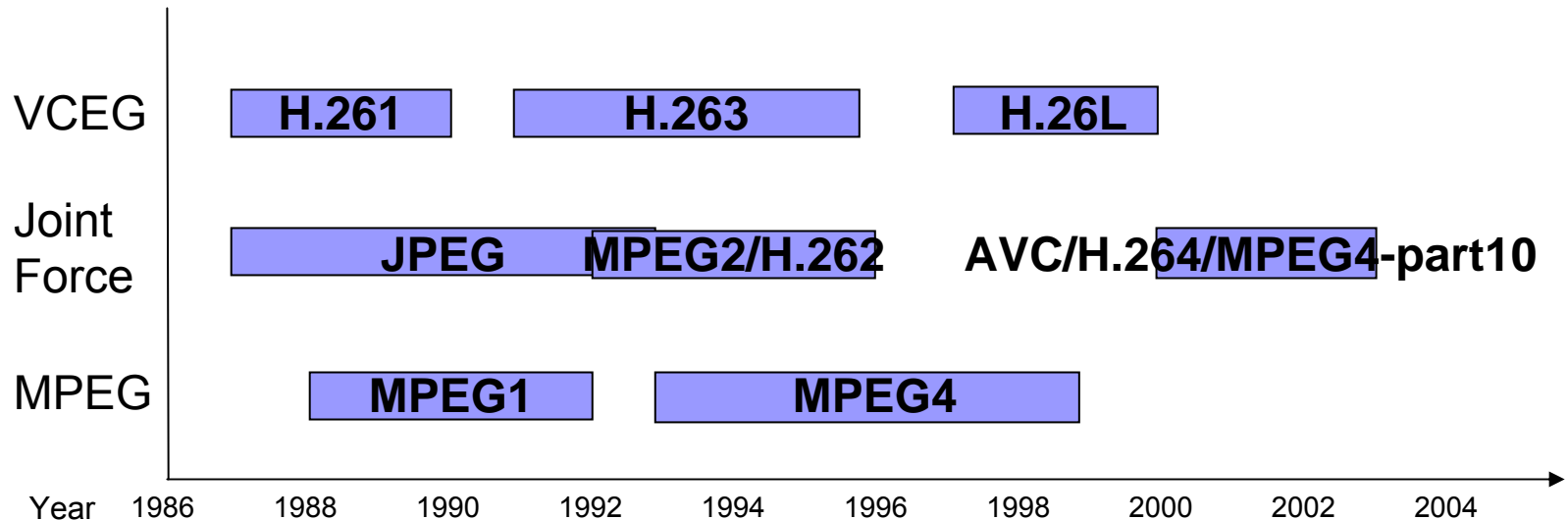


# Video Coding Standards: Up to H.264

Jing Hu and Jerry D. Gibson  
May 2007

# Time table



- Two organizations dominate video compression standardization:
  - ITU-T Video Coding Experts Group (VCEG): International Telecommunications Union – Telecommunications Standardization Sector (ITU-T, a United Nations Organization, formerly CCITT), Study Group 16, Question 6
  - ISO/IEC Moving Picture Experts Group (MPEG): International Standardization Organization and International Electro-technical Commission, Joint Technical Committee Number 1, Subcommittee 29, Working Group 11

# Joint(ITU+ISO) Photographic Experts Group (JPEG)

## ■ JPEG targets:

8 bits/pixel monochrome images	→	0.083 bits/pixel as	“recognizable”
		0.25	“useful”
		0.75	“excellent”
		2.25	“indistinguishable”

## ■ Color treatments:

Red	→	Luminance (Y)	<b>Two modes:</b>
Green		Color difference B-Y ( $C_B$ )	4:2:2
Blue		Color difference R-Y ( $C_R$ )	4:2:0

## ■ JPEG coder

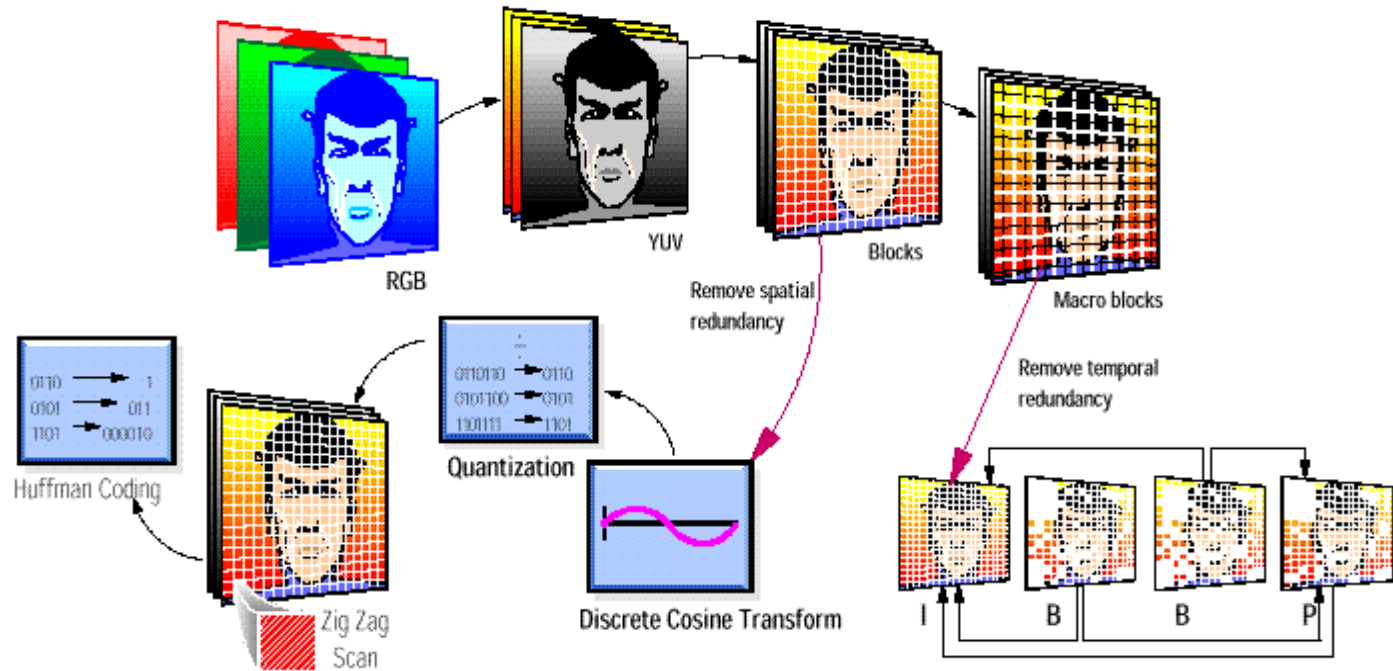
- 8×8 DCT (why DCT?)
- Quantization (Two tables by Lohscheller 1984)
- Zig-zag scanning and run-level description
- Entropy coding (Huffman and arithmetic coding)

## ■ Motion JPEG (Video coded as sequences of JPEG images)

# Motion JPEG

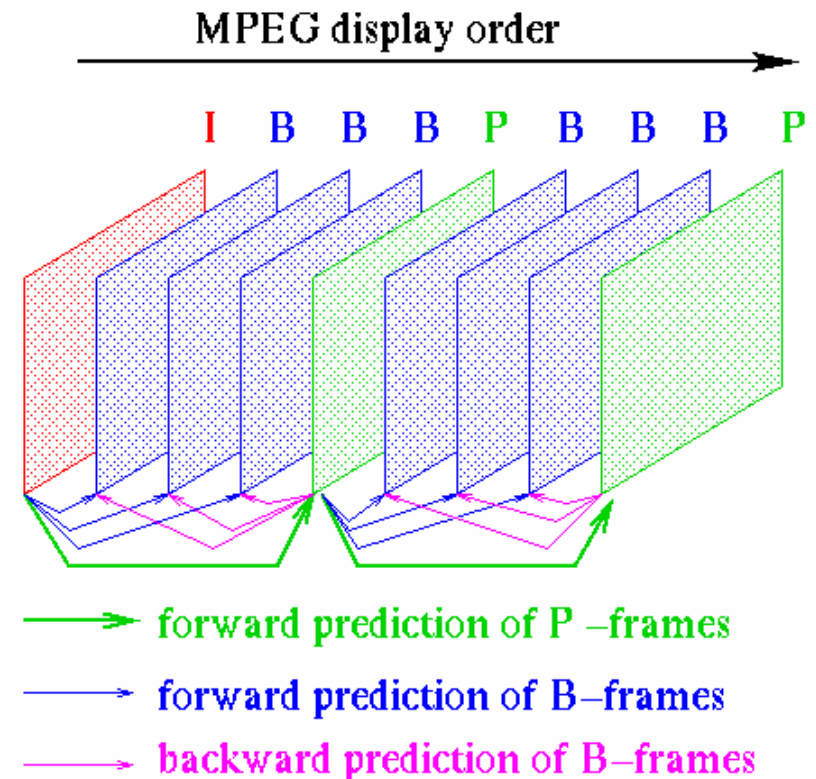
- Disadvantages
  - Loss of temporal compression capability
  - There is no document that defines a single exact format that is universally recognized as a complete specification of "Motion JPEG" for use in all contexts
- Advantages
  - Low processor overhead
  - Easy editing
- Applications
  - Digital cameras
  - Surveillance cameras

# MPEG – Generic Encoder



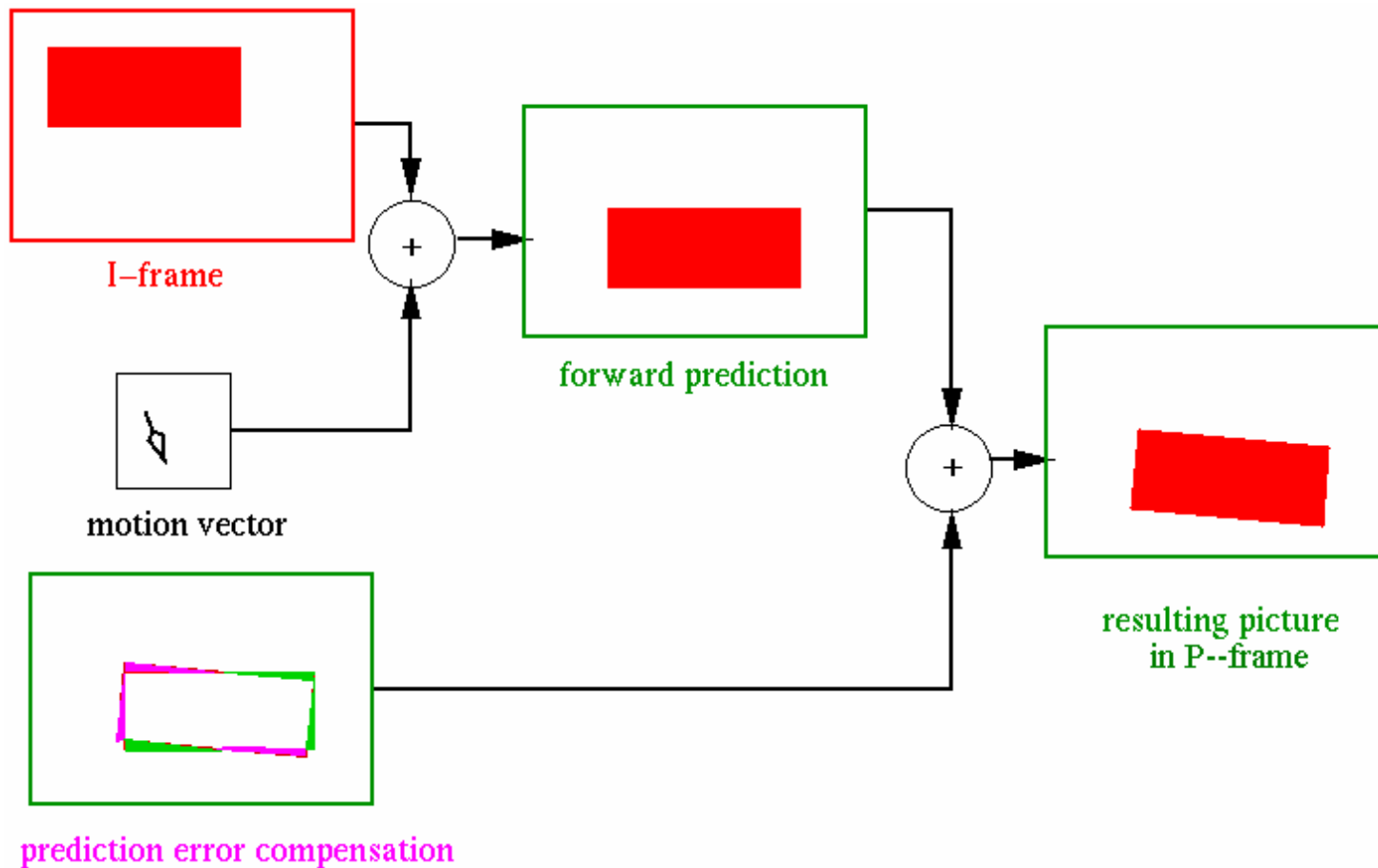
# MPEG-1 = JPEG + Motion Prediction + Rate Control

- Early motivation: to encode motion video at 1.5Mbits/s for transport over T1 data circuits and for replay from CD-ROM
- Defines the decoder but not the encoder
- Frames (pictures)
  - Intra-coded using JPEG
  - Inter-coded using (interpolated) motion estimation & compensation and JPEG for the residuals
    - Predicted and Bi-directional
- MacroBlocks (MBs)
  - 16×16 pixels block
- Rate control
  - buffer at each end
  - Test Model 5 (TM5)



# MPEG-1 – Motion Prediction

- Motion prediction = motion estimation + error compensation

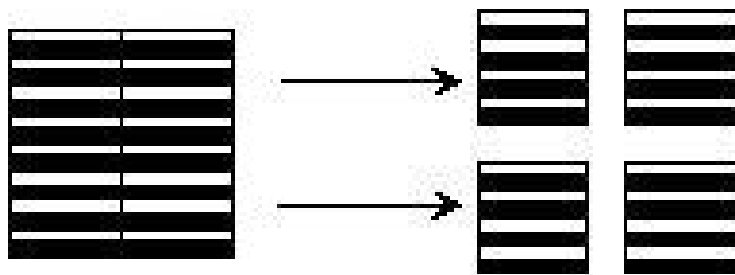
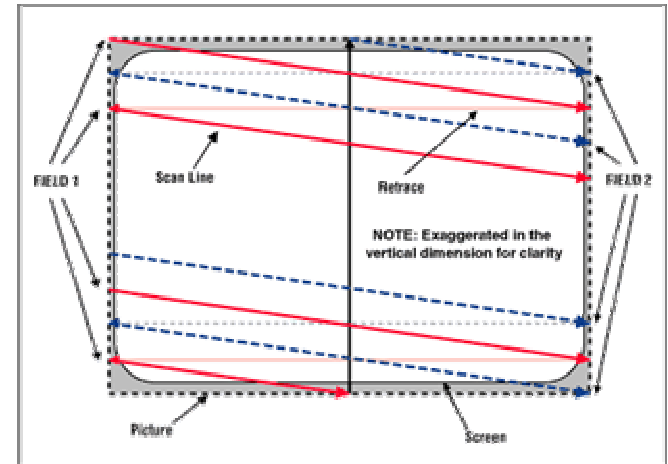


# MPEG-2 = MPEG-1 +

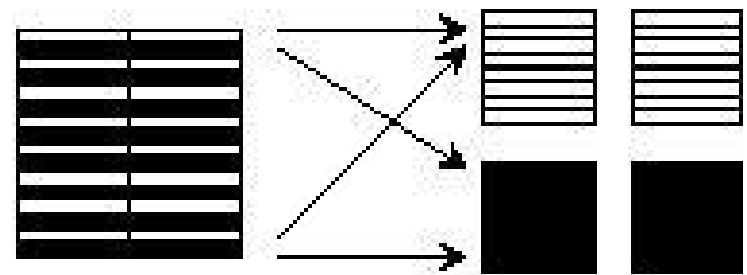
- Improvements
  - Color space: could support 4:2:2 and 4:4:4 coding
  - Quantization: could have 9- or 10- bit precision for DC coefficients
  - Concealment motion vectors: used when an intra-MB is lost
  - Pan and Scan: supports display of different aspect ratios, e.g., 16:9
- Profiles and levels
  - Profiles: define the tools or syntactical elements
  - Levels: define the permissible ranges of parameters
- Interlace tools
- Scalable coding profiles
- System layer: define two bit stream constructs
  - Program stream (PS): modeled on MPEG-1 (backward compatibility)
  - Transport stream (TS): more robust, does not need a common time base, designed for use in error-prone environment.

# MPEG-2 – Interlace Tools

- Interlaced Scanning: Image flicker is less apparent because the image is painted twice as many times as what is in non-interlaced scanning.
- Frame Pictures and Field Pictures
  - two fields are processed sequentially or not
- Frame DCT and Field DCT
  - Field pictures usually use field DCT
  - Frame pictures use field DCT when there is obvious vertical motion
- Frame Prediction and Field Prediction



Frame DCT



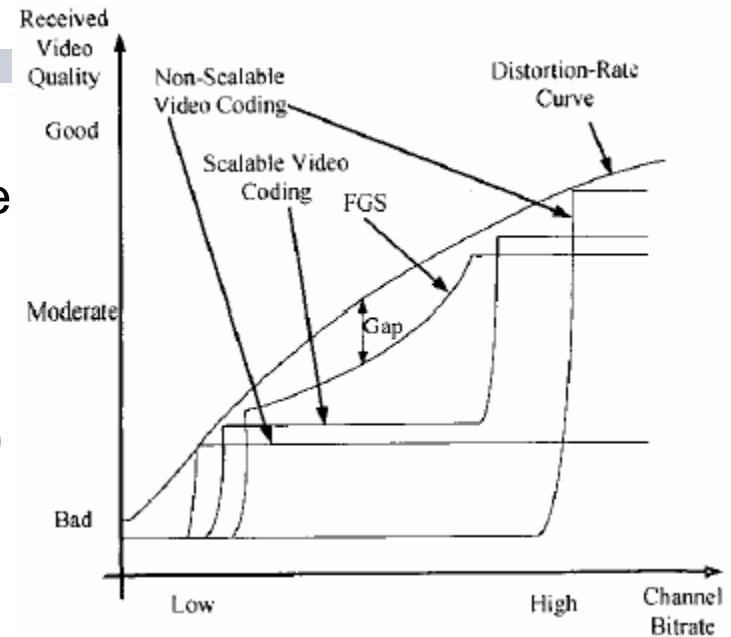
Field DCT

# MPEG-4 = MPEG-2+Objects+Other Enhancements

- Objects (optional)
  - Video (texture+shape), image, audio, speech, text, etc.
  - Encoded using different techniques
  - Transmitted independently
  - Mixed at the decoder using Binary Format for Scenes (BIFS)
- Improvements in MPEG-4 version2
  - Global motion compensation (GMC)
    - Compensating the camera motion: panning, zooming, rotation
  - Quarter pixel motion compensation
  - Shape-adaptive DCT
- Why is MPEG-4 not a success as MPEG-2?
  - Not substantially better than MPEG-2
  - Suffers from its sheer size and flexibility
  - Issue of licensing

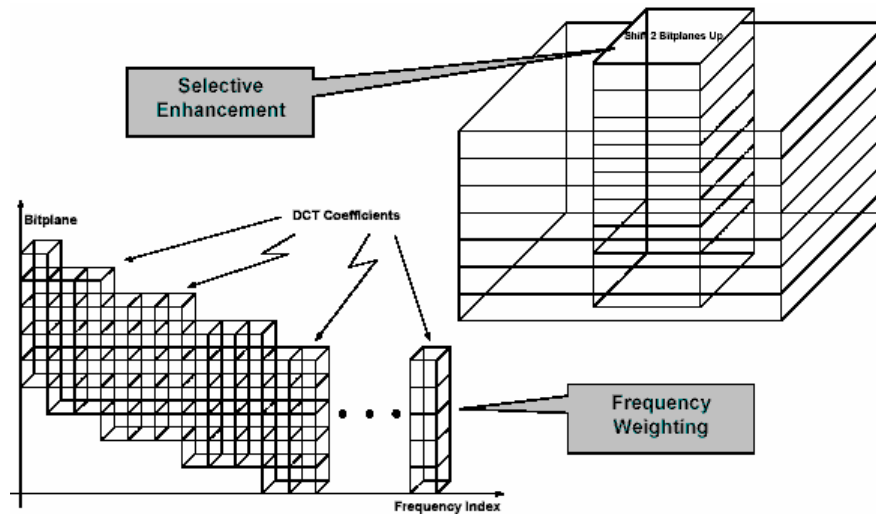
# MPEG – Scalable Coding (SC)

- Non-scalable coding
  - Optimize video quality at a given bitrate
- MPEG-2 SC profiles
  - To optimize video quality at two given bit rates.
  - SNR SC (different quantization accuracy)
  - Temporal SC (different frame rates)
  - Spatial SC (different spatial resolution)
- Fine granularity scalability (FGS)
  - To optimize the video quality over a given bit rate range
  - Also has base layer and enhancement layer
  - Enhancement layer uses bit-plane coding
    - 10,0,6,0,0,3,0,2,2,0,0,2,0,0,1,0,.....,0,0 (absolute)
    - 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,.....,0,0 (MSB)
    - 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,.....,0,0 (MSB-1)
    - 1,0,1,0,0,1,0,1,1,0,0,1,0,0,0,0,.....,0,0 (MSB-2)
    - 0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,.....,0,0 (MSB-3)



# Fine granularity scalability (FGS)

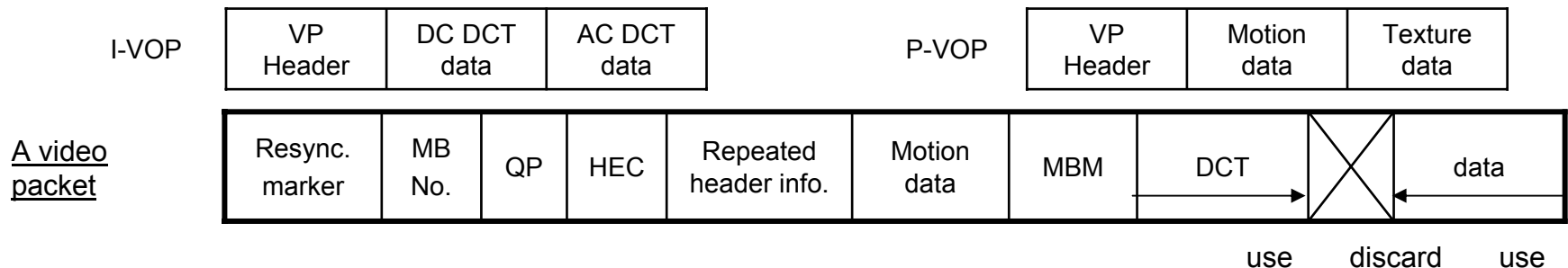
- Frequency weighting and selective enhancement



- The coding efficiency of the FGS scheme is not as good as the traditional SNR scalability schemes
- Progressive FGS
  - Using as many predictions from the enhancement reference layers as possible (for coding efficiency)
  - Keeping a prediction path from the base layer to the highest quality layers across several frames (for error recovery and channel adaptation)

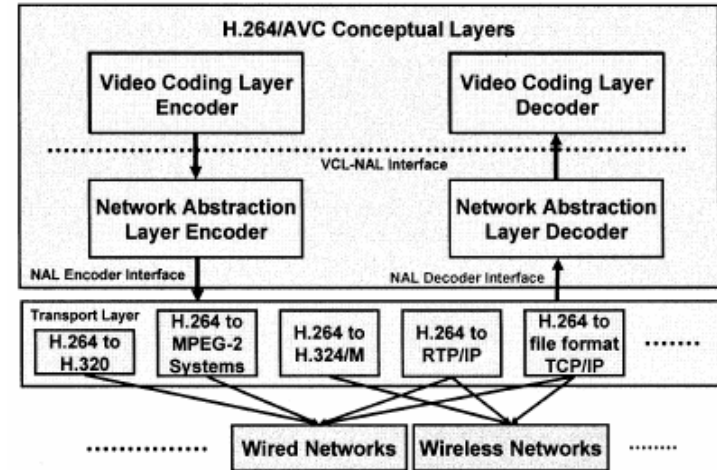
# MPEG-4 – Error Resilience Tools

- Video packet resynchronization
  - Previous coding standards: Resynchronization markers are fixed at the beginning of each row of MBs
  - MPEG-4: Resynchronization markers are inserted at every K bits
- Data partitioning
  - Partitions the data in a video packet into a motion part and a texture part separated by a motion boundary marker (MBM)
- Reversible variable length codes (RVLC)
  - Finds the next resynchronization marker and decode backwards
- Header extension code (HEC)
  - The header information is repeated after the 1-bit HEC
- Unequal error protection technique (UEP)



# Advanced Video Coding/ ITU-T Recommendation H.264/ ISO/IEC MPEG-4 (Part 10)

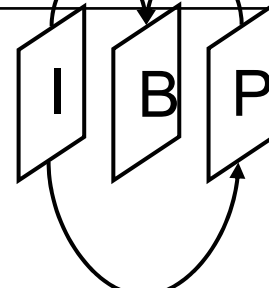
- H.264 structure
  - Video coding layer (VCL)
  - Network abstraction layer (NAL)
- Possible applications of H.264
  - Conversational services operated below 1Mbps with low latency.
    - ISDN-based H.320
    - H.324/M in circuit-switched channels
    - H.323 in packet-switched networks
  - Entertainment services operated between 1-8+ Mbps with moderate latency such as 0.5-2s in modified MPEG-2/H.222.0 systems.
    - Broadcast via satellite, cable, terrestrial or DSL
    - DVD for standard and high-definition video
    - Video-on-demand via various channels
  - Streaming services operated at 50-1500kbps with 2s or more of latency.



# Intra-Coded Macroblocks

	H.264	MPEG-1/2/4, H.261/3
Prediction in space domain	<ul style="list-style-type: none"> <li>■ Spatial prediction</li> <li>■ Encode the prediction modes (Use predictive coding if 4x4 modes are used)</li> </ul>	<ul style="list-style-type: none"> <li>■ No spatial prediction</li> </ul>
Transform	<ul style="list-style-type: none"> <li>■ Integer transform of residue</li> </ul>	<ul style="list-style-type: none"> <li>■ 8x8 Discrete Cosine Transform (DCT) for pixel values</li> </ul>
Quantization	<ul style="list-style-type: none"> <li>■ Quantization including scaling</li> </ul>	<ul style="list-style-type: none"> <li>■ Quantization</li> </ul>
Prediction in frequency domain	<ul style="list-style-type: none"> <li>■ No coefficient prediction</li> </ul>	<ul style="list-style-type: none"> <li>■ Coefficient prediction (for DC values in MPEG-2 and AC values in the first row and column in MPEG-4)</li> </ul>

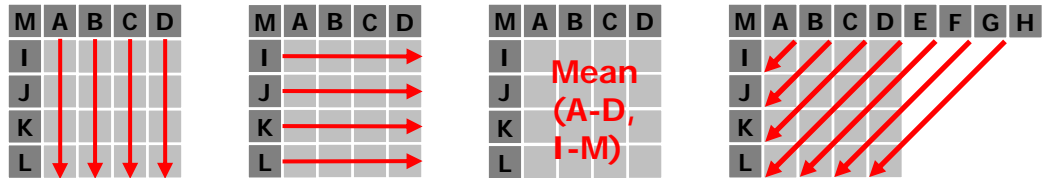
# Inter-Coded Macroblocks

	H.264	MPEG-1/2/4, H.261/3
References	<ul style="list-style-type: none"> <li>■ Permits up to 15 (2 mostly used) reference pictures</li> <li>■ Bi-predictive B-slices</li> <li>■ A P-slice may reference a picture that has B-slices</li> <li>■ Supports explicit weighting coefficients and <math>(a+b)/2</math> type</li> </ul>	<ul style="list-style-type: none"> <li>■ A P-slice references only one I-picture</li> <li>■ Bi-directional B-slices</li> <li>■ Only permit <math>(a+b)/2</math> type prediction weighting</li> </ul> 
Block Sizes	<ul style="list-style-type: none"> <li>■ Tree-structured (16x16 → 16x8, 8x16, 8x8 → 8x4, 4x8, 4x4)</li> </ul>	<ul style="list-style-type: none"> <li>■ Either 16x16 or 8x8</li> </ul>
Motion Estimation	<ul style="list-style-type: none"> <li>■ half or 1/4-pixel accuracy</li> <li>■ 6-point interpolation for half-pixel and 2-point linear interpolation for 1/4-pixel</li> </ul>	<ul style="list-style-type: none"> <li>■ MPEG2 permits half-pixel accuracy and MPEG4 permits 1/4-pixel accuracy</li> <li>■ 2-point linear interpolation</li> </ul>

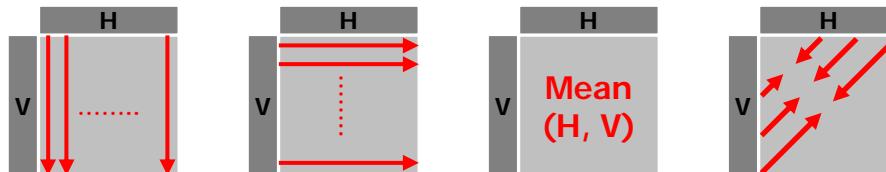
# Spatial Prediction for Intra-Coded MBs

- luma

- 4x4: 9 modes

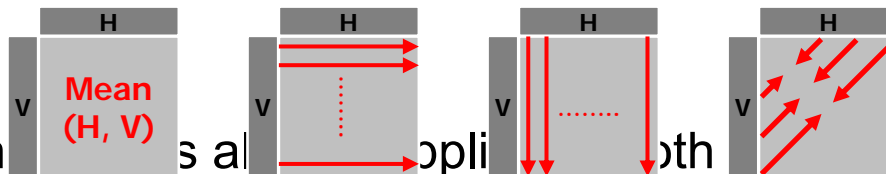


- 16x16: 4 modes



- chroma

- 8x8: 4 modes

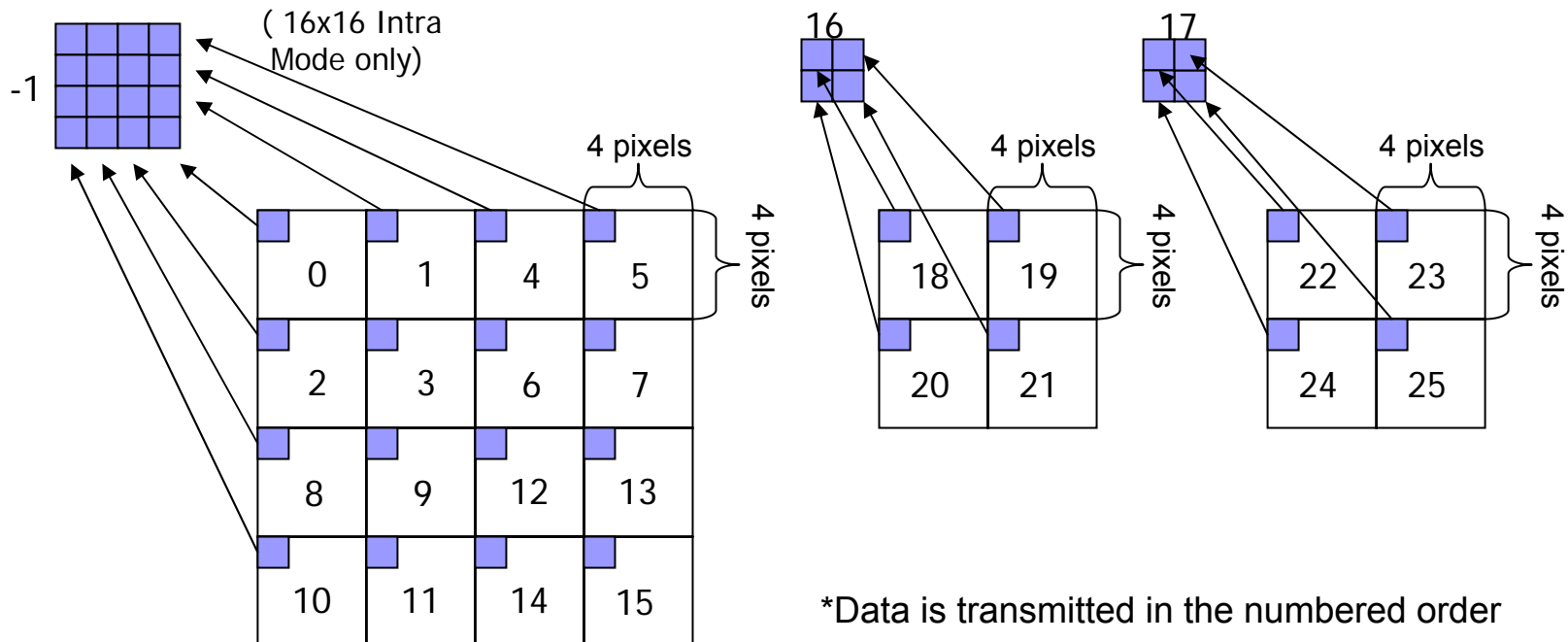


- The same prediction is applied to both blocks

# Transform and Quantization

- Three types of transform followed by quantization

- **Type 1:** for the 4x4 array of luma DC coefficients in intra MBs predicted in 16x16 mode # -1
- **Type 2:** for the 2x2 array of chroma DC coefficients #16-17
- **Type 3:** for all other 4x4 blocks # 0-15, 18-25



# Transform and Quantization – Type 3

- 4x4 DCT ( X – Input, Y – output)

$$Y = AXA^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} X \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}$$

$$a = \frac{1}{2}$$

$$b = \sqrt{\frac{1}{2}} \cos\left(\frac{\pi}{8}\right)$$

$$c = \sqrt{\frac{1}{2}} \cos\left(\frac{3\pi}{8}\right)$$

- 4x4 integer transform

- forward

$$Y = C_f X C_f^T \otimes E_f = \begin{pmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} X \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \end{pmatrix}$$

- backward

$$X' = C_i^T (Y \otimes E_i) C_i = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \left( \begin{matrix} W \\ Y \end{matrix} \right) \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \begin{matrix} \text{Post-scaling factor (PF)} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix} \end{matrix}$$

with  $a = \frac{1}{2}, b = \sqrt{\frac{2}{5}}$

# Transform and Quantization – Type 3 (2)

- 52 quantization stepsizes (Qstep) indexed by quantization parameters (QP)

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	....
QStep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5	....
QP	...	18	...	24	...	30	...	36	...	42	...	48	...	51
QStep		5		10		20		40		80		160		224

- Uniform Scalar Quantization

$$Z_{ij} = \text{round}\left(W_{ij} \cdot \frac{PF}{Qstep}\right) \quad \longrightarrow \quad Z_{ij} = \text{round}\left(W_{ij} \cdot \frac{MF}{2^{qbits}}\right)$$

$$\text{where } \frac{MF}{2^{qbits}} = \frac{PF}{Qstep} \text{ and } qbits = 15 + \text{floor}(QP/6)$$

- Integer arithmetic

$$|Z_{ij}| = (|W_{ij}| \cdot MF + f) \gg qbits \quad \text{where } f = 2^{qbits}/3 \text{ for intra MBs and } 2^{qbits}/6 \text{ for inter MBs to control the}$$

$$\text{sign}(Z_{ij}) = \text{sign}(W_{ij}) \quad \text{quantization width near the origin (the "dead zone")}$$

- The advantages of the new transform and quantization scheme:

- Integer transform avoids the inverse-transform mismatch.
- Smaller blocksize (4\*4) leads to a significant reduction in blocking artifact.
- No multiplication involved. Requires only 16-bit arithmetic.

# Entropy Coding

Parameters to be coded	entropy_coding_mode=0	entropy_coding_mode=1
Macroblock type (Intra/Inter)	Exponential Golomb codes (Exp_Golomb) Variable Length Coding (VLC)	Context-based Adaptive Binary Arithmetic Coding (CABAC)
Coded block pattern		
Quantizer parameter		
Reference frame index		
Motion vector		
Residual data	Context-adaptive variable length coding (CAVLC)	

# Exp-Golomb Entropy Coding

- Exp-Golomb codewords  
structure: [M zeros][1][M-bit info]

Code_num	Codeword
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
...	...

- Three ways of mapping parameter  $v$  to  $code\_num$ :
  1. Unsigned direct mapping: macroblock type, reference frame index and others
  2. Signed mapping: motion vector difference, delta QP and others
  3. Mapped symbols: coded block pattern parameter

# Context-adaptive variable length coding (CAVLC)

- Step0: Zig-zap ordering
  - Step 1: Encode the number of non-zero coefficients and trailing ones in 4 tables based on the number of non-zero coefficients in upper and left-hand previously coded blocks  $N_u$  and  $N_l$
  - Step2: Encode the sign of each T1 (0=+, 1=-) in reverse order
  - Step3: Encode the level of the remaining non-zero coefficients in 7 VLC tables in reverse order
  - Step4: Encode the total number of zeros and each run of zeros
- 0, 3, 0, 1, -1, -1, 0, 1, 0, 0, ...
  - TotalCoeff = 5
  - T1 = 3
  - 0000100
  - Example:
 

0	3	-1	0
0	-1	1	0
1	0	0	0
0	0	0	0
  - +, -, - → 011
  - 1 → 1, 3 → 0010
  - TotalZeros = 3 → 111  
each run → 10 1 1 01

# Context-adaptive arithmetic coding (CABAC)

- Binarization: *to binarize a non-binary-valued symbol*
- Context probability model selection: *from available models based on the statistics of recently-coded data symbols*
- Binary Arithmetic encoding
- Probability update: *The selected context model is updated based on the actual coded value*

# Deblocking Filters

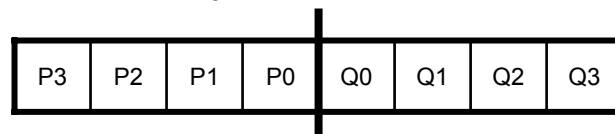
- A boundary-strength (BS) parameter is assigned to every 4×4 block

Block modes and conditions	Boundary-Strength parameter (BS)
One of the blocks is intra-coded and the edge is a MB edge	4
One of the blocks is intra-coded	3
One of the blocks has coded residuals	2
Difference of block motion $\geq$ one luma sample distance	1
Motion compensation from different reference frames	1
Else	0

- BS = 0  $\rightarrow$  No filtering
- BS = 1-3  $\rightarrow$  Slight filtering
- BS = 4  $\rightarrow$  Strong filtering

- Filters only when

- $|P_0 - Q_0| < \alpha$
- $|P_1 - P_0| < \beta$
- $|Q_1 - Q_0| < \beta$



- Thresholds  $\alpha$  and  $\beta$  depend on the average quantization parameter (QP)
- The deblocking filtering accounts for 1/3 of the computational complexity of a decoder.

# Contributions of the VCL Tools

<b>Spatial Prediction for Intra-coded Macroblocks</b>	Saves 6-9% bits
<b>Temporal Prediction</b>	Saves around 50% bits
<b>Transforms</b>	PSNR less than 0.02dB
<b>Logarithmic Quantization</b>	A change in step size by 12% also saves 12% bits
<b>CAVLC</b>	Saves 5-8% bits
<b>CABAC</b>	Saves 5-15% bits over CAVLC
<b>Picture-adaptive frame/field (PAFF) coding</b>	Saves 16%-20% bits
<b>MB-adaptive frame/field (MBAFF) coding</b>	Saves 14-16% bits over PAFF
<b>Deblocking Filter</b>	Saves 5-10% bits

# H.264 Over IP

## ■ Network Abstraction Layer Unit (NALU)

- A byte stream of variable length
- 1-byte header
  - NALU type (T)
  - NALU importance (R)
  - Error indication (F)



## ■ RTP packetization

- Simple packetization
  - One NALU in one RTP packet
  - NALU header as RTP header
- NALU fragmentation
- NALU aggregation

OSI/RM	Protocols and specifications for H.264
Application Layer	<ul style="list-style-type: none"> <li>■ RTP (Real-Time Transport Protocol)</li> </ul>
Presentation Layer	<p><u>Header size</u>: IP/UDP/RTP = 20+8+12=40 bytes</p> <p><u>Media-Unaware RTP payload specifications</u> to reduce the loss rates observed by the decoder.</p>
Session Layer	<p>Packet duplication/ Packet based FEC/Audio redundancy coding</p> <ul style="list-style-type: none"> <li>■ Control protocols: H.245, SIP (Session Initiation Protocol), SDP (Session Description Protocol), RTSP (Real-Time Streaming Protocol)</li> </ul>
Transport Layer	<ul style="list-style-type: none"> <li>■ UDP (User Datagram Protocol)</li> </ul>
Network Layer	<ul style="list-style-type: none"> <li>■ IP: best effort service</li> </ul>
Data Link Layer	
Physical Layer	

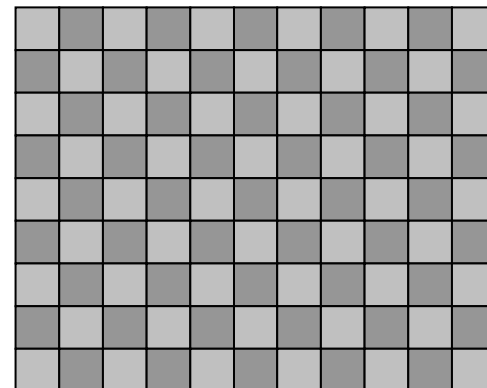
# Error-Resilience Tools

## ■ Parameter sets

- Sequence parameter set
- Picture parameter set

Slice Group #0

Slice Group #1



## ■ Flexible macroblock ordering (FMO)

- Allows to assign MBs to slices in an order other than scan order

## ■ Arbitrary slice ordering (ASO)

- Improved end-to-end delay in real-time applications

## ■ Redundant slices (RS)

- Redundant representations are coded using different coding parameters

## ■ Data partitioning with Unequal Error Protection (UEP)

## ■ Feedback from decoder to encoder

- Acknowledging correctly received slices (ACK)
- Not acknowledging message (NAK)

# Complexity, Delay & Performance of H.264 codecs

	Baseline	Main	Extended
Profile IDC	66	77	88
Targeted applications	Real-time: to minimize complexity, and provide high robustness and flexibility	Broadcast and storage: to emphasize on compression coding efficiency	Others: to combine the robustness of the Baseline profile with a higher degree of coding efficiency and extra modes
I/P/B frames	IPPP...	IBPBPBI...	IBP(SP,SI)BPBPBP...
# of frames skipped	0	1	1
# of previous frames used for inter motion search*	10	5	5
Entropy coding	CAVLC	CABAC	CAVLC
Advanced error resilience tools	FMO(flexible macroblock ordering), ASO(arbitrary slice ordering), RS		FMO, ASO, RS, DP(data partitioning)

\* Can be adjusted for each application

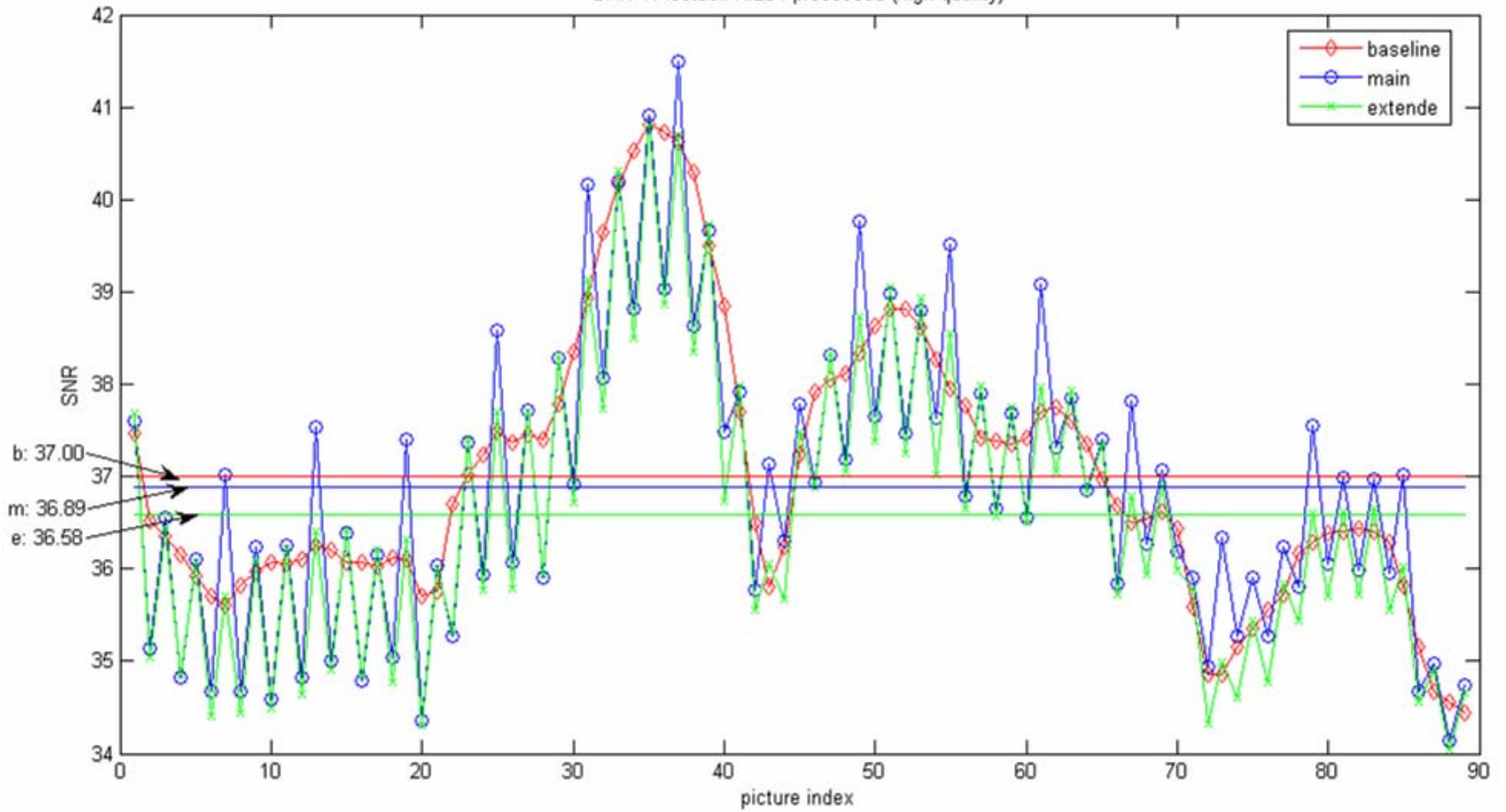
# Complexity, Delay & Performance of H.264 codecs

- Software: AVC/H.264 reference software JM9.3 (not optimized)
- Video clips:
  - football.cif 90 frames @ 30 frames/sec (fast motion)
  - paris.cif 90 frames @ 30 frames/sec (slow motion)
- Coding parameters

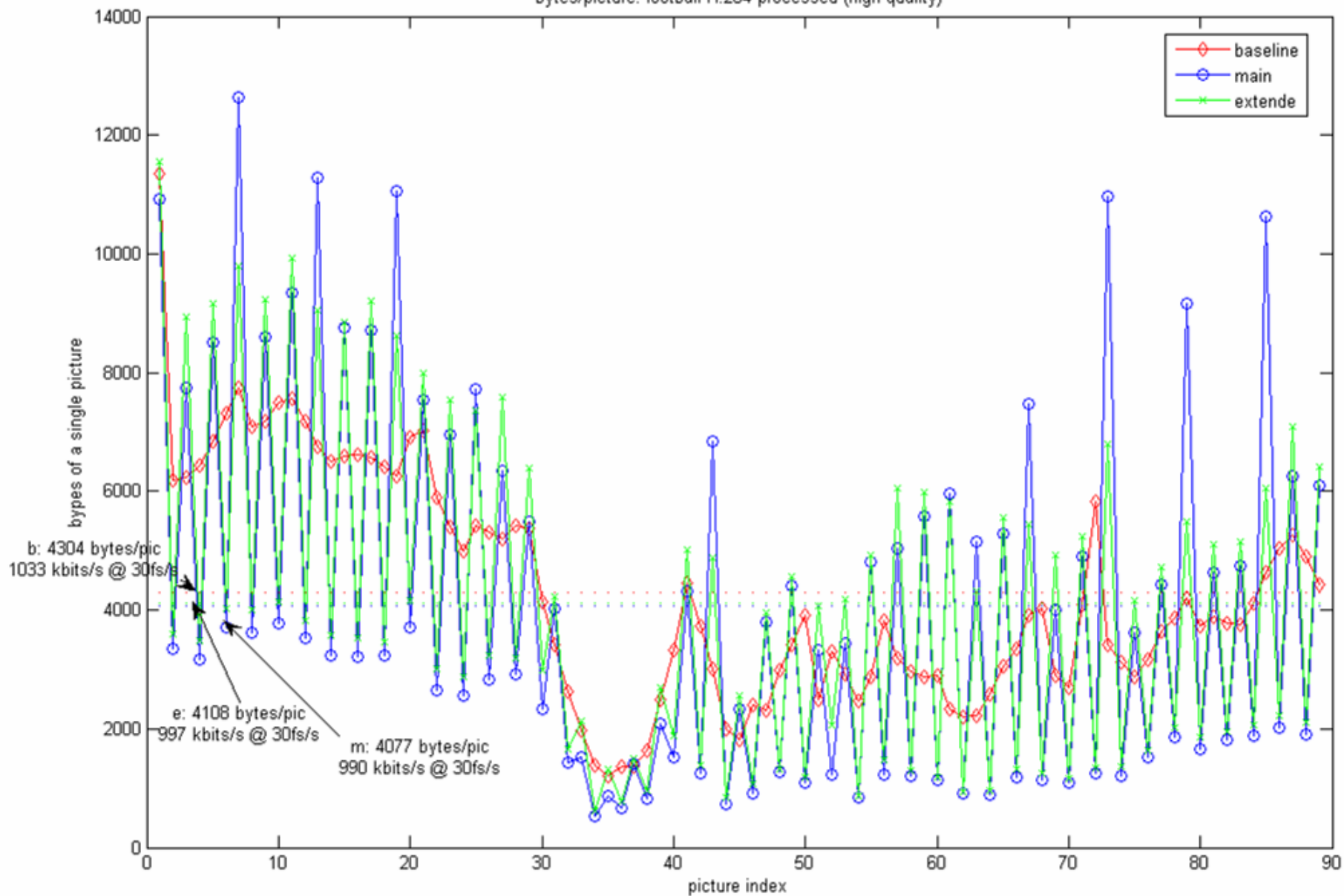
Max search range	16
Various block sizes for ME	All sizes enabled
Fast ME used?	No
SP picture	Not used
Weighted prediction	Not used
R-d optimization	On
Data partition	Not used
Slice modes	Not used

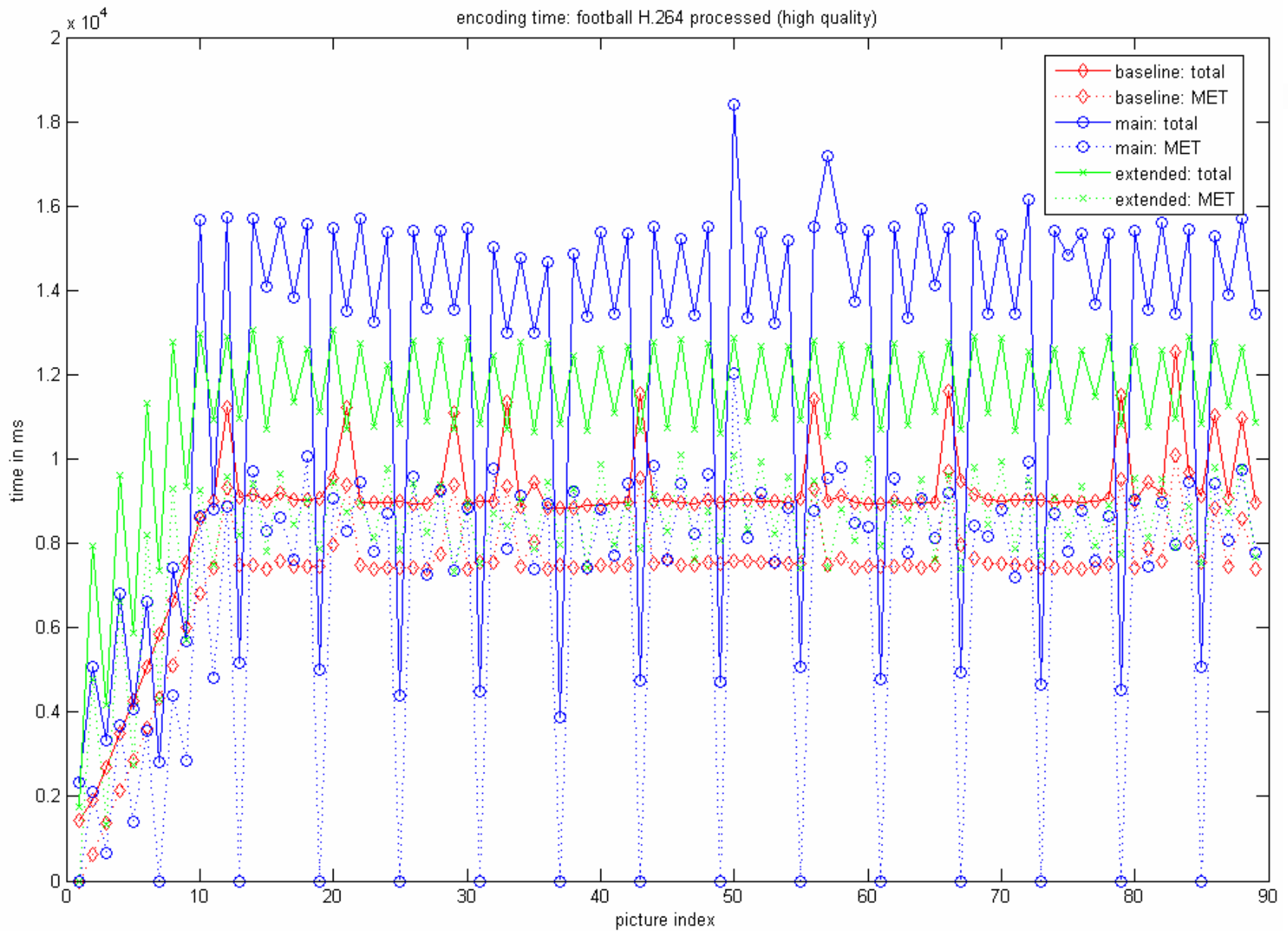
High quality	Profiles tested	B,M,E
	Video clips	Football.cif
	QP	I28/P28/B30
Low quality	Profiles tested	B
	Video clips	Football.cif & paris.cif
	QP	I30/P31

SNR-Y: football H.264 processed (high quality)

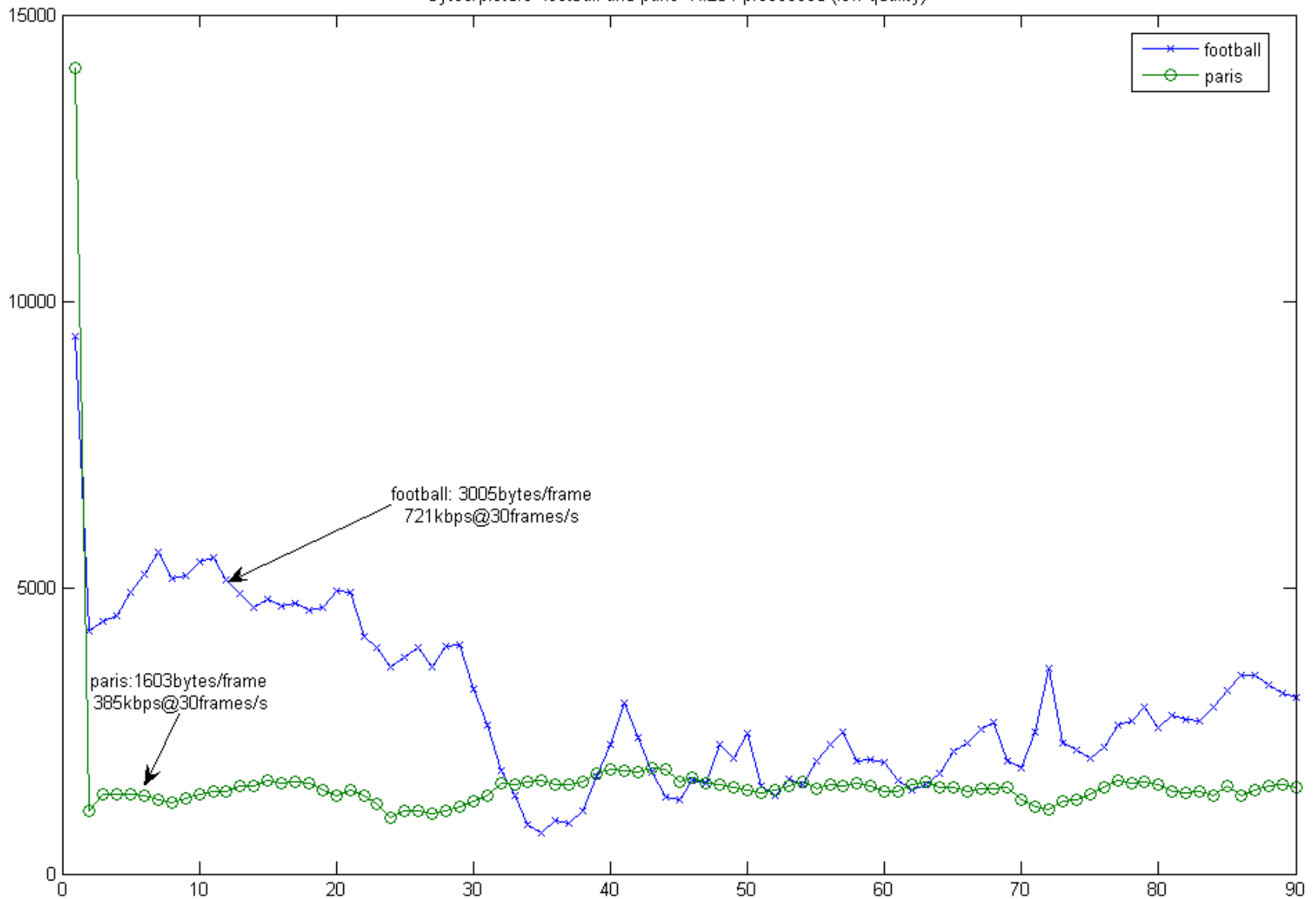


bytes/picture: football H.264 processed (high quality)

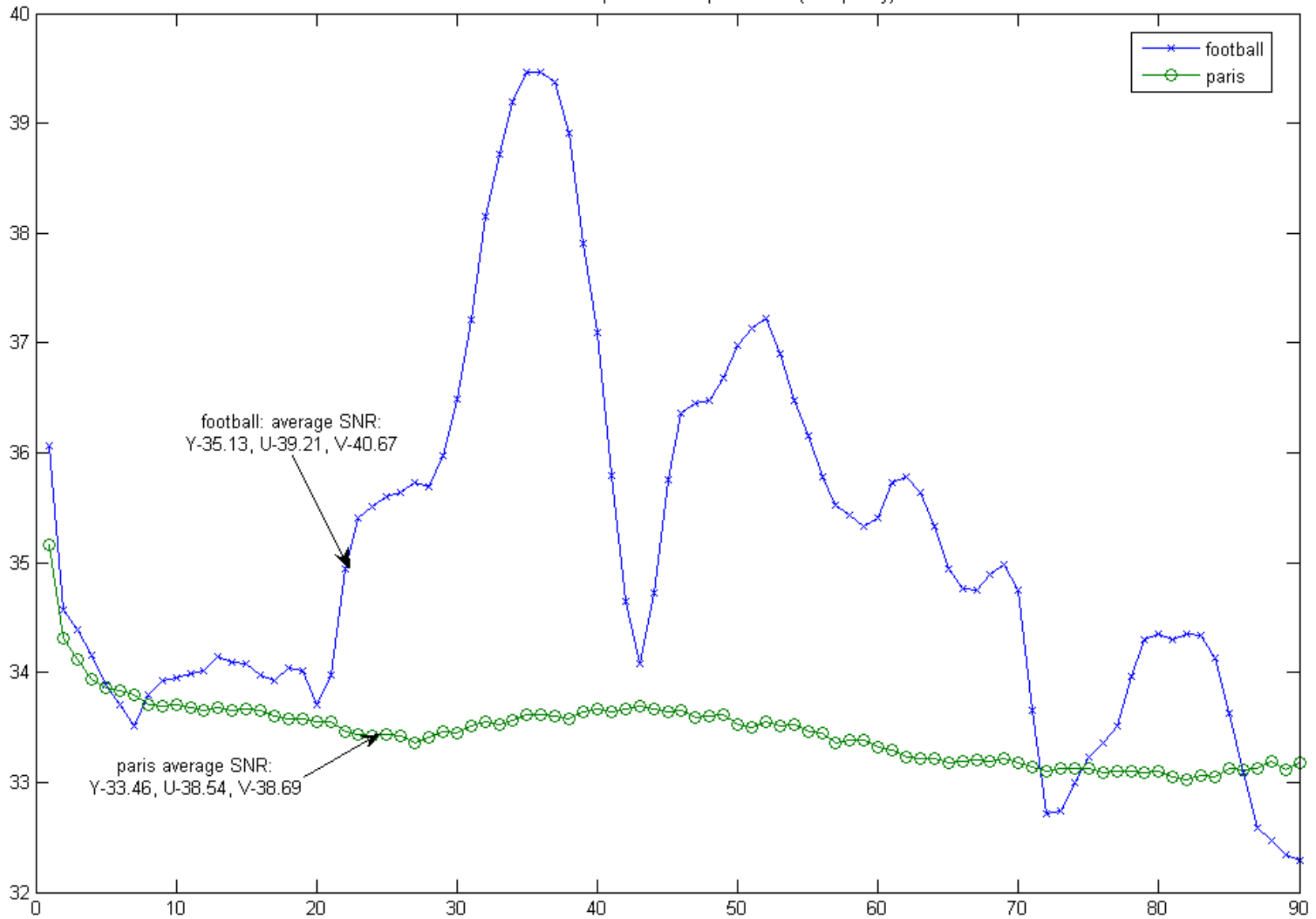




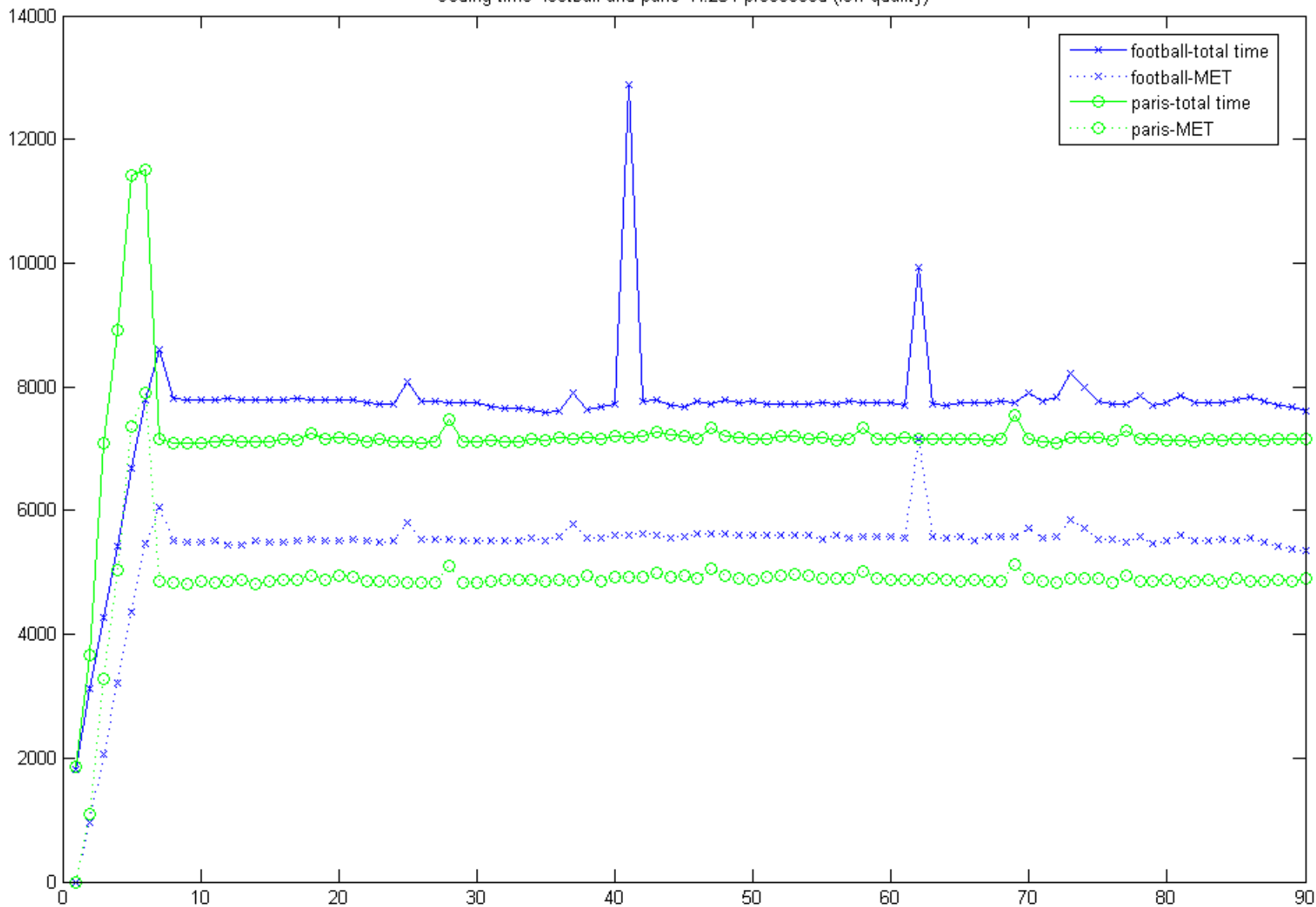
bytes/picture--football and paris--H.264 processed (low quality)



SNR-Y-football and paris-H.264 processed (low quality)



Coding time--football and paris--H.264 processed (low quality)



# Complexity, Delay & Performance of H.264 codecs

- “*The JVT Advance Video Coding Standard: Complexity and Performance Analysis on a Tool-by-Tool Basis*”, S. Saponara, C. Blanch, K. Denolf and J. Bormans, 2003
  - H.264 has a complexity increase of more than one order of magnitude at the encoder and a factor 2 for the decoder than MPEG-4 (Simple Profile)
  - The complexity, delay and performance of H.264 codecs varies largely when different tools are chosen

		MD		FOR1		FOR2		MC	
		Min	Max	Min	Max	Min	Max	Min	Max
PSNR -Y		36.24	36.77	35.19	36.00	35.77	36.51	37.30	37.90
Kbps		24.80	33.29	96.14	145.80	276.89	435.45	1305.20	2243.93
Encoder	Peak memory (Mbytes)	2.19	15.60	2.19	15.60	7.31	26.87	7.31	26.87
	Accesses ( $10^9/s$ )	1.40	79.92	1.18	65.78	4.60	258.01	2.75	134.26
	Relative time	6.48	409.12	5.40	330.87	21.70	1117.48	12.98	567.37
Decoder	Peak memory (Mbytes)	1.06	1.38	1.06	1.38	2.91	4.15	2.91	4.15
	Accesses ( $10^6/s$ )	61.10	104.41	90.16	153.88	385.21	636.90	287.10	492.54
	Relative time	0.30	0.69	0.58	1.30	3.04	5.51	2.33	4.26

# Complexity, Delay & Performance of H.264 codecs

- Contributing to most of encoding delay

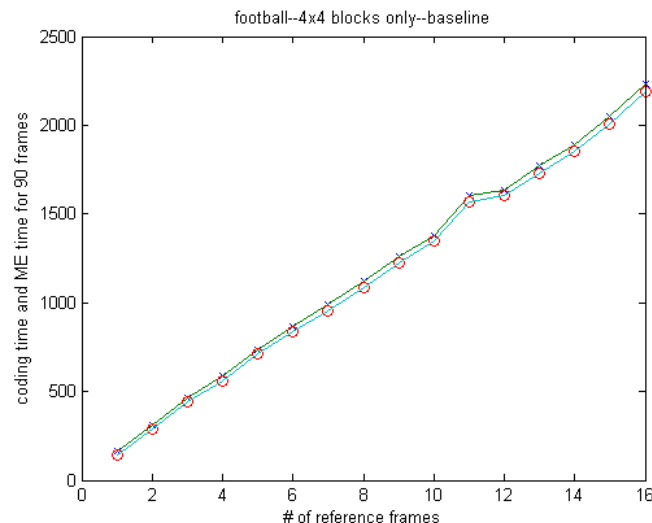
- Motion estimation
  - Around 90% of total computation
- DCT and IDCT
- Pixel interpolation

- Contributing to most of decoding delay

- Deblocking filtering
  - Around 30% of total computation

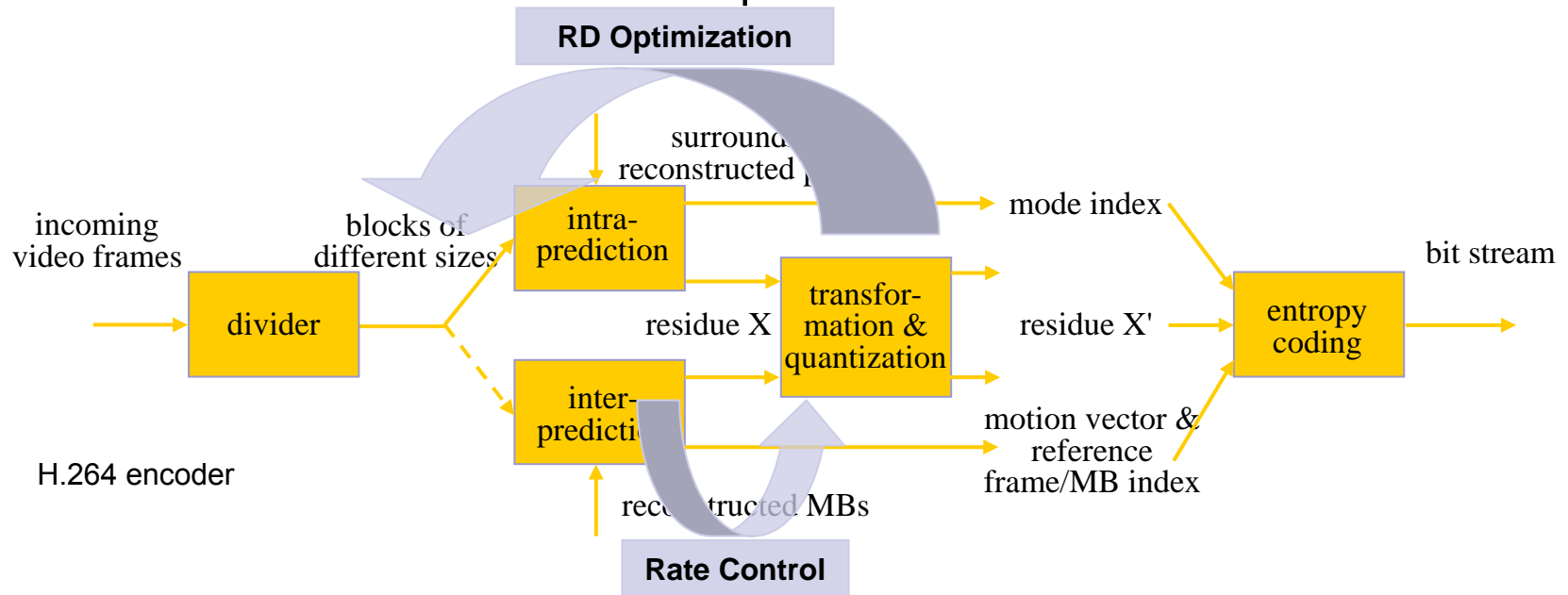
- How fast can the H.264 codecs in the market perform?'

- Most vendors have recently claimed simultaneous H.264 codecs for CIF videos at 30frames/second
- Decoding time is on the order of 1% of encoding time.



# Research Problems in H.264 (to name a few)

- Rate control and rate distortion optimization



- Fast algorithm and architecture design of low-power motion estimation
- Cross-layer design of video over IP networks
- Perceptual coding in AVC/H.264

# What's next? H.265?

- ITU-T Study Group 16 **Multimedia terminals, systems and applications** - Question 6/16 **Video coding** (Study Period 2005-2008)
- Study Items
  - New coding methods in order to achieve the following objectives:
    - improvements in compression efficiency;
    - robust operation in error/loss-prone environments (e.g. non-guaranteed-bandwidth packet networks or mobile wireless communication);
    - reduction of real-time delay;
    - reduction of channel acquisition time and random access latency;
    - reduction of complexity;
  - Organization of the compressed data format to support packetization and streaming;
  - Methods to allow streams to be easily mixed by MCUs or terminals;
  - Techniques to permit networks or terminals to adjust the bit rate of video streams efficiently;
  - Techniques for object coding and multiview operation;
  - Techniques for efficient compressed-digital to compressed-digital processing (including transcoding).
  - The impact of colorimetry, video quality assessment, and quality control requirements on video codec development.
- Final Rec. H.265: expected 2008-2010.

# H.265 – Technology Contenders



- Introduction of Wavelets in video coding
- New ways of distributed source video coding.
- Intelligent decoders & encoder – Computer Vision tools
- Development of new transforms
- New scalability schemes.
- Switch to progressive (no interlaced) only tools.
- Better Data-partitioning and Error-concealment techniques.

# References

- Peter Symes, Digital Video Compression, McGraw-Hill, 2003.
- Joan L. Mitchell, MPEG Video Compression Standard, Chapman & Hall : International Thomson, 1996.
- William B. Pennebaker and Joan L. Mitchell, JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, 1993.
- H.264/MPEG-4 Part 10 White Paper, [www.vcodex.com](http://www.vcodex.com)
- IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, July 2003.
- B. Yan and K.W. Ng, A Survey on the Techniques for the Transport of MPEG-4 Video Over Wireless Networks, IEEE Transactions on Consumer Electronics, Vol. 48, No. 4, November 2002.
- MPEG Video Compression Technique, [http://rnvs.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg\\_tech.html](http://rnvs.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg_tech.html)
- P. Hoyingcharoen and C. Schmidt, Overview of H.264/AVC, Signal Compression Lab, UCSB.

# SP and SI-Frame Design

## ■ SP and SI-frames

- allow identical reconstruction when coded using different references
- Subtract the reference in the coder and add it back in the decoder
- Coding efficiency of the SP-frame is worse than that of P-frames but much better than the I-frame
- Coding efficiency of the SI-frame is worse than that of I-frames

## ■ Bitstream switching

- In previous coding standards:  
perfect (mismatch-free) switching  
only happens at Intra-frames.

## ■ Other applications

- Bitstream splicing
- Error recovery/resilience
- Video redundancy coding

