The H.264/MPEG-4 AVC Video Compression Standards

Till Halbach

Institutt for teleteknikk
Norges teknisk-naturvitenskaplige universitet (NTNU)
Trondheim (Norway)

November 2003

Outline

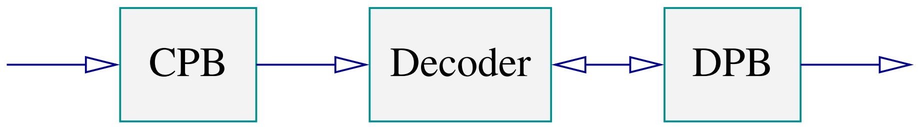
- Standardization efforts
- Decoder
- Encoder
- Basic technical concepts
- Algorithm sets: Baseline, Main, and Extended
- Performance evaluation
- Conclusions
- Outlook

Standardization efforts

- Project start: 1998
- Project name: H.26L
- ITU-T Video Coding Experts Group
- ISO/IEC Moving Pictures Experts Group
- Coordinated efforts since Dec. 2001: Joint Video Team
- ITU-T Recommendation H.264
- ISO/IEC International Standard 14496-10 (MPEG-4 AVC)
- Finalization: May/October 2003
- First 3rd-generation video coding standard

H.264-conform decoder

- Decoder issues
  - ▷ Informal supplements: Encoder matters and topics like e.g. error concealment
- Bit stream syntax and semantics
- *Coded picture buffer*
- Decoding engine
- *Decoded picture buffer*

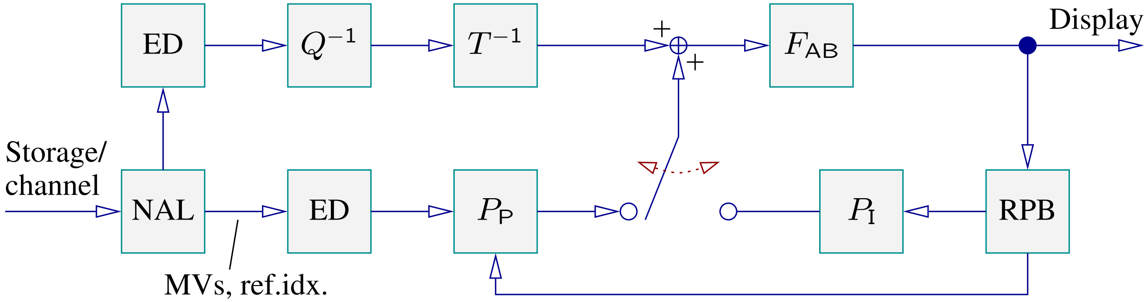


- *Hypothetical reference decoder*

H.264 decoding engine

❑ Generic block diagram

▷ Internal 16-bit implementation



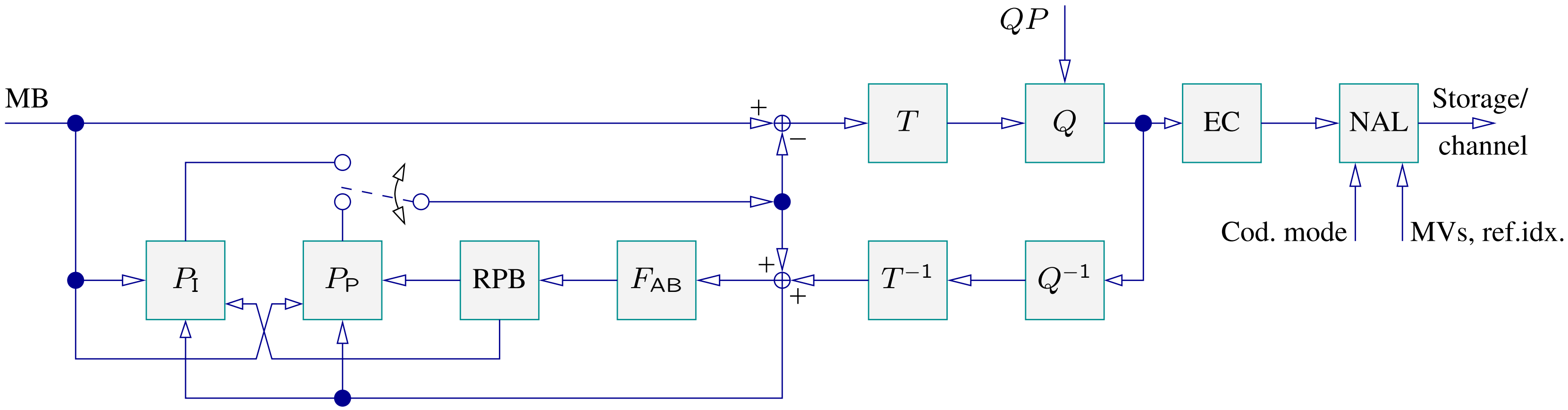
▷ Bit stream with 8 bit per sample accuracy as output

▷ 4:2:0, 4:2:2, and 4:4:4 chrominance subsampling (YCbCr color space with luma and chroma; here 4:2:0)

▷ Maximum of 15 reference pictures

▷ Network abstraction: Time-multiplexing of side information and encoded data

H.264-conform encoder



- Block-based hybrid coding scheme
  - Macroblock: 16×16-pixel luma and two 8×8-pixel chroma signals (YCbCr 4:2:0)
  - Video Coding Layer and Network Abstraction Layer
  - Spatial/temporal prediction
  - Reference picture buffer
  - Transform, quantization and entropy encoding of prediction error
  - Adaptive non-linear in-loop anti-blocking filtering of block edges

Picture segmentation

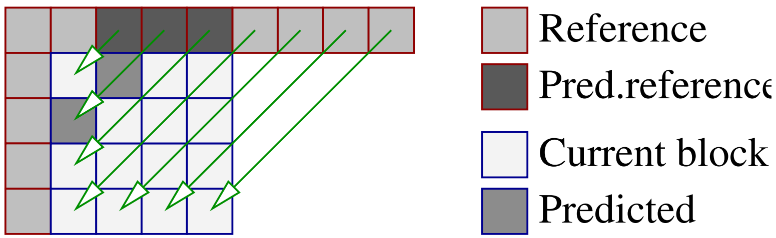
- Macroblocks (MBs)
  - Here: QCIF example; 11×9 MBs

  - Memory and complexity constraints under en- and decoding
  - MB pair interpretation: Two frame MBs or one top- and one bottom-field MB (only in interlaced coding mode)
  - Drawback: Blocking artifacts

Basic technical concepts

- *Profiles* and *levels*
  - Baseline: low complexity, low latency
  - Main: High complexity, high latency
  - Extended: Low complexity, high error resistance

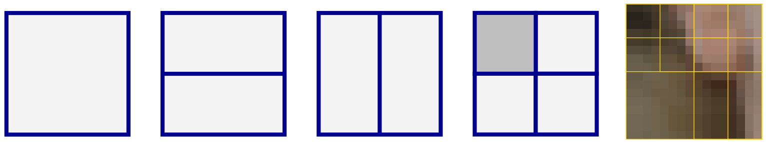
- INTRA coding (I MBs)
  - Nine 4×4 luma modes (here: '4×4 diagonal down left')



  - Four 16×16 luma modes
  - Four 8×8 chroma modes
  - Differential coding of 4×4 prediction modes

Basic technical concepts, cont'd

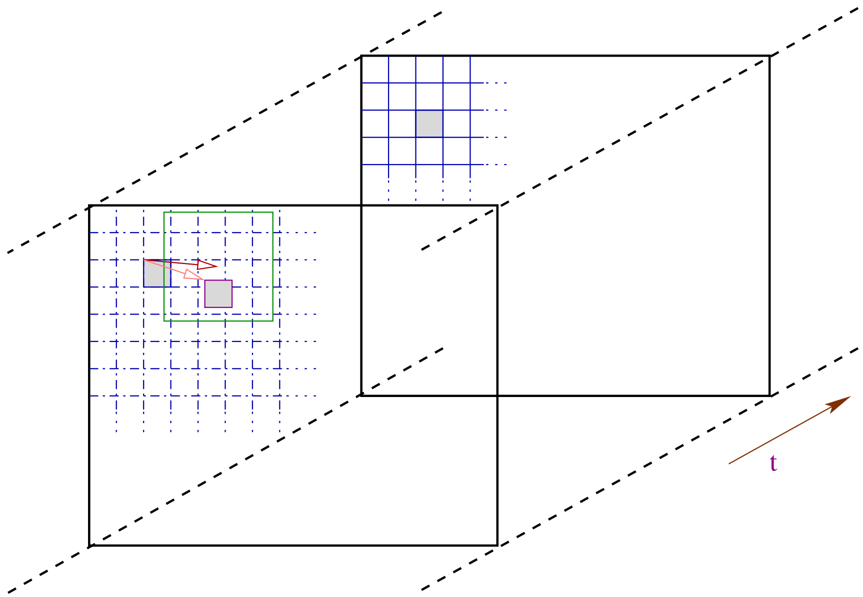
- INTER coding (P MBs)
  - Hierarchical tree-structured motion segmentation with possible luma block sizes ($x \times y$) 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4 pixels



  - Consistent I/P mode within one MB
  - Quarter pel accuracy of luma motion estimation/compensation with filter combination (1, -5, 20, 20, -5, 1) and (1, 1)
  - Bilinear interpolation of chroma signal and re-use of luma motion vectors
  - Sample extrapolation at image boundaries
  - Differential coding of (up to 16 per MB) motion vectors: Median or direct estimate

Basic technical concepts, cont'd

- P/INTER coding, cont'd
  - ▷ Multiple reference frames: Conveyance of frame index for each 8×8-pixel block
  - ▷ Example: Motion estimation/compensation (ME/MC) for a single block



  - ▷ Capture of temporal activity; motion adaptation

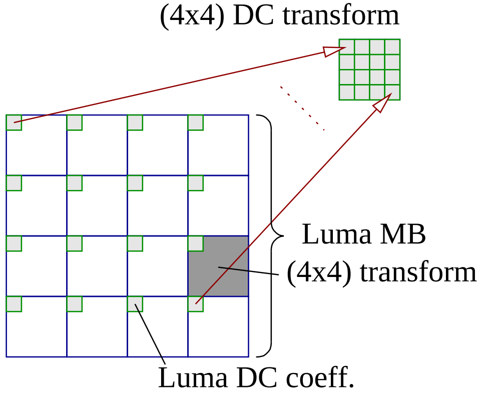
Transforms

- INTRA 4×4 mode: Separable DCT-approximating 4×4-size integer transform

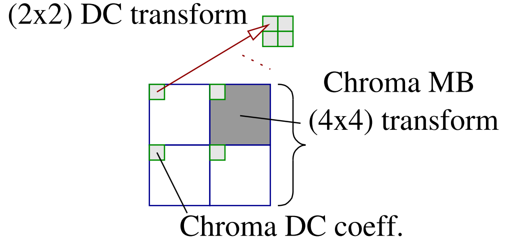
$$T_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

- No multiplications, 16-bit arithmetic realizations
- Perfect reconstruction (PR)
- Scaling combined with subsequent quantization stage

- Transforms, cont'd
  - INTRA 16×16 mode: Additional 4×4-size PR integer transform for luma DC coefficients

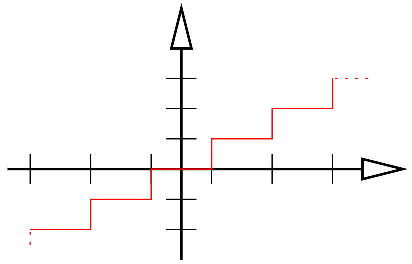


  - Additional 2×2-size PR integer transform for chroma DC coefficients



Quantization

- Set of 52 uniform mid-tread scalar quantizers



- Parameter QP for specification of step size
  - The lower QP, the higher $PSNR$ and channel bit rate
  - Increase in scaling magnitude of approximately 12% for QP increment by one
  - Quantization factor doubles with QP increase of 6
- Coarser step size for luma than for chroma
- Non-weighted quantization

Encoding example

- Original block

$$\begin{pmatrix} 43 & 216 & 255 & 249 \\ 49 & 198 & 193 & 211 \\ 48 & 194 & 177 & 171 \\ 46 & 214 & 225 & 169 \end{pmatrix}$$

- Encoder chooses 4×4 DC prediction mode: Prediction

$$\begin{pmatrix} 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 \end{pmatrix}$$

- Prediction error

$$\begin{pmatrix} -85 & 88 & 127 & 121 \\ -79 & 70 & 65 & 83 \\ -80 & 66 & 49 & 43 \\ -82 & 86 & 97 & 41 \end{pmatrix}$$

Encoding example, cont'd

- Block after transform

$$\begin{pmatrix} 610 & -1256 & -686 & -558 \\ 279 & -478 & 111 & -69 \\ 176 & -160 & -120 & 100 \\ -13 & -14 & 3 & 3 \end{pmatrix}$$

- Quantized coefficients

$$\begin{pmatrix} 7 & -9 & -8 & -4 \\ 2 & -2 & 1 & 0 \\ 2 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- (run,level) pairs: (0,7), (0,-9), (0,2), (0,2), (0,-2), (0,-8), (0,-4), (0,1), (0,-1), (2,-1), (1,1)

- Reconstructed block

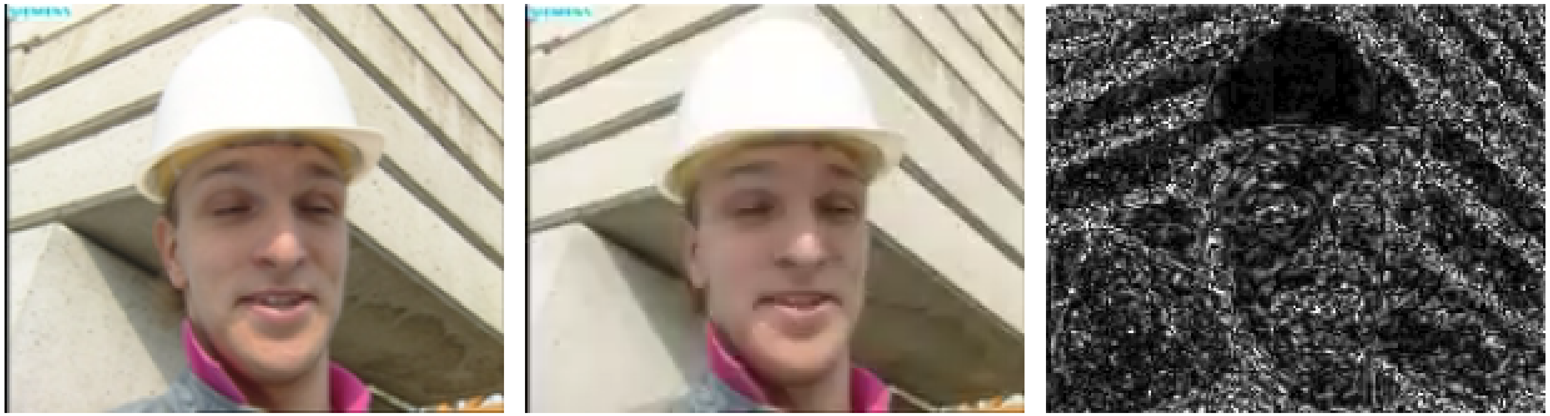
$$\begin{pmatrix} 57 & 204 & 246 & 238 \\ 52 & 194 & 189 & 204 \\ 48 & 195 & 174 & 169 \\ 50 & 207 & 217 & 167 \end{pmatrix}$$

Encoding example, cont'd

- Coding error

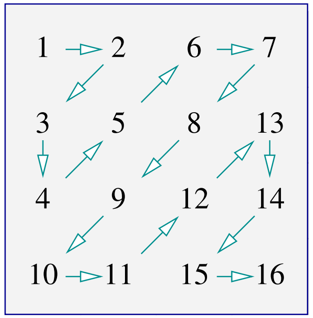
$$\begin{pmatrix} -14 & 12 & 9 & 9 \\ -3 & 4 & 4 & 7 \\ 0 & 1 & 3 & 2 \\ -4 & 7 & 8 & 2 \end{pmatrix}$$

- Original, reconstructed picture, and (scaled) coding error (INTRA frame, QP 30, luma PSNR 35.70 dB)



Baseline profile

- Zig-zag scan (mapping of transform coefficients)
  - ▷ Coefficient ordering from high to low frequency in forward scan



  - ▷ Starting at first position for luma 4×4-pixel blocks
  - ▷ Starting at second position in luma 16×16 block mode and chroma (only AC coeff.)
    - 16 Scans of luma 4×4-sample blocks in a MB
    - 4 Scans of chroma 4×4-sample blocks in a MB
  - ▷ Resulting in (level,run) pairs and EOB

Baseline profile, cont'd

- Entropy encoding
  - Exponential Golomb code with parameter zero for most side information and header data

Index	Code word
0	1
1	010
2	011
3	00100
4	00101
⋮	⋮

- Context-adaptive variable-length coding (CAVLC) of transform coefficients
  - Treating *levels* and *run's* separately
  - Coding number of coefficients and *trailing ones* (code table choice with regard to number of coefficients in neighboring blocks), and signs

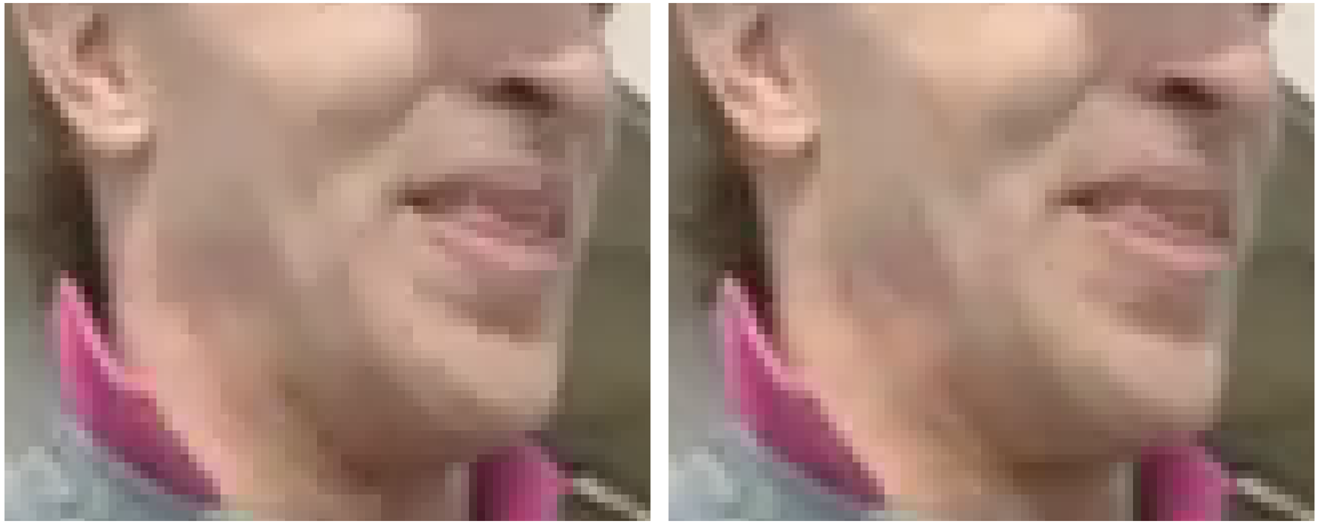
Baseline profile, cont'd

❑ CAVLC, cont'd

- ▷ Coding of sequence of remaining coefficients using adaptive Rice codes (code table choice with regard to the previous encoded coefficient)
- ▷ Coding of sum of run's, dependent on non-zero coefficients
- ▷ Coding of sequence of run's (code table choice with regard to remaining sum of run's)
- ▷ Example: Coeff. (12, -7, 0, 0, 5, 1, -1, 0, -1, 1, 0, . . . , 0) $\Rightarrow$
  Non-z. coeff.: (12, -7, 5, 1, -1, -1, 1), run's: (0, 0, 2, 0, 0, 1, 0)
  N.o. non-z. coeff.: 7; n.o. trailing 1's: 3; Sum of run's: 3
  $\Rightarrow$ EC((7,3)); EC((+, -, -)); EC((1, 5, -7, 12)); EC((0, 1, 0, 0, 2, 0, 0))
- ▷ Decoder: Placement of highest-frequency coefficient first and then remaining coefficients in a backward manner

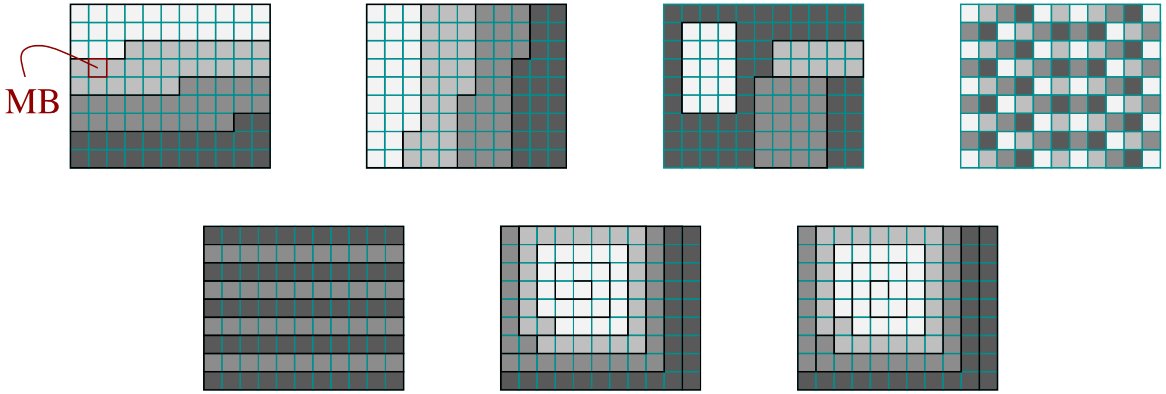
Baseline profile, cont'd

- Adaptive in-loop filter against blocking artifacts
  - Placement behind inverse transform
  - Reduction of blocking artifacts
  - Filtering along MB and block edges
  - Determination by prediction type, motion vector data, prediction error energy, and quantization level
  - Example: Without loop filter (left) vs. enabled loop filter (right)



Baseline profile, cont'd

- **Slice concept**
  - ▷ Group MBs into *slice groups* by flexible MB allocation: Horizontal and vertical raster scan, rectangular slices, dispersive allocation, interlaced slice groups, clock-wise and counter-clock-wise spiral scan, and explicit allocation

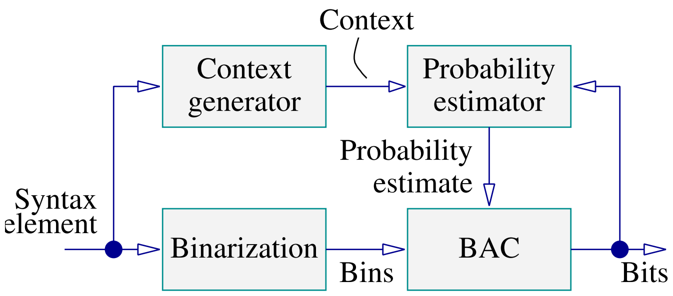


  - ▷ Break slice groups into smallest independently decodable units in encoded data stream, *slices*
  - ▷ I slice contains I MBs only, P slice may contain I and P MBs

- **RS, ASO, and SEI**

Main profile

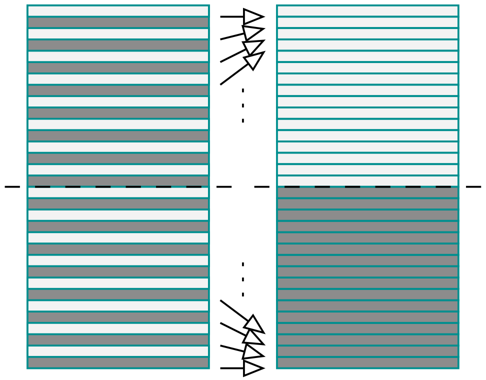
- Context-adaptive binary arithmetic coding of transform coefficients
  - Binarization: 0, 01, 001, . . .

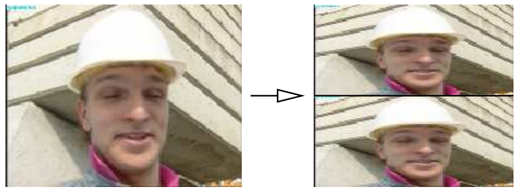


- B MBs/slices
  - Average of predictions from two reference frames
  - Temporally forward and/or backward predictions
  - B frames may be used as reference
  - B slice may contain I, P, and B MBs

Main profile, cont'd

- Linear weighted prediction
  - P MBs: $X_\mathsf{p} = a \cdot X_\mathsf{r} + c$
  - B MBs: $X_\mathsf{p} = a \cdot X_{\mathsf{r},1} + b \cdot X_{\mathsf{r},2} + c$
  - Association of each frame with separate weighting
- Interlaced modus: MB-adaptive frame/field coding
  - Split-up of MB pair into top- and bottom-field MB

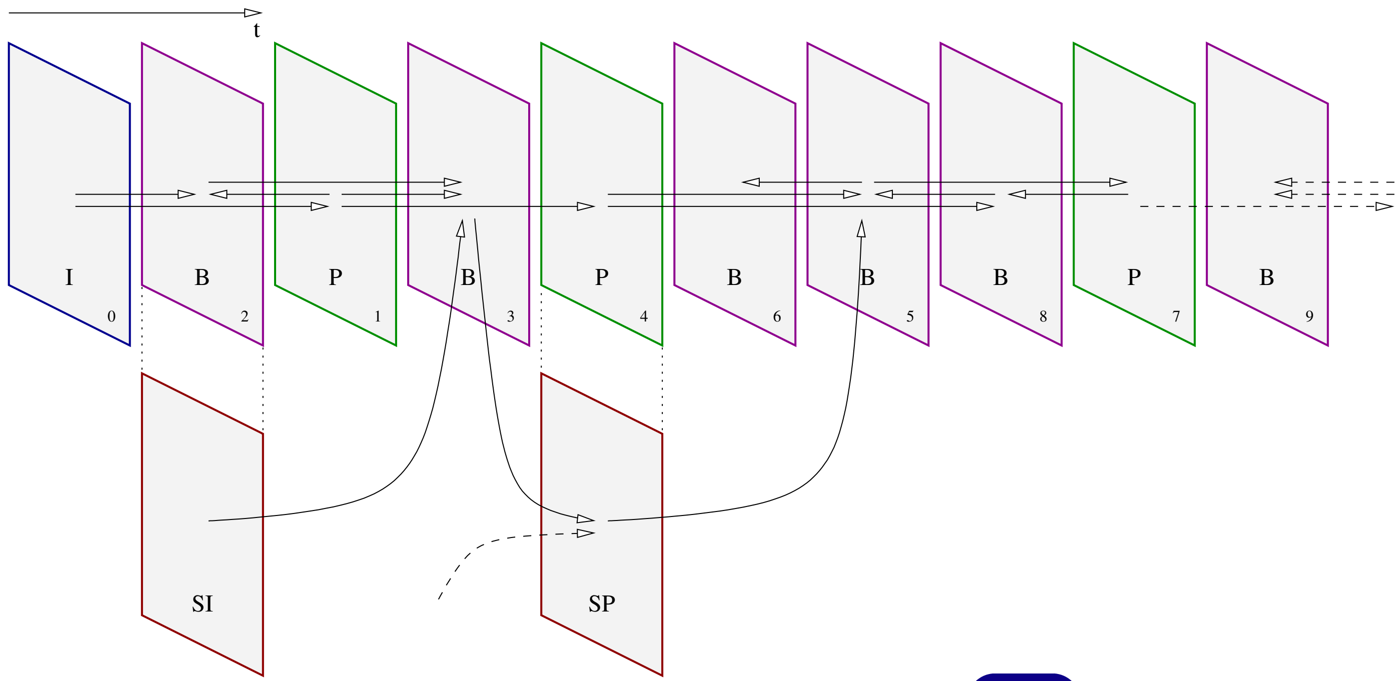




  - Temporal reference to current frame but different field possible

Extended profile

- Data partitioning
  - Header data, INTRA-specific data, INTER-specific data
  - Transmission of one partition as NAL packet
- Switching slices
  - SI and SP slices



Encoder issues

- Interpolation and fractional sample ME/MC: Integer position – E, 1/2-pel pos. – 7, 1/4-pel pos. – d

$$
\begin{array}{ccccccccc}
x & + & x & + & x & + & x & + & x \\
* & * & * & * & * & * & * & * & * \\
x & + & x & + & x & + & x & + & x \\
* & * & * & * & * & * & * & * & * \\
x & + & x & + & x & + & x & + & x \\
* & * & * & * & * & * & * & * & *
\end{array}
$$

$$
\begin{array}{ccccccc}
A & & & B & & & C \\
 & \mathsf{1} & & \mathsf{2} & & \mathsf{3} & \\
D & \mathsf{4} & & E & & \mathsf{5} & F \\
 & & a & b & c & & \\
 & \mathsf{6} & d & \mathsf{7} & e & \mathsf{8} & \\
 & & f & g & h & & \\
G & & & H & & & I
\end{array}
$$

Encoder issues, cont'd

- Fast 'spiral' MV search around predicted vector

$$\begin{array}{ccccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 15 & 9 & 11 & 13 & 16 & \cdot \\
\cdot & 17 & 3 & 1 & 4 & 18 & \cdot \\
\cdot & 19 & 5 & 0 & 6 & 20 & \cdot \\
\cdot & 21 & 7 & 2 & 8 & 22 & \cdot \\
\cdot & 23 & 10 & 12 & 14 & 24 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}$$

- Prediction mode selection
  - ▷ Lagrange functional minimization combined with $SATD$
    - Low complexity
    - Mode that minimizes the overall transformed prediction error

$$J = SATD + \lambda \cdot R$$

Encoder issues, cont'd

- Prediction mode selection, cont'd
  - ▷ Lagrange functional minimization, cont'd
    - Sum of absolute transformed differences ($SATD$)

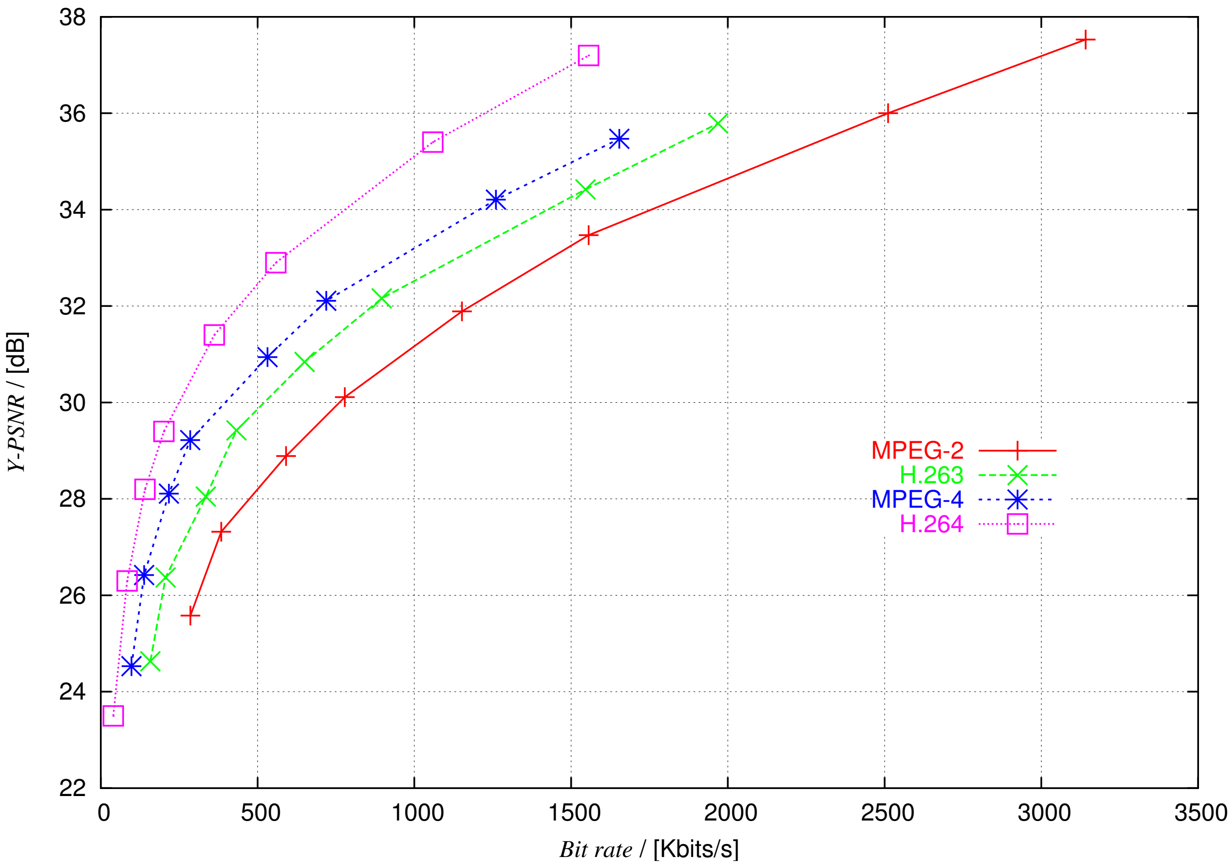
$$SATD = \frac{1}{2} \sum_{i,j=0}^{3} |T_{\mathsf{H}}\{D(i,j)\}|$$

    - 2-D Hadarmard transform $T_{\mathsf{H}}\{\cdot\}$
    - Prediction error

$$D(i,j) = X_{\mathsf{org}}(i,j) - X_{\mathsf{pred}}(i,j)$$

    - Original and predicted samples X; sample position i in line j
  - ▷ Lagrange rate distortion optimization criterion
    - Based on real rate and real distortion for each encoded block, prediction mode (INTRA /INTER), and reference pictures
    - High complexity

Performance evaluation (Main profile)



- Roughly 2 dB gain in $PSNR$ to closest competitor
- Bit rate saving of approximately 40%

Conclusions and outlook

- Standardization efforts
- Decoder, Encoder
- Basic technical concepts
- Profiles: Baseline, Main, and Extended
- Performance evaluation
- Outlook
  - Royalty-free Baseline
  - Real-time market solutions already in position
  - Gradual replacement of MPEG-4 (Visual) and H.263++
  - Broad spectrum of applications
  - Continuation of standardization
  - Probably the last hybrid coding standard

References and links

- JVT, VCEG, and MPEG, as well as H.26L/H.264/MPEG-4 AVC
  - `http://kbs.cs.tu-berlin.de/~stewe/vceg/index.htm` (preliminary)
  - `http://www.tele.ntnu.no/users/halbach/h26l/`
- Introduction to/Overview of H.264
  - Till Halbach. The H.264 Video Compression Standard. in Proc. Norwegian Signal Processing Symposium (NORSIG), Bergen (Norway), October 2003
- The latest standard description (FDIS, almost identical with IS), as well as all other standardization documents
  - Document `VCEG-G050.doc` on `ftp://ftp.imtc-files.org/jvt-experts/` (anonymous login)
- News about MPEG-4 and reference software
  - `http://www.m4if.org`