): signals 1 of 10 different NAL unit types
  - Coded slice (regular VCL data),
  - Coded data partition A, B, C (DPA, DPB, DPC),
  - Instantaneous decoder refresh (IDR),
  - Supplemental enhancement information (SEI),
  - Sequence and picture parameter set (SPS, PPS),
  - Picture delimiter (PD) and filler data (FD).

NAL unit payload: an emulation prevented sequence of bytes.

RTP Payload Format for H.264/AVC

- The specification of an RTP payload format is on the way within the IETF AVT
- The draft also follows the goals “back-to-basic” and simple syntax specification
- RTP payload specification expects that NAL units are transmitted directly as the RTP payload
- Additional concept of aggregation packets is introduced to aggregate more than one NAL unit into a single RTP packet (helpful for gateway designs between networks with different MTU size requirements)
- RTP time stamp matches presentation time stamp using a fixed 90 kHz clock
- Open Issue: media unaware fragmentation

Byte-stream Format for H.264/AVC

- Not all transport protocols are packet-based, e.g. MPEG-2 systems over S/C/T, H.320 over ISDN
- H.264/AVC standard defines a byte-stream format to transmit a sequence of NAL units as an ordered stream of bytes
- NAL unit boundaries need to be identified to obtain NAL units with correct size to guarantee integrity
- A byte-oriented HDLC-like framing including start codes (1or 2 bytes) and emulation prevention is specified
- For simplified gateway operation, the emulation prevention on byte basis is applied to all raw byte sequence payloads (RBSPs).

Byte Alignment, Emulation Prevention and Framing

Sequence of binary video data

Slice Boundary

.... 01000100000000000000000111010101010 | 10100101010101010 ....

Byte Alignment ⇒ Sequence of raw byte sequence payloads

...... 01000100|00000000|00000011|10101010|1010**1000** | 10100101|01010101|0 ....

...... 0100|01000000|00000000|00111010|10101010|**10000000** | 10100101|01010 ......

Emulation Prevention + *NAL unit header* ⇒ NAL unit

...... 0x44 0x00 0x01 0xAA 0xA8 | 0xA5 0x55 0x00 0x00 0x02 |

...... 0x44 0x00 **0x03** 0x01 0xAA 0xA8 ***0x21*** 0xA5 0x55 0x00 **0x03** 0x00 **0x03** 0x02 ......

Framing only for Byte Stream Format according to Annex B

.... 0x44 0x00 0x03 0x01 0xAA 0xA8 **0x00 0x01** 0x21 0xA5 0x55 0x00 0x03 0x00 0x03 0x02 ......

Access Unit Delimiter

- Observation: No Picture Header and no Picture Type
  - No need for either in many applications
  - Their existence harms the performance in some applications
- But: some applications need a picture type
  - Primarily Storage Applications, for trick modes
- Hence: Introduction of the access unit delimiter
  - Optional tool
  - Signals the picture type and whether the picture is stored in the reference frame buffer
  - Inserted before the first NAL unit of a picture in decoding order, hence signals implicitly the boundary between pictures

Data Partitioning NAL Units 1/2

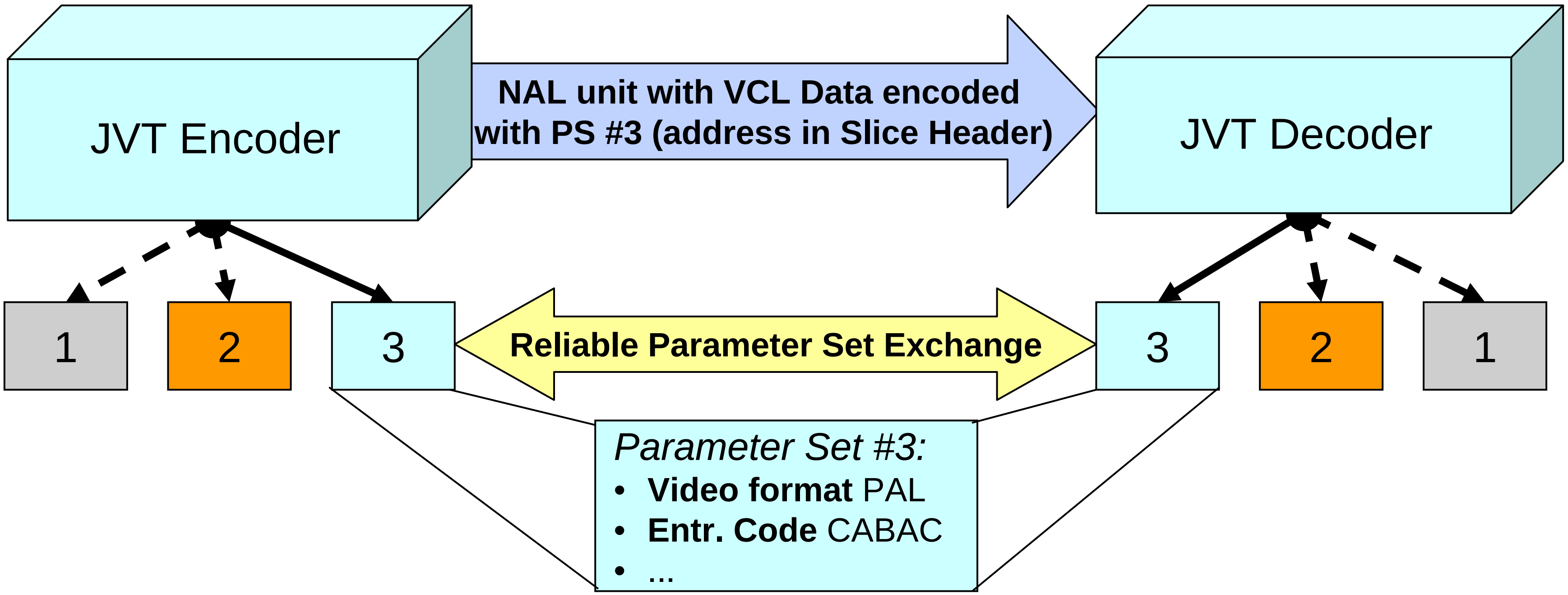
- H.264 | AVC contains Data Partitioning w/ 3 Partitions
  - Data partition A (DPA) contains header info
    - Slice header
    - All macroblock header information
    - Motion vectors
  - Data partition B (DPB) contains intra texture info
    - Intra CBPs
    - Intra coefficients
  - Data partition C (DPC) contains inter texture info
    - Inter CBPs
    - Inter Coefficients
- When DP is used, all partitions are in separate NAL units

Data Partitioning NAL Units 2/2

- Properties of the Partition Types
  - DPA is (perceptually) more important than DPB
  - DPB cleans up error propagation, DPC does not
- Transport DPA w/ higher QoS than DPB, DPC
  - In lossy transmission environments typically leads to overall higher reconstructed picture quality *at the same bit rate*
  - Most packet networks contain some prioritization
    - Sub-Transport and Transport level, e.g. in 3GPP networks or when using DiffServ in IP
    - Application Layer protection
      - Packet Duplication
      - Packet-based FEC

Parameter Set Concept

- Sequence, random access, picture headers can get lost
- Solutions in previous standards: duplication of headers
- H.264/AVC coding applies a new concept: parameter sets



Parameter Set Discussion

- Parameter Set: Information relevant to more than one slice
  - Information traditionally found in sequence / picture header
  - Most of this information is static, hence transmission of a reference is sufficient
  - Problem: picture-dynamic info, namely timing (TR)
  - Solution: picture-dynamic info in every slice
    - Overhead is smaller than one would expect
- Parameter Sets are conveyed out-of-band and reliable
  - No corruption/synchronization problems
  - Aligned with closed control application
  - Need in-band transmission mechanism for broadcast

Nested Parameter Sets

- Each slice references a picture parameter set (PPS) to be used for decoding its VCL data:
  - PPS selected by short variable length codeword transported in slice header
  - Contains, e.g. entropy coding mode, FMO parameters, quantization initialization, weighted prediction indications, etc.
  - PPS reference can change between pictures
- Each PPS references a sequence parameter set (SPS)
  - SPS is referenced only in the PPS
  - Contains, e.g. profile/level indication, display parameters, timing concept issues, etc.
  - SPS reference can change only on IDR pictures

Establishment and Updates of Parameter Sets

- If possible, SPS and PPS should be established and updated reliably and out-of-band
  - Typically established during capability exchange (SIP, SDP, H.245) or in session announcement,
  - Updates also possible by control protocols,
  - SPS and PPS could be pre-defined, e.g. in multicast or broadcast applications
- Special NAL unit types are specified to setup and change SPS and PPS in-band
  - Intended ONLY for those applications where no control protocol is available
  - Allows to have self-contained byte-streams
  - Use of in-band and out-of-band Parameter Set transmission mutually exclusive (to avoid sync problems)

Supplemental Enhancement Information (SEI)

- Supplemental Enhancement information NAL unit contains synchronously delivered information that is not necessary to decode VCL data correctly
- SEI is helpful for practical decoding or presentation purpose
- An SEI message is associated with the next slice or data partitioning RBSP in decoding order
- Examples are
  - Display information, absolute timing, etc.
  - Scene transition information (fades, dissolve, etc.)
  - Control info for videoconferencing (e.g. FPR)
  - Error resilience issues, e.g. repetition of reference picture buffer management information
  - Arbitrary user data, etc.

Summarizing NAL

- In H.264/AVC, the transport of video has been taken into account from the very beginning
- Flexibility for integration to different transport protocols is provided
- Common structure based on NAL units and parameter sets is maintained for simple gateway operations
- Mapping to MPEG-2 transport stream is provided via byte-stream format
- On the way are payload specification to different transport protocols, e.g. to RTP/IP

Grouping of Capabilities into Profiles

- Three profiles now: **Baseline**, **Main**, and **Extended**
- **Baseline** (e.g., Videoconferencing & Wireless)
  - I and P picture types (not B)
  - In-loop deblocking filter
  - 1/4-sample motion compensation
  - Tree-structured motion segmentation down to 4x4 block size
  - VLC-based entropy coding (CAVLC)
  - Some enhanced error resilience features
    - Flexible macroblock ordering/arbitrary slice ordering
    - Redundant slices
  - **Note**: No support for interlaced video in Baseline

Non-Baseline Profiles

- **Main** Profile (esp. Broadcast/Entertainment)
  - All Baseline features *except enhanced error resilience features*
  - B pictures
  - Adaptive weighting for B and P picture prediction
  - Picture and MB-level frame/field switching
  - *CABAC*
  - **Note**: Main is not exactly a superset of Baseline
- **Extended** Profile (esp. Streaming/Internet)
  - All Baseline features
  - B pictures
  - Adaptive weighting for B and P picture prediction
  - Picture and MB-level frame/field switching
  - *More error resilience: Data partitioning*
  - *SP/SI switching pictures*
  - **Note**: Extended *is* a superset of Baseline (but not of Main)

Complexity of Codec Design

- Codec design includes relaxation of traditional bounds on complexity (memory & computation) – rough guess 3x decoding power relative to MPEG-2, 4x encoding
- Problem areas:
  - Smaller block sizes for motion compensation (cache access issues)
  - Longer filters for motion compensation (more memory access)
  - Multi-frame motion compensation (more memory for reference frame storage)
  - More segmentations of macroblock to choose from (more searching in the encoder)
  - More methods of predicting intra data (more searching)
  - Arithmetic coding (adaptivity, computation on output bits)

Implementations: The Early Reports

- **UB Video** (JVT-C148) CIF resolution on 800 MHz laptop
  - Encode: 49 fps
  - Decode: 137 fps
  - Encode+Decode: 36 fps
  - Better quality than R-D optimized H.263+ Profile 3 (IJKT) while using 25% higher rate and low-delay rate control
- **Videolocus/LSI** (JVT-D023) SDTV resolution
  - 30 fps encode on P4 2 GHz with hardware assist
  - Decode on P3 1 GHz laptop (no hardware assist)
  - No B frames, no CABAC (approx baseline)
- **Tandberg** Videoconferencing (http://tinyurl.com/k4lp)
  - All Tandberg end-points ship with H.264/AVC since July 14, '03
- **Reference software** (super slow)
- Others: **HHI, Deutsche Telekom, Broadcom, Nokia, Motorola, &c**
- Caution: These are preliminary implementation reports only – mostly involving incomplete implementations of non-final draft designs

Companies Publicly Known to be Doing Preliminary Implementation Work

- Amphion
- British Telecom
- Broadcom (chip)
- Conexant (chipset for STB)
- DemoGraFX (with bit precision extension)
- Deutsche Telekom
- Envivio
- Equator
- Harmonic (filtering and motion estimation)
- HHI (PC & DSP encode & decode; demos)
- iVast
- LSI Logic (chip, plus Videolocus acquisition demoing real-time FPGA+P4 encode, P4 dec)
- Mainconcept
- Mobile Video Imaging
- Modulus Video
- Moonlight Cordless
- Motorola
- Nokia
- PixelTools
- PixSil Technology
- Polycom (videoconferencing & MCUs)
- Sand Video (demoed 2 Xilinx FPGA decoder, encode/decode & decode-only chips to fab in '03)
- Sony (encode & decode, software & hardware, including PlayStation Portable 2004 & videoconferencing systems)
- ST Micro (decoder chip in '03)
- Tandberg (videoconferencing – shipping in all end points and as software upgrade)
- Thomson
- TI (DSP partner with UBV for one of two UBV real-time implemenations)
- Toshiba
- UB Video (demoed real-time encode and decode, software and DSP implementations)
- Vanguard Software Solutions (s/w, enc/dec)
- VCON

CAUTION: All such information should be considered preliminary and should not be considered to be product announcements – only preliminary implementation work. It will be awhile before robust interoperable conforming implementations exist.

Product Plans Announced

- Amphion http://www.eetimes.com/story/OEG20020920S0049
- DemoGraFX http://www.demografx.com/products/
- Envivio http://www.envivio.com/news/news/021121_h264.html
- Equator http://www.embeddedstar.com/press/content/2002/10/embedded5816.html
- Envivio http://biz.yahoo.com/prnews/030407/sfm088_1.html
- HHI http://www.eetimes.com/sys/news/OEG20020916S0072
- IVast http://www.ivast.com/company/press/2003/SandVid_NAB_033103.pdf
- LSI Logic / Videolocus (evaluation platform) http://www.videolocus.com/products/product.htm
- Mainconcept http://www.mainconcept.com/h264.shtml
- Mobile Video Imaging http://www.digitalwebcast.com/2003/03_mar/news/dlmvi32703.htm
- Modulus Video http://www.modulusvideo.com/
- Moonlight Cordless http://www.prweb.com/releases/2003/3/prweb59692.php
- PixelTools http://www.pixeltools.com/experth264.html
- PixSil Tech http://www.pixsiltech.com/products.htm
- Polycom (videoconferencing & MCUs) http://www.polycom.com/investor_relations/0,1406,pw-2573,FF.html
- Sand Video http://www.sandvideo.com/pressroom.html
- Sony http://www.eetimes.com/issue/mn/OEG20030801S0024 & http://news.sel.sony.com/pressrelease/3691
- ST Microelectronics http://www.eetuk.com/tech/news/OEG20021113S0026
- Tandberg http://tandberg.net/tb.asp?s=pagesimple&aid={8395730F-6D6F-4101-812F-B10A37412E16}
- UB Video http://www.eetimes.com/semi/news/OEG20021202S0048
- Vanguard Software Solutions (software encode & decode) http://www.vsofts.com/codec/h264.html
- VCON http://www.vcon.com/press_room/english/2003/03031102.shtml

Conclusions

- Video coding layer is based on hybrid video coding and similar in spirit to other standards but with important differences
- New key features are:
  - Enhanced motion compensation
  - Small blocks for transform coding
  - Improved deblocking filter
  - Enhanced entropy coding
- Bit-rate savings around 50 % against any other standard for the same perceptual quality (especially for higher-latency applications allowing B pictures)
- Standard of both ITU-T VCEG and ISO/IEC MPEG

Resources

- Anonymous ftp site and documents: **ftp.imtc-files.org** (directory jvt-experts)
- H.264 / MPEG-4 AVC FDIS text on ftp site
- Reference software: http://bs.hhi.de/~suehring
- E-Mail reflectors for experts group
- Special Issue in IEEE Transactions on Circuits and Systems for Video Technology, July 2003