# ITU-T H.264 / MPEG-4 AVC (Advanced Video Coding) 14496-10





## H.264/MPEG-4 AVC Development via JVT

- ISO/IEC MPEG and ITU-T VCEG formed a Joint Video Team (JVT) in December 2001
- JVT starting from TML9 Test model of ITU-T H.26L is working on an Advanced Video Coding (AVC) standard
- This new standard will be a joint standard, MPEG-4 Part 10/AVC, as well as ITU-T Recommendation H.264
- Goal of AVC was significantly higher (2x) coding efficiency for a broad range of applications including interlaced video
- Development activity
  - Dec. 2001 : Working Draft (WD-1.0)
  - May 2002 : Committee Draft (CD)
  - Aug. 2002 : Final Committee Draft (FCD)
  - Dec. 2002 : Cleanup of FCD
  - Feb. 2003 : Study of FCD
  - June 2003 : Approval of Standard

# Potential Applications of H.264/AVC

- *Broadcast Video* over cable, satellite, DSL, terrestrial
- *Stored Video* on DVD, CD, and magnetic devices
- *Interactive Conversation* over ISDN, Ethernet, LAN, DSL Wireless Networks, modems, etc. or a mixture of several
- Video-on-demand or Multimedia *Streaming* over DSL, ISDN, Ethernet, LAN, Wireless Networks
- *Multimedia Messaging Services* (MMS) over DSL, ISDN, Ethernet, LAN, Wireless Network





# H.264/AVC Profiles and Applications

- Baseline Profile : Low Delay Real Time Communication
- Extended Profile: Streaming
- Main Profile: TV and DVD

| Tools in Baseline Profile | Tools in Main Profile |
|---|---|
| • Only I and P slice types may be present<br>• No interlace support<br>• No CABAC<br>• Arbitrary slice order (ASO)<br>• Flexible macroblock ordering (FMO)<br>• Redundant pictures allowed | • I, P, B slices<br>• CABAC<br>• Support for interlaced video<br>• No ASO<br>• No FMO<br>• No redundant pictures |

# Coding Efficiency vs Complexity

- Difficult to say for sure, but there are vendor white papers that are making claims that H.264/AVC tradeoffs are as follows

| AVC Profiles | Target Applications | Rough Decoder Complexity Increase over MPEG -2 | Preliminary Estimates of Efficiency Improvements over MPEG -2 |
|---|---|---|---|
| Baseline Profile | Low delay applications, video phone, mobile … | 2.5 X more complex | 1.5 x better |
| Extended Profile | Mobile, streaming,… | 3.5 X more complex | 1.75 x better |
| Main Profile | Interlaced video applications, broadcast, packaged media, … | 4.0 X more complex | 2.0 x better |

- Need to compare with MPEG-4 ASP (or ASP without GMC) coding efficiency/complexity tradeoffs





# Tools in H.264/AVC vs Other Standards

| Visual Tools | MPEG-2 | MPEG-4 part 2 (Simple, Advanced Simple profiles) | H.263 (Baseline, H.263+, H.263++) | AVC H.264/MPEG-4 part 10 (Baseline, Extended, Main profiles) |
|---|---|---|---|---|
| I and P pictures | Yes | Yes | Yes | Yes |
| B pictures | Yes | Many profiles | Some profiles | Yes (Main & Extended profiles) |
| 16x16, 16x8 Motion compensation | Yes 16x16, 16x8 field | Yes 16x16, 16x8 field | Basically 16x16 | Yes 16x16, 16x8, 8x16 |
| 8x8 Motion comp | No | Yes | Yes | Yes |
| Sub 8x8 Motion comp | No | No | No | Yes |
| Motion vector prediction | Simple | Adaptive | Adaptive | Adaptive |
| unrestricted MVs | No | Yes | Yes | Yes (Main profile) |
| OBMC | No | No | Yes | No |
| > two reference frame prediction | No | No | Some profiles | Yes |
| Deblocking Filter | No | as Postprocessing | In-loop | In-loop |
| Global MC | No | Some profiles | No | No |
| 1/4 pixel MC | No | Some profiles | No | Yes |
| Interlaced video support | Adaptive at macroblock level | Some profiles | Minimal Support | Yes, Main profile |
| Intra Prediction | DC only | Adaptive (DC&AC) | Some profiles | Adaptive Spatial |
| Transform size | 8x8 | 8x8 | 8x8 | 4x4 |
| Arbitrary shape coding | No | Some profiles | No | No |
| Dynamic picture resolution | No | No | Some profiles | No |
| Efficient quantizer coding | No | Yes | Some profiles | Yes |
| Adaptive arithmetic encoding | No | No, not for DCT | Some profiles | Yes (Main profile) |

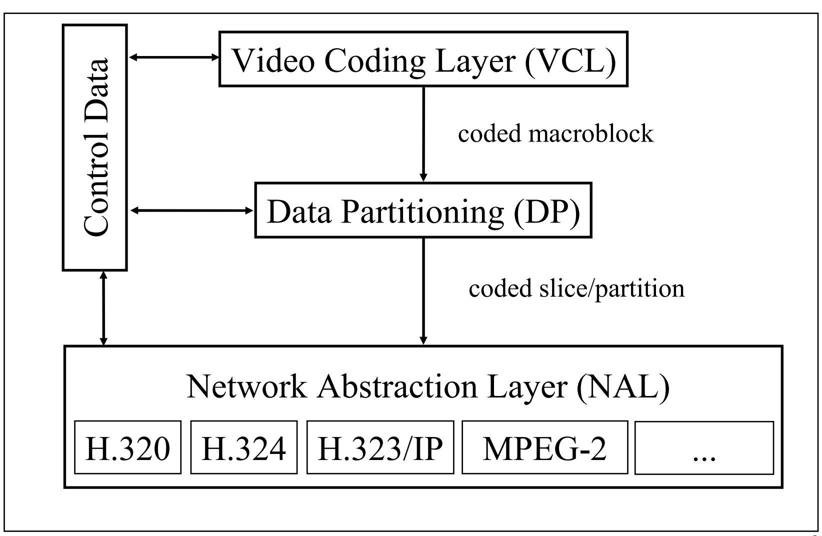
# Motion Compensation Tools

- Motion Estimation principle as in MPEG-1,2,4
- For a MB, Motion Estimation/Compensation of 16x16 Luma
- For Luma of a MB, Motion Estimation/Compensation of 16x8, 8x16 and 8x8 Luma blocks
- For 8x8 Luma block, Motion Estimation/Compensation of 8x4 4x8 and 8x8 blocks
- Picture boundary can be extended for Motion Compensation
- 1/4 pel motion compensation accuracy





# H.264/AVC Interfaces



# H.264/AVC Video Coding Layer

- I, P and B-pictures/Slices
- Intra Spatial Prediction
  - 4x4 Block Prediction, 9 modes, 8 prediction directions and mean
  - 16x16 Block Prediction, 4 modes, 3 prediction directions and mean
- Motion Compensated Prediction
  - Multiple Block sizes, e.g., 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4
  - Multiple Past Decoded Reference Frames for Prediction
  - Weighted Prediction
  - Quarter pixel accurate Motion Vectors
  - Motion Vectors differentially coded with Prediction (median or segmented)
- Transform Coding of Intra/Inter Blocks
  - 4x4 Block Integer Transform, an approximation of DCT





# H.264/AVC Video Coding (cont)

- Quantization and Scanning
  - Quantizer can be adapted on MB basis, quantization parameter in 0-51 range
  - Zig-zag scan and double scan
- In-loop Adaptive Deblocking Filter
- B-Pictures/Slices with Increased Flexibility
  - Normal B-pictures as well as those that predict other (P- or B-) pictures
  - Normal B-pictures as well as those with both references either in the past, or in the future
- Entropy Coding
  - Context Adaptive VLC Coding (CAVLC)
  - Context Adaptive Binary Arithmetic Coding (CABAC)
- Interlaced Video coding
  - Frame Pictures, including adaptive frame/field (AFF) coding
  - Field Pictures

# H.264/AVC Video Coding (cont)

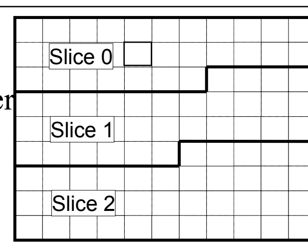
- Error Resilience Tools
  - Flexible Macroblock Ordering
  - Redundant Slices
  - Arbitrary Slice Ordering
  - Data Partitioned Slices
- Switching Pictures/Slices
  - SI and SP pictures/slices
- Network Access Layer (NAL)



U. of Rochester Apple Computer

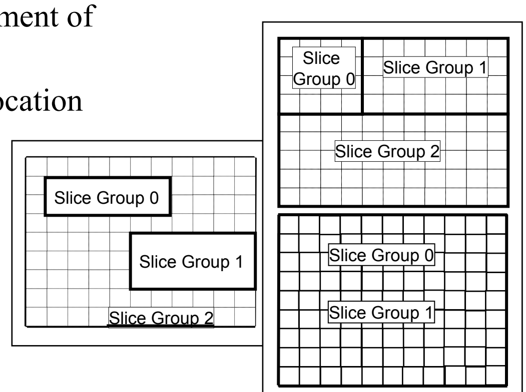
# H.264/JVT Video Data Structures

- H.264/JVT relies on the core concept of processing slices
- A picture may be partitioned into 1 or more slices
- Slice are self contained
- Three Primary slice types are: I-, P- and B-slices
- Two Secondary slice types are: SI- and SP-slices
- Slices are a sequence of macroblocks
- Macroblocks are as in previous standards
- Macroblocks of slice depend on each other
- Macroblocks can be partitioned further



# Flexible Macroblock Ordering (FMO)

- A slice group can contain one or more slices
- A macroblock allocation map can define a pattern of assignment of macroblocks to a slice
- Types of Macroblock allocation maps
  - Interleaved
  - Dispersed
  - Explicit
  - Leftover

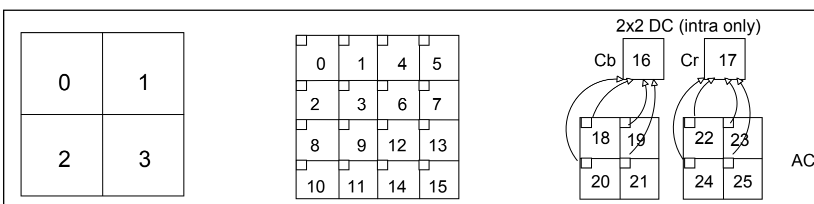







# Macroblock to Blocks Mapping

- Both for intra prediction as well as for residual coding a MB is considered as consisting of 4x4 blocks as follows



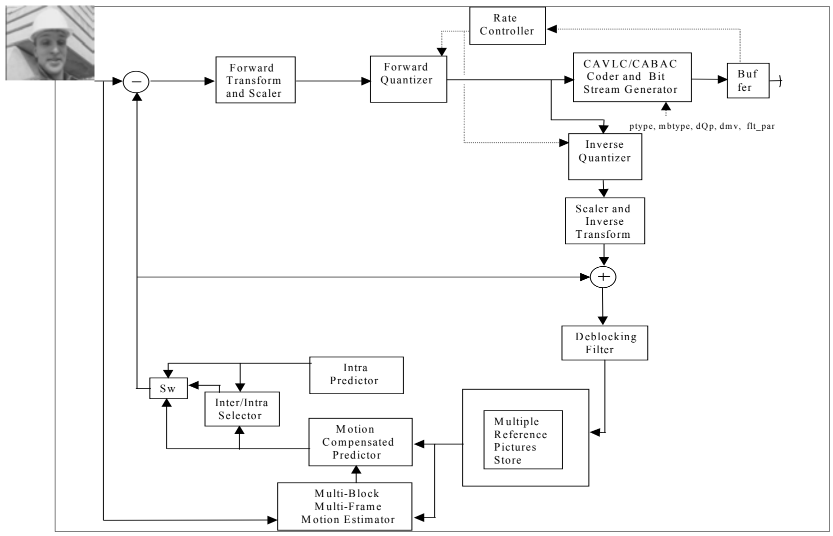

4 Luma 8x8 blocks and their order within a MB

16 Luma 4x4 blocks and their order within a MB

4 Chroma 4x4 blocks each and their order for Cb and Cr chroma components

U. of Rochester Apple Computer

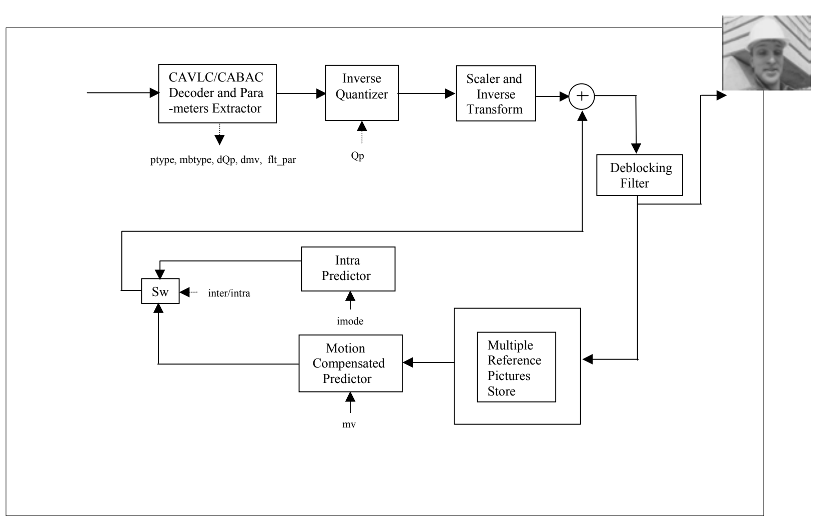
# H.264/AVC Encoder {NonNormative}





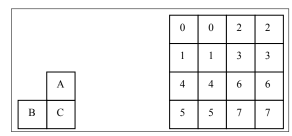

U. of Rochester Apple Computer

# H.264/AVC Decoder



# Intra Block Spatial Prediction

- 16x16 Macroblock Intra Spatial Prediction Modes
  - Mode 0: Vertical Prediction
  - Mode 1: Horizontal Prediction
  - Mode 2: DC Prediction
  - Mode 3: Planar Prediction
- 4x4 block Intra Spatial Prediction Modes
  - Mode 0: Vertical Prediction
  - Mode 1: Horizontal Prediction
  - Mode 2: DC Prediction
  - Mode 3: Diagonal Down/Left Prediction
  - Mode 4: Diagonal Down/Right Prediction
  - Mode 5: Vertical Left Prediction
  - Mode 6: Horizontal Down Prediction
  - Mode 7: Vertical Right Prediction
  - Mode 8: Horizontal Up Prediction
- Coding of 4x4 Intra Prediction Modes Info

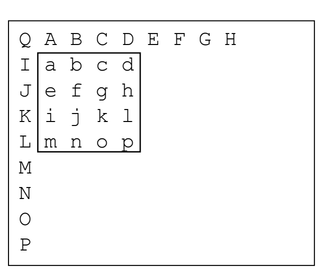


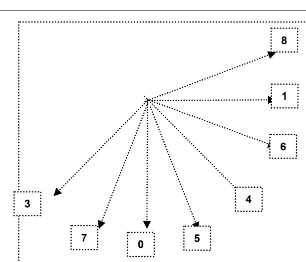





# Intra Block Spatial Prediction Details

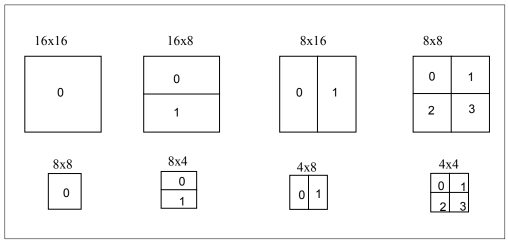
- 16x16 Intra Spatial Prediction Directions
  - DC: (A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P)/16
  - 3 Directions:
    - » Vertical:A, B,C,D,E,F,G, H to predict corresponding columns of pixels given by a,b,c,d,.,.,.,.
    - » Horizontal:I,J,K,L,M,N,O,P to predict corresponding rows of pixels given by a,e,i,m,.,.,.,.
    - » Planar
- 4x4 Intra Spatial Prediction Directions
  - DC: (A+B+C+D+I+J+K+L)/8
  - 8 Directions:
    - » Horizontal
      a=b=c=d=I
      e=f=g=h= J
      i=j=k=l= K
      m=n=o=p=L
    - » Vertical
      a=e=i=m=A
      b=f=j=n= B
      c=g=k=o=C
      d=h=l=p=D
    - » 6 Planar



# Motion Compensation

- Macroblock Partitioning for Motion Estimation/Compensation



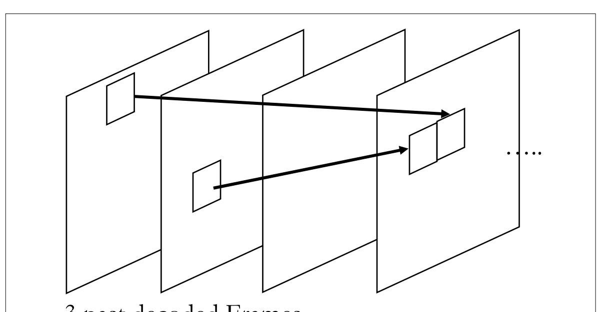

- Reference Frame for MC Prediction
  - can be selected on a 16x16, 16x8 or 8x16 basis
  - can be selected once for each 8x8 block if each block is coded in 8x8 subpartition mode



U. of Rochester Apple Computer

# Multiple Reference Motion Compensation

- Can use more than 1 past decoded frames for prediction




- Encoder informs the decoder which frames to keep in store
- Actual frame used as reference needs to be signaled to decoder as used for prediction on macroblock (or submacroblock basis)

# Quarter-Pel Motion Compensation

- Interpolation for 1/4 pel accuracy Motion Compensation
  - 'A' are original pixels
  - 'b' half sample positions are obtained first as intermediate values by applying filter {1,-5,20,20,-5,1} to nearest 'A pixels' in horizontal direction. Then, final value is found by dividing by 32, round, and clip to lie in 0-255 range
  - 'c' are obtained by applying {1,-5,20,20,-5,1}on intermediate values 'b' of closest 1/2 sample positions in either vertical or horizontal direction to get intermediate values of c. Final value is then found by dividing by 1024, round, clip to lie in 0-255 range
  - 'd', 'g', 'e' and 'f' by averaging with truncation, two nearest samples at integer or 1/2 sample position such as d=(A+b)/2, g=(b+c)/2, e=(A+b)/2, f=(b+c)/2
  - 1/4 sample positions 'h' by averaging with truncation two nearest 'b' samples in diagonal direction
  - 'i' (funny position) by averaging (A1+A2+A3+A4+2)/4 using four nearest original pixels

| | | | | |
|---|---|---|---|---|
| **A** | d | **b** | d | **A** |
| e | h | f | h | |
| **b** | g | **c** | g | **b** |
| e | h | f | i | |
| **A** | | **b** | | **A** |





# Block Transform: Basics Revisited

- Forward 1d Transform converts a 1d block of pixel data [S] into a 1d block of coefficients [C] by applying a transform basis matrix [T] (and its transpose $[T]^t$) as follows

  [C]=[T][S]

- Inverse 1d Transform converts a 1d block of coefficient data [C] into a 1d block of pixels [S] by applying

  $[S]=[T]^{-1}[C]$

  if $[T]^{-1} = [T]^t$ the basis matrix [T] is also called orthonormal, thus

  $[S]=[T]^t[C]$

- [T] is orthogonal if the leading diagonal in $[T][T]^t$ is nonzero while all other terms are zero.

# 2D Block Transform - More Basics and Examples

- To transform 2d pixel data, we consider 2d separable transforms only
- Forward 2d Transform converts a 2d block of pixel data [S] into a 2d block of coefficients [C] by applying basis matrix [T] (and its transpose $[T]^t$) as follows

  $[C]=[T][S][T]^t$
- Inverse 2d Transform converts a 2d block of coefficient data [C] into a 2d block of pixels [S] as follows

  $[S]=[T]^t[C][T]$
- Examples of 4x4 Block Transform Basis Matrices

WHT

$$T=\begin{matrix}1/2\\1/2\\1/2\\1/2\end{matrix}\begin{bmatrix}1 & 1 & 1 & 1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\\1 & -1 & 1 & -1\end{bmatrix}$$

Slant Transform

$$T=\begin{matrix}1/2\\1/2\sqrt{5}\\1/2\\1/2\sqrt{5}\end{matrix}\begin{bmatrix}1 & 1 & 1 & 1\\3 & 1 & -1 & -3\\1 & -1 & -1 & 1\\1 & -3 & 3 & -1\end{bmatrix}$$

DCT

$$T=\begin{bmatrix}.50 & .50 & .50 & .50\\.65 & .27 & -.27 & -.65\\.50 & -.50 & -.50 & .50\\.27 & -.65 & .65 & -.27\end{bmatrix}$$





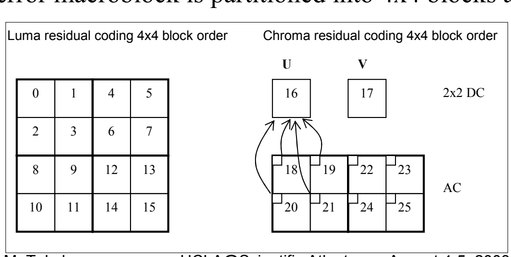
# 2D Block Transform of Size 4x4 Used

- A (Slant like) approx of 4x4 DCT but with coefficients that are powers of 2
- 4x4 Basis Matrix used by AVC/JVT

$$T=\begin{matrix}1/2\\1/\sqrt{10}\\1/2\\1/\sqrt{10}\end{matrix}\begin{bmatrix}1 & 1 & 1 & 1\\2 & 1 & -1 & -2\\1 & -1 & -1 & 1\\1 & -2 & 2 & -1\end{bmatrix}$$

where, the scaling constants per row are absorbed in the quantization process

- Prediction error macroblock is partitioned into 4x4 blocks and 4x4 transformed



# Quantization

- Quantization Features
  - Logrithmic step size control
  - Extended range of step sizes (52 values rather than 31)
  - Smaller step size for chroma
  - Step size can be changed to any value at macroblock level
  - Step size prediction difference is coded
- There are 52 values for QP in range [0-51]. Scaling equations are designed so that parameter doubles for every 6 increase in Qp. There is an increase in scaling magnitude of approx 12% from one Qp to next
- Chroma QP, $QP_C$ is derived from luma Qp, $QP_Y$ as follows:

| $QP_Y$: | <30 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $QP_C$: | $= QP_Y$ | 29 | 30 | 31 | 32 | 32 | 33 | 34 | 34 | 35 | 35 | 36 | 36 | 37 | 37 | 37 | 38 | 38 | 38 | 39 | 39 | 39 | 39 |



**U. of Rochester** **Apple Computer**

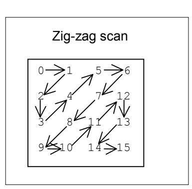
# Quantization (cont) and Coefficient Scanning

- Scaling for forward/inverse transform is performed together with quantization/dequantization. The entire operation is designed to be handled in 16 bit integer arithmetic
- Approximate relationship of AVC/JVT QP to MPEG-4 Qp is as follows

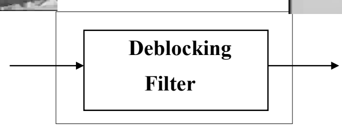
$$QP_{MPEG\text{-}4} \sim= 2^{(QP - 12)/6}$$

where $QP_{MPEG\text{-}4}$ is the MPEG-4's QP, and QP is the AVC/JVT's QP

- 4x4 Blocks of transform coefficients are scanned by zig-zag scan



# Deblocking Filter: Principle





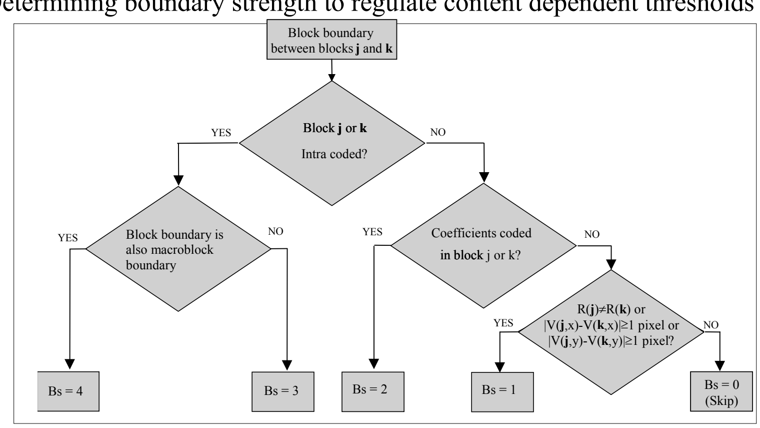



# Deblocking Filter : Details

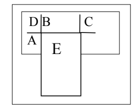
- Picture content dependent deblocking filter is used in the loop to reduce blockiness artifacts

$p_4,p_3,p_2,p_1 \mid q_1,q_2,q_3,q_4$

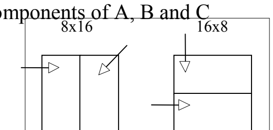
- Determining boundary strength to regulate content dependent thresholds



# Motion Vector Prediction




- Median Prediction of MV
  - prediction of A, B, C, D, E may use different reference pictures
  - if C is outside picture or slice, it assumes mv and ref pic index of D
  - if B,C,D are outside picture or slice, they assume mv and ref pic index of A
  - if a predictor not specified above is coded as intra, mv is 0 and ref pic index different from E
  - if only one of A,B and C has same ref picture as E, then its mv is used for prediction of E, otherwise, each component of pred mv of E is median of mv components of A, B and C




- Segmented Directional Prediction

  Vector Block size 8x16
  - left block, A is used as prediction if it has same ref picture as E, otherwise median prediction
  - right block, C is used as prediction it it has same ref picture as E, otherwise median prediction

  Vector Block size 16x8
  - upper block, A is used as prediction if it has same ref picture as E, otherwise median prediction
  - lower block, C is used as prediction if it has same ref picture as E, otherwise median prediction





# Macroblock Types in P-slices : semantics

- **mb_type** indicates macroblock types. For P slices basic values range from 0 to 5, but with intra modes, additional values 6,30 are needed for intra

**Table:** Macroblock Types 0 to 4 for P and SP slices

| value | mb_type | num_mb_ partition() | mb_partition_ pred_mode(,1) | mb_partition_ pred_mode(,2) |
|---|---|---|---|---|
| 0 | MbSkip | 1 | Direct | |
| 1 | Pred_L0_16x16 | 1 | Pred_L0 | |
| 2 | Pred_L0_L0_16x8 | 2 | Pred_L0 | Pred_L0 |
| 3 | Pred_L0_L0_8x16 | 2 | Pred_L0 | Pred_L0 |
| 4 | Pred_8x8 | 4 | na | na |
| 5 | Pred_8x8ref0 | 4 | na | na |

- Semantics
  - MbSkip is a macroblock for which no further information is transmitted
  - Pred_L0_16x16, Pred_L0_L0_16x8, Pred_L0_L0_8x16, Pred_8x8: macroblock predicted from past picture with luma block sizes 16x16, 16x8, 8x16 and 8x8. For 16x16, 16x8, 8x16, a motion vector is present for each NxM luma block, where as for 8x8 block additional (sub_mb_type) syntax is decoded
  - Pred_8x8ref0: same as Pred_8x8 but ref_idx_l0 not sent, set to 0 for all submacroblocks

# Submacroblock Types in P-slices: semantics

- **sub_mb_type** indicates sub macroblock types

**Table:** Submacroblock Types for P slices

| value | sub_mb_type | num_sub_mb_partition() | sub_mb_pred_mode() |
|---|---|---|---|
| 0 | Pred_L0_8x4 | 2 | Pred_L0 |
| 1 | Pred_L0_4x8 | 2 | Pred_L0 |
| 2 | Pred_L0_4x4 | 4 | Pred_L0 |
| 3 | Intra_8x8 | na | Intra |

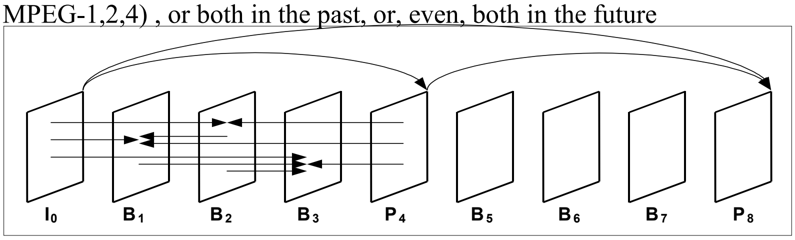
- Semantics
  - Pred_L0_8x4, Pred_L0_4x8, Pred_L0_4x4: corresponding partition of submacroblock is predicted from past picture with luma block sizes of 8x4, 4x8 and 4x4. A motion vector is transmitted for each NxM block. Depending on N and M, up to 4 motion vectors may be decoded for a submacroblock, and up to 16 motion vectors for a macroblock.
  - Intra_8x8: 8x8 subpartition is coded in intra mode. Not present in SPred slices.
- B-pictures/slices use a longer table of submacroblock types with 14 entries





# B-slices/Pictures

- Prediction References can be one in the past and one in the future (as in MPEG-1,2,4) , or both in the past, or, even, both in the future




- 5 prediction types: Direct, forward, backward, bi-directional and intra
- Bidirectional mode sends explicit motion vectors (as do forward/backward modes ) while direct mode uses scaled motion vectors of colocated macroblock
- One reference frame parameter for each 16x16, 16x8, 8x16, as well as for each 8x8 subpartition. Additionally for each 16x16, 16x8, 8x16, and 8x8 subpartition, prediction type can be chosen separately

# Macroblock Types in B-slices

**Table:** Macroblock Types 0 to 22 for B slices

| value | mb_type | num_mb_ partition() | mb_partition_ pred_mode(,1) | mb_partition_ pred_mode(,2) |
|---|---|---|---|---|
| 0 | Direct_16x16 | 1 | Direct | |
| 1 | Pred_L0_16x16 | 1 | Pred_L0 | |
| 2 | Pred_L1_16x16 | 1 | Pred_L1 | |
| 3 | BiPred_Bi_16x16 | 1 | BiPred | |
| 4 | Pred_L0_L0_16x8 | 2 | Pred_L0 | Pred_L0 |
| 5 | Pred_L0_L0_8x16 | 2 | Pred_L0 | Pred_L0 |
| 6 | BiPred_L1_L1_16x8 | 2 | Pred_L1 | Pred_L1 |
| 7 | BiPred_L1_L1_8x16 | 2 | Pred_L1 | Pred_L1 |
| 8 | BiPred_L0_L1_16x8 | 2 | Pred_L0 | Pred_L1 |
| 9 | BiPred_L0_L1_8x16 | 2 | Pred_L0 | Pred_L1 |
| 10 | BiPred_L1_L0_16x8 | 2 | Pred_L1 | Pred_L0 |
| 11 | BiPred_L1_L0_8x16 | 2 | Pred_L1 | Pred_L0 |
| 12 | BiPred_L0_Bi_16x8 | 2 | Pred_L0 | BiPred |
| 13 | BiPred_L0_Bi_8x16 | 2 | Pred_L0 | BiPred |
| 14 | BiPred_L1_Bi_16x8 | 2 | Pred_L1 | BiPred |
| 15 | BiPred_L1_Bi_8x16 | 2 | Pred_L1 | BiPred |
| 16 | BiPred_Bi_L0_16x8 | 2 | BiPred | Pred_L0 |
| 17 | BiPred_Bi_L0_8x16 | 2 | BiPred | Pred_L0 |
| 18 | BiPred_Bi_L1_16x8 | 2 | BiPred | Pred_L1 |
| 19 | BiPred_Bi_L1_8x16 | 2 | BiPred | Pred_L1 |
| 20 | BiPred_Bi_Bi_16x8 | 2 | BiPred | BiPred |
| 21 | BiPred_Bi_Bi_8x16 | 2 | BiPred | BiPred |
| 22 | BiPred_8x8 | 4 | na | |





# Submacroblock Types in B-slices

**Table:** Submacroblock Types for B slices

| value | sub_mb_type | num_sub_mb_partition() | sub_mb_pred_mode() |
|---|---|---|---|
| 0 | Direct_8x8 | 1 | Direct |
| 1 | Pred_L0_8x8 | 1 | Pred_L0 |
| 2 | Pred_L1_8x8 | 1 | Pred_L1 |
| 3 | BiPred_Bi_8x8 | 1 | BiPred |
| 4 | Pred_L0_8x4 | 2 | Pred_L0 |
| 5 | Pred_L0_4x8 | 2 | Pred_L0 |
| 6 | BiPred_L1_8x4 | 2 | Pred_L1 |
| 7 | BiPred_L1_4x8 | 2 | Pred_L1 |
| 8 | BiPred_Bi_8x4 | 2 | BiPred |
| 9 | BiPred_Bi_4x8 | 2 | BiPred |
| 10 | Pred_L0_4x4 | 4 | Pred_L0 |
| 11 | BiPred_L1_4x4 | 4 | Pred_L1 |
| 12 | BiPred_Bi_4x4 | 4 | BiPred |
| 13 | Intra_8x8 | 1 | Intra |

# Macroblock Prediction: Semantics

- **ref_idx_l0** if num_ref_idx_l0_active_minus1 indicates possibility of prediction from more than one reference picture, the exact picture needs to be indicated

| code_num | Reference frame |
|---|---|
| 0 | First Frame in the forward reference set |
| 1 | Second frame in the forward reference set |
| 2 | Third frame in the forward reference set |
| : | |

  if num_ref_idx_l0_active_minus1 is 0, ref_idx_l0 is not present
  If num_ref_idx_l1_active_minus1 is 0, ref_idx_l1 is not present
  Reference index parameter is sent for each 16x16, 16x8, 8x16 MC block. If MB is coded in 8x8 mode, ref frame parameter sent every 8x8, unless 8x8 coded in Intra mode
- **ref_idx_l1** similar to ref_idx_l0 except that it applies to the second reference index list
- **mvd_l0** if indicated by mb_type, vector data for 1-16 blocks is sent. Prediction is formed for horizontal and vertical components of motion vector. mvd_l0 signals the difference between vector and prediction. Motion vectors are allowed to point to outside reference frame (use edge or corner samples for extrapolation). Reconstructed mv clipped to +-19 integer samples outside the frame
- **mvd_l1** similar to mvd_l0 except that it applies to second reference index list





# Slice and Lower Layer Syntax Flow

- slice_data()
  - mb_skip_run/mb_skip_flag and MoreDataFlag
  - macroblock_layer()
- macroblock_layer()
  - mb_type
  - mb_pred() or sub_mb_pred()
  - coded_block_pattern, SendResidual
  - delta_qp and residual()
- mb_pred()
  - intra_pred_mode
  - intra_chroma_mode
  - ref_idx_l0
  - ref_idx_l1
  - mvd_l0[i][0..1]
  - mvd_l1[i][0..1]

# Slice and Lower Layer Syntax Flow (cont)

- sub_mb_pred(mb_type)
  - sub_mb_type[0..3]
  - intra_pred_mode
  - intra_chroma_mode
  - ref_idx_l0
  - ref_idx_l1
  - mvd_l0[0..3][j][0..1]
  - mvd_l1[0..3][j][0..1]
- residual(mb_type)
  - residual_4x4block() is either residual_4x4block_cavlc() or residual_4x4block_cabac()
  - residual_4x4block(intra16x16DC,16)
  - residual_4x4block(intra16x16AC, 16)
  - residual_4x4block(luma, 16)
  - residual_4x4block(chromaDC, 4)
  - residual_4x4block(chromaAC,16)





# VLC Coding of Overhead - Goulomb Codes

- In Goulomb Codes (UVLC), codewords have a regular structure and are generated according to a rule, e.g., where $x_n$= 0 or 1

```
        1
      0 1 x0
    0 0 1 x1 x0
  0 0 0 1 x2 x1 x0
0 0 0 0 1 x3 x2 x1 x0
 .................
```

- VLC Table

Table A: Unsigned Symbols

| Code_number | Code word |
|---|---|
| 0 | 1 |
| 1 | 0 1 0 |
| 2 | 0 1 1 |
| 3 | 0 0 1 0 0 |
| 4 | 0 0 1 0 1 |
| 5 | 0 0 1 1 0 |
| 6 | 0 0 1 1 1 |
| 7 | 0 0 0 1 0 0 0 |
| 8 | 0 0 0 1 0 0 1 |
| 9 | 0 0 0 1 0 1 0 |
| 10 | 0 0 0 1 0 1 1 |
| ...... | . . . . . . . |

Table B: Mapping of Signed Symbols

| Code_number | Signed Symbol |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | -1 |
| 3 | 2 |
| 4 | -2 |
| 5 | 3 |
| 6 | -3 |
| 7 | 4 |
| 8 | -4 |
| 9 | 5 |
| 10 | -5 |
| ...... | . . . . . . . |

# Context Adaptive VLC (CAVLC)- Coef Coding

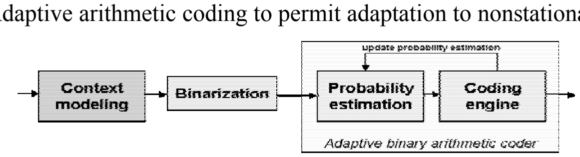
- Transform coefficients are Adaptive VLC coded
  - No EOB number of coefs is decoded, coefs backward scanned, uses context
  - Number of Nonzero Coefficients, Levels and signs of all Nonzero Coefficients, Total number of zeros before last Nonzero Coefficient, Run before each Nonzero Coefficient
- Number of Coefficients/Trailing '1s'
  - Last non-zero coefficients have |level| of 1
  - Number of Nonzero coefficients (e.g., 6) and number of trailing 1's (2) are coded together as a combined symbol, typically 50% of coefficients can be coded as trailing 1s and no other level info than sign is needed
  - VLC table selected depends on number of coefficients in neighbor blocks
- Reverse Scanning and Level Coding
  - Statistics of last nonzero coef (trailing 1s) more stable than first (high value)
  - Reverse scan, default VLC for first coef, use lvl context to select next table
- Run info - TotalZeros and RunBefore
  - TotalZeros:Total number of zeros before last nonzero coef in forward scan
  - RunBefore:In reverse scan, run before each nonzero coef is coded



U. of Rochester Apple Computer

# Context Adaptive Binary Arithmetic Coding

- Context-based Adaptive Binary Arithmetic Coding (CABAC), involves
  - Context modeling to provide estimate of conditional probabilities of the symbols
  - Arithmetic codes to permit noninteger number of bits to be assigned to each symbol of the alphabet
  - Adaptive arithmetic coding to permit adaptation to nonstationary symbol statistics




- Context modeling for coding of Motion and Mode information
  - Context models for Macroblock types
  - Context models for Motion vector data
  - Context models for Reference Frame parameters

# CABAC Arithmetic Coding (cont.)

- Context modeling for coding of Texture data
  - Context models for Coded Block Patterns
  - Context models for Intra prediction
  - Context model for Run/Level and coefficient count
    - » context based coding of coefficient count
    - » context based coding of
    - » context based coding of level info
- Double Scan always for CABAC intra mode
- Context modeling for coding of Dquant
- Binarization of nonbinary valued signals
- Adaptive Binary Arithmetic coding



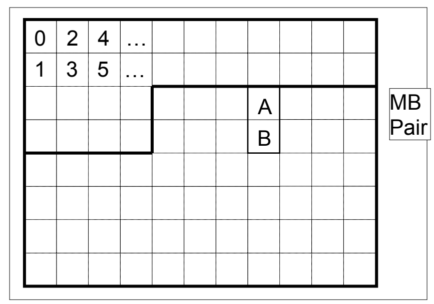


# Additional Tools

- Interlaced Video Coding Efficiency Tools
- S-pictures, a Tool for Stream Switching
- Efficient Memory Management Tools
- Adaptation for Network Layer

# H.264/JVT Interlaced Video Data Structures

- Field Coding
  - Consider each field as a separate picture for coding
- Frame Coding
  - Type 1: Consider each frame as a separate picture
  - Type 2: Scan each frame as pairs of macroblocks. For each macroblock pair either frame or field coding can be selected

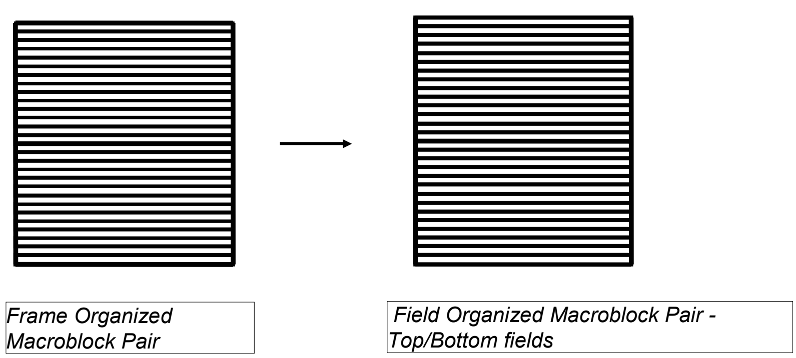






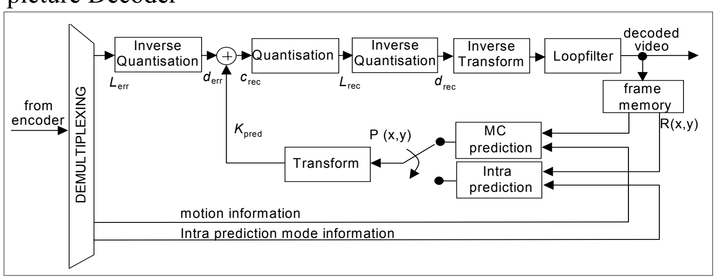

# Interlaced Video Frame Coding

- Macroblock Based Adaptive Frame/Field (MBAFF) Coding



# Switching-Slices/Pictures (SI and SP)

- Switching pictures intended for bitstream switching, splicing, random access, VCR functionalities such as fast forward, error resilience/recovery
- SI-pictures uses spatial prediction (like I-pictures), and SP-pictures uses motion compensated prediction temporal redundancy (like P-pictures)
- SP-pictures unlike P-pictures allow identical reconstruction of a frame even when different reference frames are being used
- S-picture Decoder








# Reference Memory Management (RMM): Terms

- **reference picture**: picture containing samples used in inter prediction
- **reference picture buffer**: buffer for reference pictures
- **reference index**: is a relative index into list of reference to identify which picture out of reference picture buffer is used for motion compensation.
- **reference index list**: list of indices assigned to reference pictures in reference picture buffer
- **reference index list 0**: list of reference indices for use in inter prediction in P, B or SP slice
- **reference list 0 motion vector**: a motion vector associated with reference index pointing to reference index list 0
- **reference list 0 prediction**: inter prediction of content of a slice using reference index into reference index list 0
- **reference index list 1**: list of reference indices for use in inter prediction for B slice
- **reference index list 1 motion vector**: a motion vector associated with reference index pointing to reference index list 1
- **reference list 1 prediction**: inter prediction of a B slice using reference index into reference index list 1
- **non-reference picture**: a decoded picture that is not used for inter prediction
- **coded pictures input buffer**: FIFO containing coded pictures in decoding order
- **decoded picture**: obtained by decoding a coded picture, and is either a decoded frame or decoded field
- **decoded pictures buffer**: buffer specified in video buffering verifier (reference picture + reordering buffer)
- **memory management control operations**: specified in coded data that decoding process applies to decoded pictures buffer
- **picture order count**: picture position in output order relative to latest IDR picture in decoding order
- **picture reordering**: re-ordering of pictures when decoding order is different from output order
- **reference field**: used for inter prediction
- **reference frame**: used for inter prediction

# Reference Picture Buffer Management

- **ref_pic_buffering_mode** specifies buffering mode of current decoded pictures and specifies how picture buffer is modified after current picture is decoded

**Table** Reference picture Buffering

| value | Reference picture buffering |
|---|---|
| 0 | Sliding window buffering; a simple FIFO for pictures without a long term index |
| 1 | Adaptive buffering; mark pictures as unused, assign long term index, reset ref pic buffer |

- **memory_management_control_operation** is a command that specifies the operation to be applied to manage reference picture buffer
  - if memory_management_control_operation is Reset, all frames in reference picture buffer shall be marked as unused
  - frame height and width shall not change within bitstream except within a frame containing Reset memory_management_control_operation
  - a stored frame shall not contain memory management_control_operation command which marks it unused. If current frame is a non-stored frame, a memory_management_control_operation cannot contain following commands
    - » a reset memory_management_control_operation command
    - » any memory_management_control_operation command that marks a frame as unused that has not also been marked unused in RPS layer
    - » any memory_management_control_operation command that assigns long-term index to a frame that has not also been assigned the same long term index in RPS layer





# Reference Picture Buffer Management (cont)

**Table** Memory management control operations

| value | Memory Management Control Operation | Associated data fields that follow |
|---|---|---|
| 0 | End memory_management_control_operation loop | None |
| 1 | Mark a short term frame as "unused" | difference_of_pic_numbers |
| 2 | Mark a long term frame as "unused" | long_term_pic_idx |
| 3 | Assign a long term index to a frame | difference_of_pic_nums and long term_pic_idx |
| 4 | Specify maximum long term frame index | max_long_term_pic_idx_plus1 |
| 5 | Reset | none |

- **difference_of_pic_nums** is used in order to assign a long term index to a frame, or to mark a short term frames as "unused"
- **long_term_pic_idx** represents long term picture index to be remapped. Follows difference_of_pic_nums in case of assigning a long term index to a picture
- **max_long_term_pic_idx_plus1** determines maximum index allowed for long term reference frames (till receipt of max_long_term_pic_idx_plus1). Decoder initially assumes max_long_term_pic_idx_plus1 is 0 till some other value is received. Upon receiving a max_long_term_pic_idx_plus1, decoder considers all long term frames having indices greater than decoded value of max_long_term_pic_idx_plus1 -1 as "unused". Status of all other frames in reference picture buffer is unchanged.

# NAL and File Format

- Network Adaptation Layer (NAL)
  - A byte stream NAL format is defined that consists of a sequence of packets prefixed as: zero stuffing bytes, 3 byte start code prefix, payload type indicator byte, one or more EBSP's as indicated by payload type
  - NAL layer takes information from interim file format and converts it directly into packets that can be conveyed directly over RTP. I t has to arrange packets intelligently, split/recombine partitions to match MTU size constraints, avoid redundancy of RTP header, and specify action in case of packet loss
- File Format
  - An interim file format defined based om ISO MPEG-4's file format
  - File contains boxes, each box can contain other boxes
  - Boxes may contain member attributes, if a box contains attributes, other boxes shall follow after the attributes
  - A number of boxes contain index values into sequences in other boxes
  - File format is specified in MPEG-4 Syntax Description Language





# Video Encoding Issues *{NonNormative}*

- I,P,B picture structure selection
- B-frames
  - Number of B-frame and structure selection
  - Stored or Nonstored B-frames
  - Spatial Direct vs Temporal Direct
- Motion Estimation/Compensation and Reference Frames
  - Reduced or Full search
  - MB partitions selection
  - Number of Reference frames for Motion Compensation
  - Reference frame selection for Motion Compensation
  - SAD vs SATD
- Mode Decision *(important!!)*
  - Non RD based (low complexity)
  - RD Optimization using Lagrange (high complexity)

# Video Encoding Issues (cont) *{NonNormative}*

- Quantization and Coefficient Thresholding
  - Without or with perceptual basis
  - Elimination of single coefficients in Inter MB: Luma and Chroma
- Rate Control
  - Picture Based
  - MB Based
- Loop Filter
  - Default
  - Send Control Parameters
- Error Resilient and Stream Switching Coding
  - FMO, ASO,..
  - Reference Frame updates and resend
  - Switched (SI and SP) pictures
- Others specific to Profile selected (e.g. CABAC initialization)





# Motion Est/Mode Decision: Low Complexity Method

- Steps in Prediction Mode decision
  - SA(T)D0: initial bias to favor prediction modes requiring fewer bits
  - Block Difference: block difference between original and prediction
  - SAD, or SATD (using Hadammard Transform): for intra and fractional pel, use SATD
  - Choose Prediction mode resulting in minimum $SA(T)D_{min}$
- Intra and Inter Modes
  - Intra Mode
    - Code MB in intra mode with 4x4 blocks and prediction, calculate $SATD_{intra}$
    - Table for Intra Prediction Modes for use by Encoder: Inverse of table used by decoder
  - Inter Mode
    - Search 7 block structures and 5 past decoded frames, resulting in 35 combinations of block sizes and reference frames
    - Integer pixel Search: Use spiral search around predicted vector. Use full range for 16x16, but half range for other MC block sizes. Use quarter range for older pictures
    - Fractional pixel Search: First, search 8 half-pel positions around best integer vecor, then search for 8 quarter-pel positions around the best half-pel position. Calculate $SATD_{inter}$
  - Decision between Intra and Inter Modes: If $SATD_{intra} < SATD_{inter}$, use intra coding

# Motion Est/Mode Decision: High Complexity Method

- Motion Estimation Steps
  - Integer-pel search
    - As in low complexity mode, use spiral search around prediction vector
    - Use full range for all block sizes and reference frames
    - To speed search use 16x16 block MC as the center for spiral search
  - Fractional-pel search
    - As in low complexity mode
  - Best motion vector: integer/sub-pel refinement minimizes $SA(T)D(s,c(\mathbf{m})) + \Lambda_{motion}R(\mathbf{m}-\mathbf{p})$
  - Finding best ref: minimizes $SATD(s,c(ref, \mathbf{m}(ref))) + \Lambda_{motion}(R(\mathbf{m}(ref)-\mathbf{p}(ref)) + R(ref))$

s: original video
c: coded video
**m**: motion vector
**p**: prediction motion vector
R(): Rate term
Λ : Lagrangian multiplier
SSD: Sum of square difference

- Mode Decision
  - Macroblock Mode Decision: minimize $SSD(s,c,mode|Qp) + \Lambda_{mode}R(s,c,mode|Qp)$
  - Intra16x16: Choose mode resulting in minimum SATD value
  - Intra4x4: as in MB mode decision, minimize $SSD(s,c,Imode|Qp) + \Lambda_{mode}R(s,c,Imode|Qp)$
- Steps in Motion Estimation and Prediction Mode Decision
  - Given previous decoded frames, $\Lambda_{mode}$, $\Lambda_{motion}$, and MB quantizer Qp
  - INTRA4x4 MB mode, choose prediction for each 4x4 from 9 possible choices
  - INTRA16x16 MB mode, choose prediction for 16x16 from 4 possible choices
  - Perform motion estimation and ref frame selection by minimizing motion cost function
  - Choose MB prediction mode by minimizing mode cost function





# Issues in Migration to H.264/AVC

- Significant coding improvements. Are they application specific?
  – Using parameters of desired application, detailed independent study may be needed to actually determine its cost/benefits for that application
- Encoder development. Tradeoffs with respect to MPEG-4?
  – Encoder complexity can be very high (10 x MPEG-2)
  – Encoder algorithm development challenging, expensive and time consuming
  – Encoder development cycle can be long.
- MPEG-4 vs JVT Profile Choices. Benefits of economies of scale ?
  – Check if tradeoffs it offers addresses desired goals of your application
  – Cost of incremental upgrade vs replacement of hardware/software
- Decoder development & deployment. Flexibility, testing costs ?
  – Cost of development of decoder hardware/software vs flexibility.
  – Testing and coinformance issues in decoder deployment
- Look at: Profiles, Results, Demo, Resources, Licensing costs, next.

# H.264/AVC Profiles Detailed Comparison

| Tool | Profile: Baseline | Extended | Main |
|---|---|---|---|
| I, P pictures/slices | yes | yes | yes |
| 1/4 sample MC | yes | yes | yes |
| MC up to 4x4 Block size | yes | yes | yes |
| Multiple Reference Pictures | yes | yes | yes |
| 4x4 Block Transform | yes | yes | yes |
| In Loop Deblocking Filter | yes | yes | yes |
| Adaptive VLC Entropy Coding | yes | yes | yes |
| Redundant Slices | yes | yes | no |
| Arbitrary Slice Ordering | yes | yes | no |
| Flexible Macroblock Ordering | yes | yes | no |
| B-pictures/slices | no | yes | yes |
| Weighted Prediction | no | yes | yes |
| CABAC Arithmetic Entropy Coding | no | no | yes |
| SI/SP Slices | no | yes | no |
| Data Partitioned Slices | no | yes | no |
| Interlaced Frame/Field Video Coding | no | yes | yes |





# H.264/AVC Levels in Profiles

| Level | Max MB Rate, MB/s | Max Frm Size, MB | Pic Buf, Kbytes | Max Video Bitrate kbit/s | Horiz MV Range | Vert MV Range |
|---|---|---|---|---|---|---|
| 1 | 1485 | 99 | 148.5 | 64 | [-2048,2047.75] | [-64,63.75] |
| 1.1 | 3000 | 396 | 337.5 | 192 | [-2048,2047.75] | [-128,127.75] |
| 1.2 | 6000 | 396 | 891.0 | 384 | [-2048,2047.75] | [-128,127.75] |
| 1.3 | 11880 | 396 | 891.0 | 768 | [-2048,2047.75] | [-128,127.75] |
| 2 | 11880 | 396 | 891.0 | 2000 | [-2048,2047.75] | [-128,127.75] |
| 2.1 | 19800 | 792 | 1782.0 | 4000 | [-2048,2047.75] | [-256,255.75] |
| 2.2 | 20250 | 1620 | 3037.5 | 4000 | [-2048,2047.75] | [-256,255.75] |
| 3 | 40500 | 1620 | 3037.5 | 10000 | [-2048,2047.75] | [-256,255.75] |
| 3.1 | 108000 | 3600 | 6750.0 | 14000 | [-2048,2047.75] | [-512,511.75] |
| 3.2 | 216000 | 5120 | 7680.0 | 20000 | [-2048,2047.75] | [-512,511.75] |
| 4 | 245760 | 8192 | 12288.0 | 20000 | [-2048,2047.75] | [-512,511.75] |
| 4.1 | 245760 | 8192 | 12288.0 | 50000 | [-2048,2047.75] | [-512,511.75] |
| 4.2 | 491520 | 8192 | 12288.0 | 50000 | [-2048,2047.75] | [-512,511.75] |
| 5 | 589824 | 22080 | 41310.0 | 135000 | [-2048,2047.75] | [-512,511.75] |
| 5.1 | 983040 | 36864 | 69120.0 | 240000 | [-2048,2047.75] | [-512,511.75] |

# **Comparison of Standards -** Videotelephony

- Published results show significant benefits of H.264/AVC (T. Wiegand et al, IEEE CSVT, July 2003). However, using parameters of your desired application, perform careful, independent evaluation as *your mileage may vary*.

| | H.263 Baseline | | | | H.263 CHC | | | | MPEG-4 SP | | | | H.264/AVC Baseline | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sequence | Rate | PSNR-Y | PSNR-U | PSNR-V | Rate | PSNR-Y | PSNR-U | PSNR-V | Rate | PSNR-Y | PSNR-U | PSNR-V | Rate | PSNR-Y | PSNR-U | PSNR-V |
| **A: QCIF, 10 Hz, 24 kbit/s** | | | | | | | | | | | | | | | | |
| Akiyo | 24.14 | 37.34 | 39.73 | 41.31 | 24.06 | 38.54 | 41.89 | 42.93 | 24.19 | 38.01 | 40.24 | 41.95 | 24.00 | 40.68 | 42.90 | 43.58 |
| Foreman | 24.21 | 27.73 | 35.39 | 34.95 | 24.25 | 28.52 | 37.39 | 37.37 | 24.09 | 29.10 | 36.27 | 35.95 | 23.87 | 30.08 | 37.45 | 37.58 |
| Mother & Daughter | 23.78 | 31.27 | 36.49 | 36.32 | 23.82 | 31.68 | 37.80 | 37.65 | 23.97 | 31.75 | 36.62 | 36.37 | 24.08 | 33.19 | 37.96 | 37.71 |
| Silent | 24.08 | 31.12 | 35.44 | 36.93 | 23.90 | 32.31 | 37.28 | 38.83 | 24.14 | 31.68 | 35.51 | 37.02 | 24.09 | 32.42 | 36.34 | 38.07 |
| **B: QCIF, 15 Hz, 32 kbit/s** | | | | | | | | | | | | | | | | |
| Akiyo | 32.31 | 37.93 | 40.53 | 41.87 | 32.05 | 38.68 | 41.97 | 42.98 | 31.76 | 38.62 | 41.12 | 42.60 | 32.07 | 41.15 | 43.22 | 43.95 |
| Foreman | 31.78 | 28.17 | 35.38 | 35.01 | 32.10 | 28.66 | 37.39 | 37.34 | 32.13 | 29.35 | 36.19 | 36.13 | 32.37 | 30.51 | 37.58 | 37.60 |
| Mother & Daughter | 31.77 | 31.56 | 36.62 | 36.49 | 31.74 | 31.87 | 37.81 | 37.61 | 32.27 | 31.96 | 36.73 | 36.70 | 32.14 | 33.66 | 37.99 | 37.81 |
| Silent | 31.79 | 31.21 | 35.46 | 36.90 | 31.88 | 32.58 | 37.58 | 38.89 | 31.97 | 31.95 | 35.74 | 37.39 | 32.18 | 32.47 | 36.45 | 38.04 |
| **C: CIF, 15 Hz, 128 kbit/s** | | | | | | | | | | | | | | | | |
| Carphone | 129.71 | 31.53 | 35.94 | 37.03 | 127.64 | 32.32 | 38.02 | 39.24 | 127.82 | 32.50 | 36.62 | 37.73 | 125.64 | 33.50 | 37.75 | 39.23 |
| Foreman | 128.32 | 29.92 | 36.40 | 37.00 | 127.97 | 30.76 | 38.50 | 39.39 | 128.65 | 31.52 | 37.71 | 38.45 | 127.24 | 32.96 | 38.77 | 40.06 |
| Paris | 127.38 | 28.30 | 33.30 | 33.84 | 128.29 | 29.34 | 35.56 | 36.32 | 127.95 | 29.18 | 33.59 | 34.25 | 128.52 | 30.81 | 35.80 | 36.18 |
| Sean | 129.74 | 36.64 | 40.56 | 41.07 | 128.47 | 37.91 | 41.71 | 42.29 | 127.37 | 36.75 | 40.50 | 41.31 | 129.89 | 39.46 | 42.22 | 43.05 |
| **D: CIF, 30 Hz, 256 kbit/s** | | | | | | | | | | | | | | | | |
| Carphone | 258.89 | 32.47 | 36.35 | 37.54 | 256.20 | 33.31 | 38.20 | 39.62 | 256.71 | 33.34 | 36.99 | 38.20 | 257.42 | 34.39 | 37.79 | 39.21 |
| Foreman | 254.66 | 31.60 | 37.23 | 37.86 | 256.49 | 32.06 | 38.96 | 40.05 | 258.48 | 32.39 | 38.08 | 39.03 | 253.62 | 34.27 | 39.59 | 40.85 |
| Paris | 257.05 | 29.55 | 34.08 | 34.70 | 258.19 | 30.56 | 36.19 | 36.65 | 254.91 | 30.34 | 34.44 | 34.95 | 256.43 | 32.24 | 36.67 | 36.93 |
| Sean | 254.91 | 37.94 | 41.42 | 42.06 | 258.52 | 39.53 | 43.03 | 43.65 | 258.09 | 37.89 | 41.59 | 42.45 | 257.54 | 40.72 | 43.26 | 44.17 |





# **Comparison of Standards -** Streaming

- Published results show significant benefits of H.264/AVC (T. Wiegand et al, IEEE CSVT, July 2003). However, using parameters of your desired application, perform careful, independent evaluation as *your mileage may vary*.

| | MPEG-2 | | | | H.263 HLP | | | | MPEG-4 ASP | | | | H.264/AVC MP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sequence | Rate | PSNR-Y | PSNR-U | PSNR-V | Rate | PSNR-Y | PSNR-U | PSNR-V | Rate | PSNR-Y | PSNR-U | PSNR-V | Rate | PSNR-Y | PSNR-U | PSNR-V |
| **A: QCIF, 10 Hz, 32 kbit/s** | | | | | | | | | | | | | | | | |
| Foreman | 32.12 | 27.81 | 35.14 | 34.96 | 32.18 | 29.90 | 37.73 | 37.70 | 31.92 | 30.09 | 37.33 | 37.33 | 31.49 | 32.40 | 38.68 | 38.98 |
| Container | 32.22 | 32.71 | 39.75 | 39.04 | 31.97 | 35.96 | 41.38 | 41.12 | 31.82 | 36.42 | 42.46 | 42.23 | 31.89 | 38.57 | 43.00 | 42.98 |
| News | 32.44 | 29.97 | 35.07 | 36.75 | 32.50 | 34.06 | 38.68 | 39.31 | 32.24 | 33.30 | 37.50 | 38.58 | 31.96 | 35.75 | 39.45 | 40.05 |
| Tempete | 36.91 | 24.83 | 29.38 | 32.04 | 32.24 | 26.62 | 32.50 | 34.74 | 31.68 | 27.87 | 31.61 | 34.21 | 31.83 | 29.62 | 33.58 | 36.02 |
| **B: QCIF, 15 Hz, 64 kbit/s** | | | | | | | | | | | | | | | | |
| Foreman | 63.45 | 30.36 | 37.07 | 37.28 | 65.14 | 32.38 | 38.65 | 38.93 | 64.38 | 32.81 | 38.73 | 39.15 | 63.42 | 35.21 | 40.00 | 40.67 |
| Container | 63.95 | 34.34 | 40.95 | 40.40 | 63.97 | 38.26 | 43.32 | 43.22 | 63.87 | 38.47 | 44.21 | 43.87 | 63.67 | 40.67 | 44.80 | 44.92 |
| News | 63.45 | 32.61 | 37.33 | 38.55 | 63.80 | 36.25 | 39.79 | 40.43 | 64.00 | 35.78 | 39.37 | 40.67 | 63.98 | 38.80 | 41.71 | 42.27 |
| Tempete | 65.21 | 26.36 | 30.65 | 33.14 | 64.39 | 28.39 | 33.34 | 35.57 | 64.13 | 29.39 | 32.59 | 35.14 | 63.43 | 31.78 | 34.65 | 36.89 |
| **C: CIF, 15Hz, 128 kbit/s** | | | | | | | | | | | | | | | | |
| Foreman | 130.37 | 28.94 | 35.78 | 36.30 | 128.40 | 30.91 | 38.35 | 39.26 | 127.83 | 31.30 | 38.16 | 38.99 | 128.70 | 33.66 | 39.49 | 40.87 |
| Container | 127.90 | 32.63 | 39.94 | 39.95 | 129.02 | 34.99 | 42.00 | 41.84 | 128.62 | 35.28 | 42.16 | 41.91 | 128.67 | 36.74 | 42.40 | 42.40 |
| News | 129.84 | 32.73 | 37.92 | 38.98 | 129.02 | 36.68 | 40.82 | 41.47 | 126.97 | 35.71 | 39.20 | 40.58 | 128.25 | 38.21 | 41.21 | 42.09 |
| Tempete | 165.75 | 25.60 | 30.67 | 33.33 | 129.07 | 26.47 | 33.42 | 35.66 | 129.11 | 27.51 | 32.03 | 34.78 | 126.34 | 29.16 | 34.41 | 36.71 |
| **D: CIF, 15 Hz, 256 kbit/s** | | | | | | | | | | | | | | | | |
| Bus | 260.78 | 25.96 | 35.78 | 36.25 | 258.76 | 26.97 | 37.60 | 38.87 | 256.15 | 28.31 | 37.57 | 39.15 | 256.14 | 29.86 | 38.44 | 39.96 |
| Mobile | 256.01 | 24.59 | 29.96 | 30.17 | 259.20 | 25.66 | 31.97 | 32.40 | 258.88 | 27.07 | 32.24 | 32.63 | 254.87 | 29.73 | 34.26 | 34.69 |
| Flower | 261.67 | 23.93 | 28.82 | 32.37 | 257.85 | 24.89 | 31.58 | 33.56 | 255.97 | 26.07 | 30.89 | 33.90 | 257.89 | 28.08 | 33.02 | 35.08 |
| Tempete | 257.65 | 27.68 | 32.45 | 34.82 | 259.28 | 29.06 | 34.54 | 36.75 | 256.58 | 29.86 | 34.09 | 36.60 | 254.37 | 31.74 | 35.83 | 37.98 |
| **E: CIF, 30 Hz, 512 kbit/s** | | | | | | | | | | | | | | | | |
| Bus | 506.29 | 27.35 | 36.43 | 37.62 | 511.98 | 28.77 | 38.16 | 39.41 | 511.88 | 29.75 | 38.28 | 39.89 | 511.85 | 31.89 | 39.29 | 40.85 |
| Mobile | 506.26 | 25.31 | 30.26 | 30.47 | 513.05 | 26.74 | 32.40 | 32.85 | 505.03 | 28.36 | 33.12 | 33.54 | 512.58 | 31.27 | 35.18 | 35.65 |
| Flower | 518.64 | 25.71 | 30.25 | 33.08 | 517.90 | 26.35 | 31.99 | 34.14 | 511.76 | 27.96 | 32.16 | 34.79 | 514.59 | 30.16 | 33.95 | 35.67 |
| Tempete | 521.40 | 28.43 | 32.91 | 35.14 | 513.73 | 29.45 | 34.94 | 37.11 | 510.55 | 30.84 | 34.74 | 37.18 | 515.49 | 32.79 | 36.36 | 38.38 |
| **F: CIF, 30 Hz, 1024 kbit/s** | | | | | | | | | | | | | | | | |
| Bus | 1022.54 | 30.72 | 38.70 | 40.12 | 1025.80 | 31.91 | 39.55 | 41.21 | 1022.54 | 32.82 | 39.94 | 41.60 | 1025.51 | 35.24 | 40.77 | 42.59 |
| Mobile | 1029.58 | 28.16 | 33.00 | 33.27 | 1024.27 | 29.82 | 34.43 | 34.83 | 1029.18 | 31.37 | 35.29 | 35.74 | 1026.00 | 34.64 | 37.27 | 37.74 |
| Flower | 1034.33 | 28.66 | 32.92 | 35.10 | 1033.05 | 29.77 | 33.77 | 35.27 | 1024.30 | 31.20 | 34.58 | 36.61 | 1020.08 | 33.67 | 36.23 | 37.32 |
| Tempete | 1029.56 | 31.30 | 35.17 | 37.13 | 1022.81 | 32.55 | 36.53 | 38.50 | 1025.77 | 33.34 | 36.51 | 38.69 | 1020.06 | 35.54 | 37.90 | 39.68 |

# H.264/AVC Video Demo

- Demo of H.264/AVC Video Quality
  - Coding conditions:
    - » 1 Intra/segment
    - » 2 B-pictures between anchor (I or P) frames
    - » 2 References for prediction
  - MPEG standard test sequences
  - Coding Bitrates in 100 kbps - 800 kbps for CIF material
- Comparison with MPEG-4 at same bitrates
  - MS MPEG-4 codec used in comparisons with H.264/AVC
  - Bitrates for CIF are 250 kbps, 500 kbps and 750 kbps





# H.264/AVC Resources

- Reference encoder+decoder implementation
  http://bs.hhi.de/~suehring/tml/download/jm72.zip
- Other H.264 software
  http://www.vsofts.com/codec/h264.html
  http://sourceforge.net/projects/hdot264
- H.264/AVC over RTP
  http://community.roxen.com/developers/idocs/drafts/
  draft-ietf-avt-rtp-h264-01.html
- Ian Richardson's H.264 notes http://www.vcodex.com
- T. Wiegand's page http://bs.hhi.de/~wiegand/jvt.html

# Decoder Licensing Terms Comparison

## MPEG-LA

| MPEG-2 | MPEG-4 Part2 | AVC | MPEG-4 AAC Audio | High Efficiency AAC Audio |
|---|---|---|---|---|
| $2.50 per decoder w/o limit | $1.25 per decoder w/o limit $0.25 w/limit | TBD | $1.00 per stereo pair per decoder | $1.50 per stereo pair per decoder |

**Terms subject to change**





# H.264/AVC Summary

- MPEG-4 part10/AVC/H.264 standard approved
- Suitable for wide range of applications, resolution and bitrates
- Significant improvement in coding efficiency over earlier standards
- Encoder/Decoder lot more complex than earlier standards
- Uses familiar motion compensated predictive coding structure, but motion compensation with multiple reference frames and nany block sizes, smaller block transforms, and in-loop deblocking filter
- Error Resilience and Stream Switching considerations
- File Format, and Network Adaptation Layer (NAL)
- Publicly available software, current version called JM7.2
- 3 profiles, Baseline, Main, and Extended