
UNIVERSITÀ DEGLI STUDI
DI MODENA E REGGIO EMILIA

# Lecture notes for Multimedia Data Processing

## Other Video Standards

Last updated on: 28/04/2022

# The MPEG standard

- MPEG stands for Moving Picture Expert Group, which has worked to create the specification within the ISO, the International Organization for Standardization and the IEC, the International Electrotechnical Commission.
- MPEG-1 and MPEG-2 standards are based on very similar concepts: both use discrete cosine transform on blocks, and motion compensation. Video construction features from software-superimposed parts, to allow bit-rates lower than 64Kb/sec, where added in the MPEG-4. MPEG-1 and especially MPEG-2 have been widely used for applications as DVD, broadcast via satellite and terrestrial digital.
- Note that there is no MPEG-3. It should have been a standard for high-definition television but, with small extensions to the MPEG-2, it was possible to meet the higher bit-rate requirements, so the idea of an additional standard was abandoned.

# MPEG-1 and MPEG-2

- The MPEG-1 was made permanent in 1991 and was originally optimized to work on 352x240 pixel resolution at 30 fps (NTSC) or 352x288 pixels at 25 fps (PAL).
- In the standard, this is called Source Input Format (SIF) video.
- Often, incorrectly, it is thought that resolution in the MPEG-1 is limited to the size NTSC and PAL, but it actually supports up to 4095x4095 and 60 fps.
- The bit-rate is optimized for applications with 1.5 MB/sec, but again this is not forced into the standard. MPEG-1 is defined only for progressive frames and does not have support for applications that use interlaced video, such as television applications.
- For this reason, the MPEG-2 was introduced in 1994. It fulfilled exactly this type of needs, and introduced the first concepts of scalability. The reference bit-rate was also brought into the range between 4 and 9 MB/sec, thus potentially allowing very high quality video.
- MPEG-2 defines profiles (scale and color) and levels (resolution and bit-rate). The most widely used pair is known as Main Profile, Main Level.

# Differences from H.261 (1)

- The first and fundamental difference is that the concept of key-frame is introduced, which is a frame from which you can “restart" as if you were at the beginning of the video.
- This type of frame is called intra, hence the name “I-frames". In I-frames, all macroblocks must be intra, with no reference to the previous blocks.
- The other frames are called P-frames, as in predicted, because they can also contain macroblocks with references to other frames. Specifically to previously transmitted frames.
- It is important to note that there can also be intra macroblocks in P-frames, as is the case with H.261.

# Differences from H.261 (2)

- The second difference is in the quantization of the coefficients of the discrete cosine transform. The quantization in this case is not uniform, but is weighed for two arrays that by default are

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8 | 16 | 19 | 22 | 26 | 27 | 29 | 34 |
| 16 | 16 | 22 | 24 | 27 | 29 | 34 | 37 |
| 19 | 22 | 26 | 27 | 29 | 34 | 34 | 38 |
| 22 | 22 | 26 | 27 | 29 | 34 | 37 | 40 |
| 22 | 26 | 27 | 29 | 32 | 35 | 40 | 48 |
| 26 | 27 | 29 | 32 | 35 | 40 | 48 | 58 |
| 26 | 27 | 29 | 34 | 38 | 46 | 56 | 69 |
| 27 | 29 | 35 | 38 | 46 | 56 | 69 | 83 |

Quantization matrix for *intra* blocks

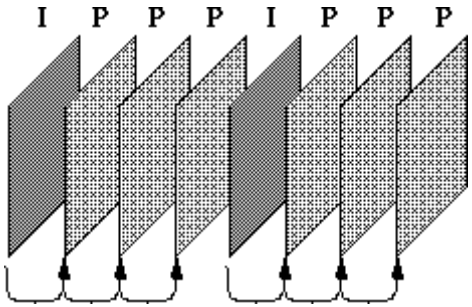
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |

Quantization matrix for *inter* blocks

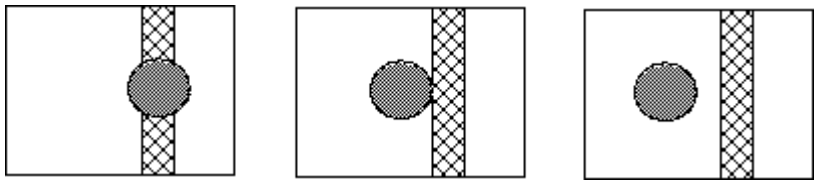
- There is also a scale factor in the range (0.0,1.0] that is multiplied by each of the values, allowing quantization to be adjusted during encoding.

# Novelty in the MPEG standards

- From what we said, the frame structure would such a dependency as shown in the figure (the arrow indicates data that comes from one frame and serves in another) :



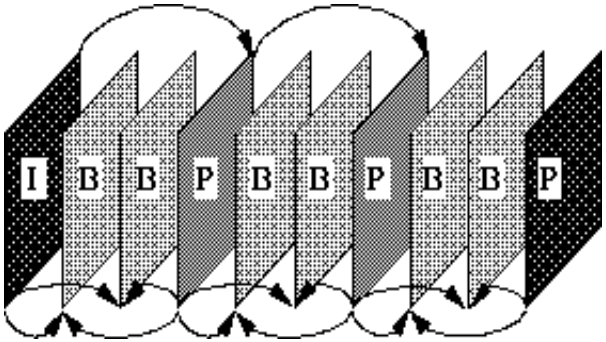

- Unfortunately, in many cases the information that is used to make the prediction is not available in the previous frame:



- While considering the frame in the middle, you can't find information about what was under the circle in the previous frame. But if we could look at the next frame, we would have that information!

# Frames with bidirectional prediction

- The solution adopted by the MPEG is to define a third type of frame, called bidirectional frame, o *B-frame*
- B-frames look for macroblocks in previous and following frames.




- The typical sequence found in an MPEG video is something like: IBBPBBPBB IBBPBBPBB IBBPBB
- This sequence can still be changed as needed, and each encoder can choose the best solution.

# Video Layer

- An MPEG video is divided into a hierarchy of layers that allow you to handle any transmission errors, stream search, editing, and audio synchronization.
- The first layer is called *video sequence layer*, and is a complete video with no external references (such as a movie or advertisement).
- The second layer is the GOP, which is the Group of Pictures: one or more I frames and possibly P or B frames.
- The third layer is the *picture*, then divided in *slices*. Each slice is a sequence of macroblocks (typically a multiple of the row).
- Each slice is then composed of macroblocks and blocks, similarly to H.261.
- Each one of these layers has its own 32-bit start code, defined in the MPEG syntax, and consists of 23 bits set to 0, followed by a 1, then followed by 8 bits that specify which start code we're actually looking at.

# B-frames

- Introducing forward prediction significantly complicates data stream management.
- Let's consider a sequence of frames of this type: $I_0B_1B_2P_3B_4B_5P_6B_7B_8\ I_9B_{10}B_{11}P_{12}$...

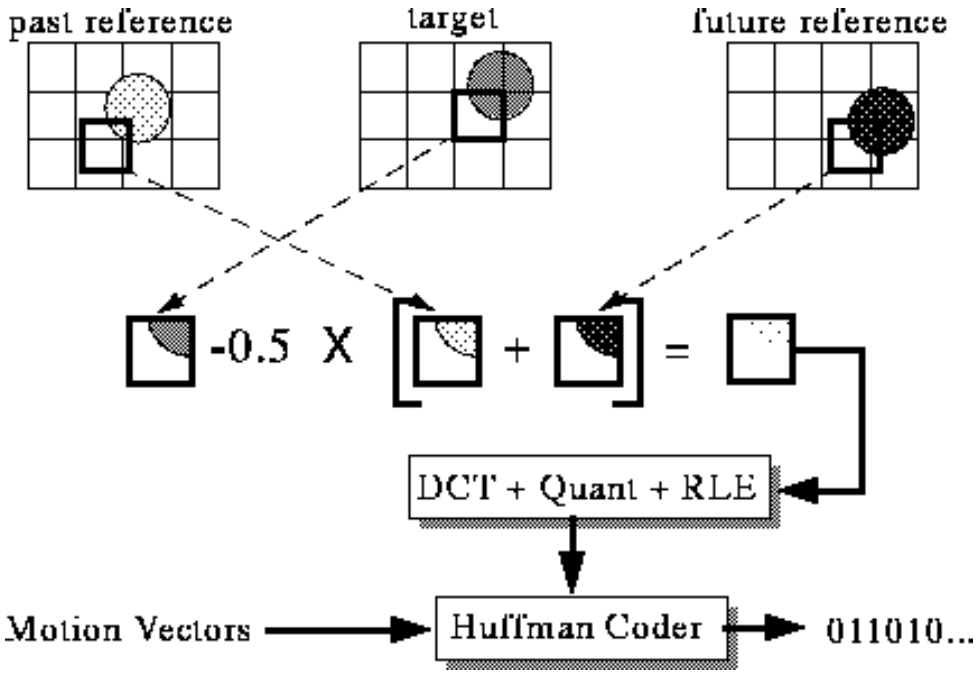

- Frame 0 is type I, so it can be decoded without any other information. Frame 3 is predicted, and will need information from frame 0. Frames 1 and 2 will instead refer to frames 0 and 3.
- It is thus clear that the decoding of frame 3 must be available before 1 and 2 can be decoded, so it is useless to transmit them first. For this reason, in the stream we will find this: $I_0P_3B_1B_2P_6B_4B_5\ I_9B_7B_8P_{12}B_{10}B_{11}$...


- Not all of the information needed to decode GOP1 is within it, as frames 7 and 8 depend on frame 6, which is in GOP0.
-

# Bidirectional prediction in B frames

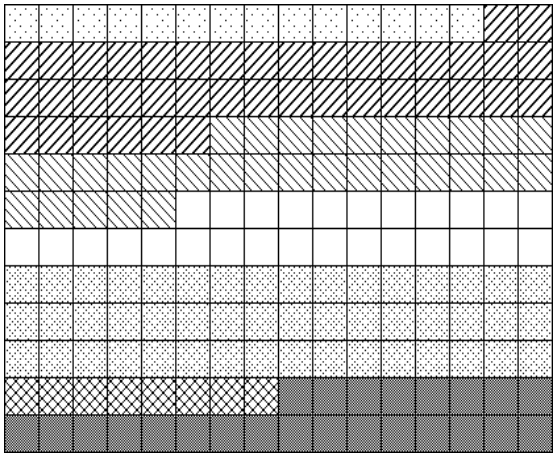
- For each macroblock in a B frame, there can be a backward reference, a forward reference, or both! What does it mean?

- This means that the current macroblock is predicted as an interpolation (average) between a previous macroblock and a subsequent macroblock:

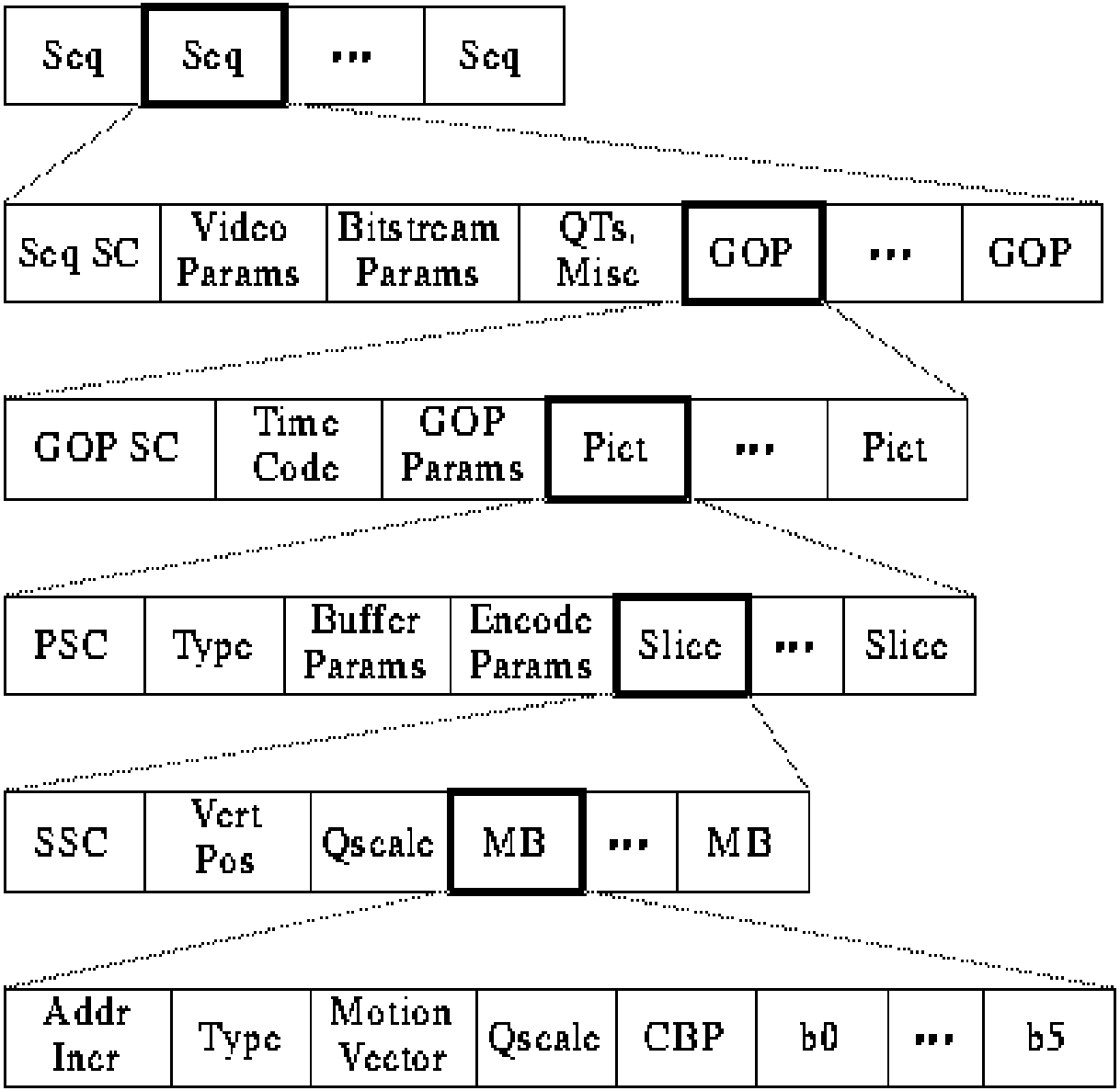


# More differences from H.261

- The MPEG has a greater time distance between frames I and P, so it is necessary to expand the motion vector search area (±32 pixels).
- Moreover, motion vectors are specified with a precision of half a pixel for more effective encoding. This means that two whole positions must be interpolated to obtain the correct reference.
- Thanks to the bitstream syntax, it is also possible to access the desired location (random access) and to fast forward effortlessly: decoding only P frames, or only I, or an I every *n*.
- Moreover, the notion of *slices* allows for faster synchronization after data loss. Example of a 7 slices image:



# Bitstream structure



# H.264/AVC

- Basic design architecture similar to MPEG-x or H.26x
- Better compression efficiency
  - Up to 50% in bit rate savings
- Subjective quality is better
- Advance functional element

- Initiate by the Video Coding Experts Group (VCEG) in early 1998
- Previous name H.26L
- Target to double the coding efficiency
- First draft was adopted in Oct. of 1999
- In Dec. of 2001, VCEF and the Moving Pictures Experts Group (MPEG) formed a Joint Video Team (JVT)
- Approved by the ITU-T as H.264 and ISO/IEC as International Standard 14496-10 (MPEG-4 part 10) Advanced Video Codec (AVC) in Mar. 2003

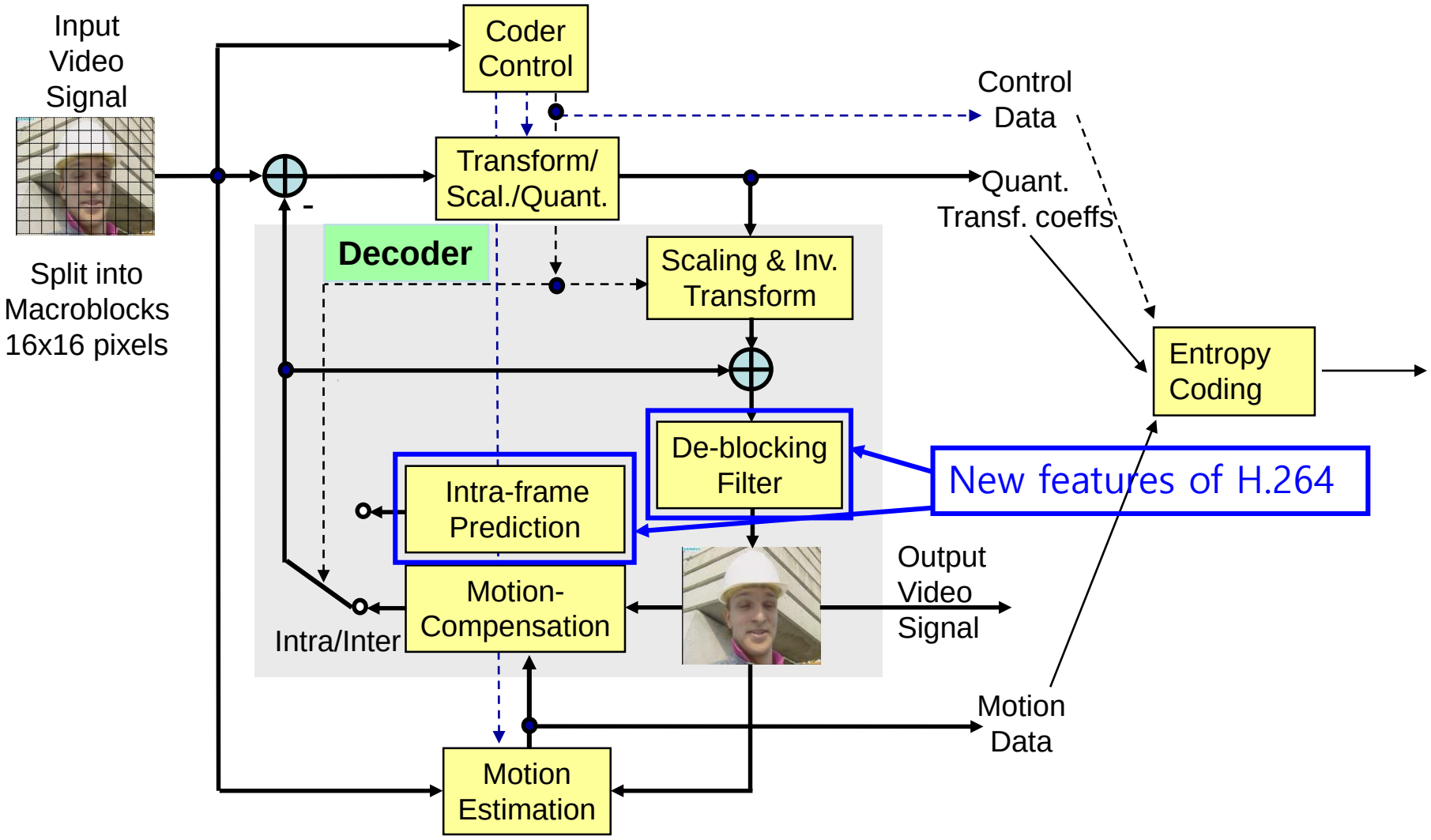
# Design Features Highlights

- Features for enhancement of prediction
  - Directional spatial prediction for intra coding
  - Variable block-size motion compensation with small block size
  - Quarter-sample-accurate motion compensation
  - Motion vectors over picture boundaries
  - Multiple reference picture motion compensation
  - Decoupling of referencing order form display order
  - Decoupling of picture representation methods from picture referencing capability
  - Weighted prediction
  - Improved “skipped” and “direct” motion inference
  - In-the-loop deblocking filtering

# Design Features Highlights

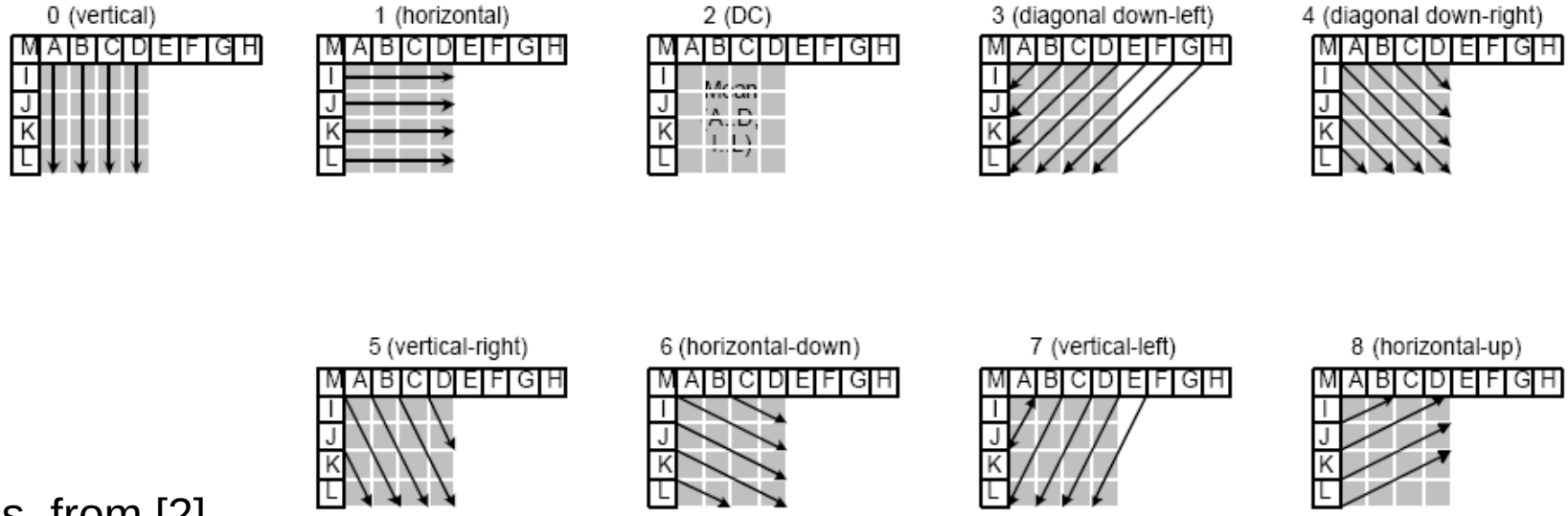
- Features for improved coding efficiency
  - Small block-size transform
  - Exact-match inverse transform
  - Short word-length transform
  - Hierarchical block transform
  - Arithmetic entropy coding
  - Context-adaptive entropy coding

# Architecture of the H.264 encoder



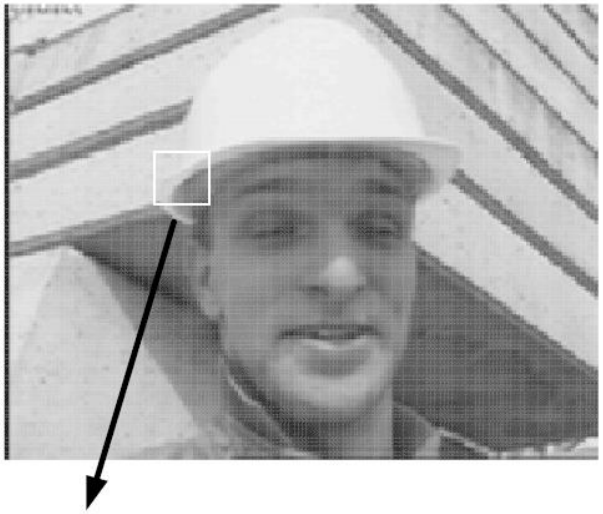
# Directional spatial prediction for intra coding

- Intra prediction is to predict the texture in current block using the pixel samples from neighboring blocks
- Intra prediction for 4×4 and 16 × 16 blocks are supported in H.264

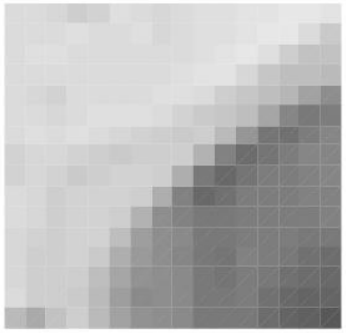



Figs. from [2]

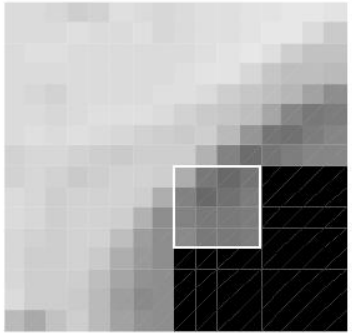
# Directional spatial prediction for intra coding (4×4)

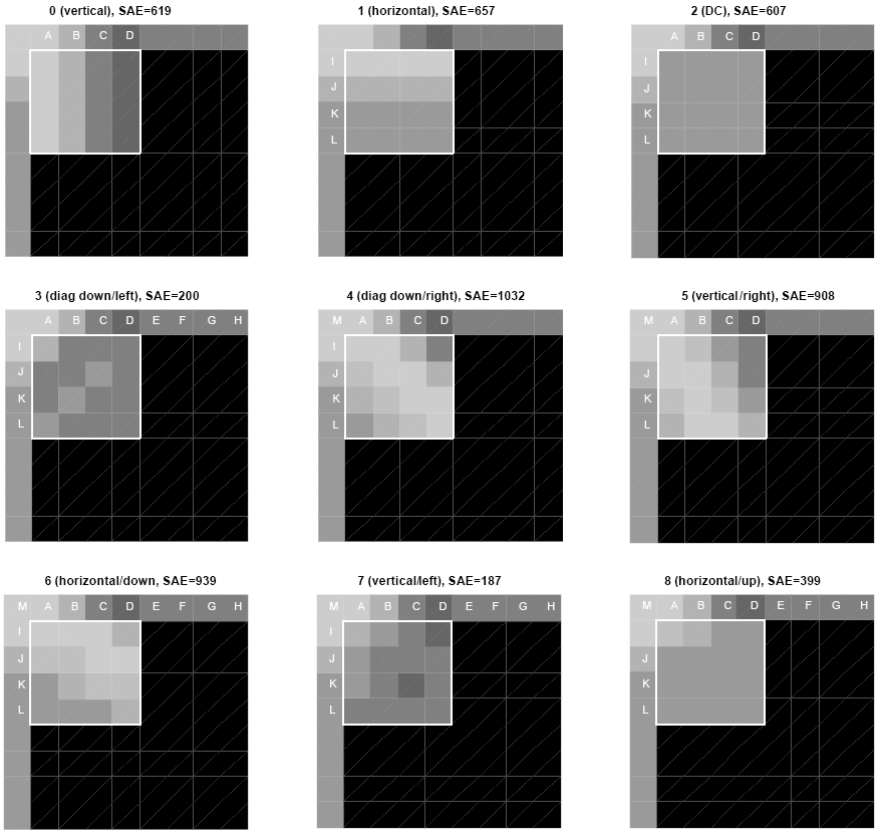


Original macroblock



4x4 luma block to be predicted

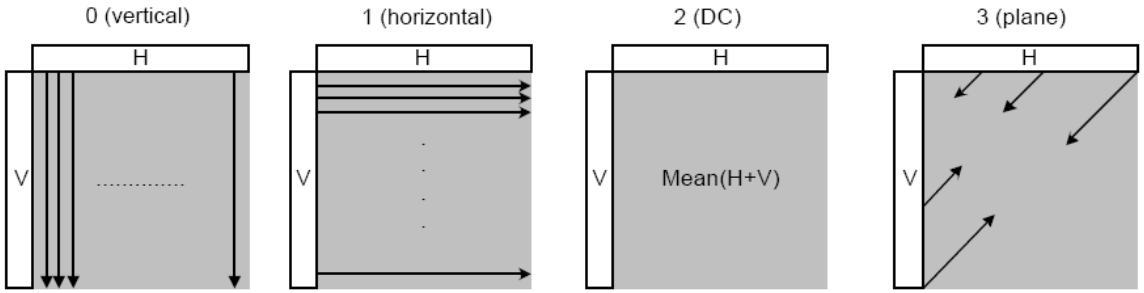


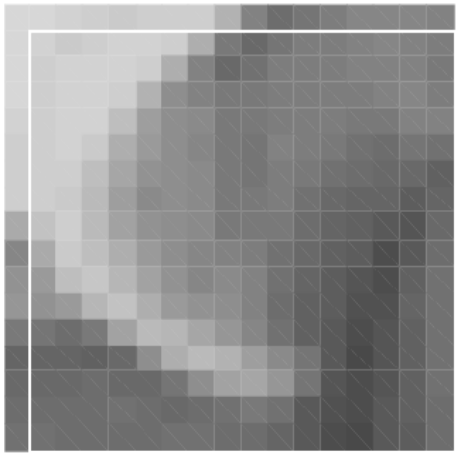


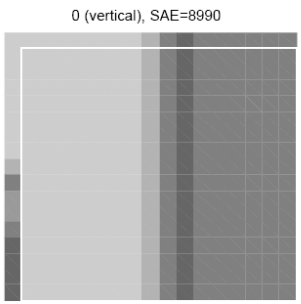

*Mode 7 is selected*

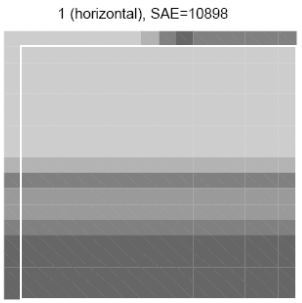
Figs. from [2]

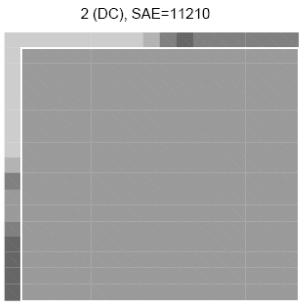
# Directional spatial prediction for intra coding (16×16)

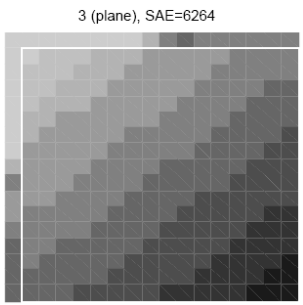




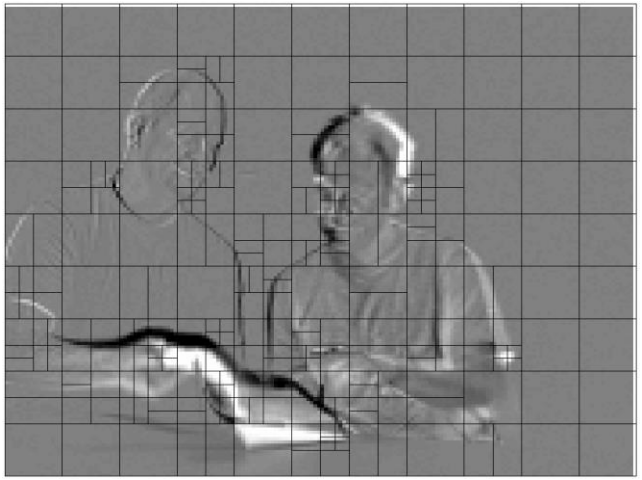




*Mode 3 is selected*

# Variable block-size motion compensation

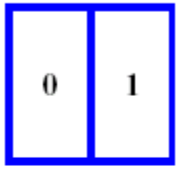
- Partitioned in 2 stages
- In the 1st stage, determine first 4 modes
  - 16×16, 16×8, 8×16, 8×8
- If mode 4 (8×8) is chosen, further partition into smaller blocks for every 8×8 block
  - 8×4, 4×8, 4×4
- At most 16 motion vectors may be transmitted for a 16×16 macroblock
- Large computational complexity to determine the modes



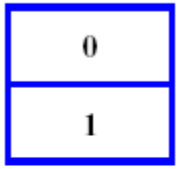
# Variable block-size motion compensation

**Mode 1**
One 16x16 block
One motion vector

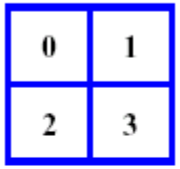

**Mode 2**
Two 8x16 blocks
Two motion vectors



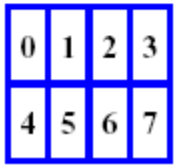

**Mode 3**
Two 16x8 blocks
Two motion vectors



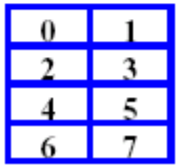

**Mode 4**
Four 8x8 blocks
Four motion vectors



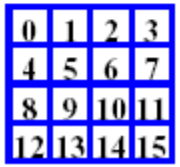

**Mode 5**
Eight 4x8 blocks
Eight motion vectors



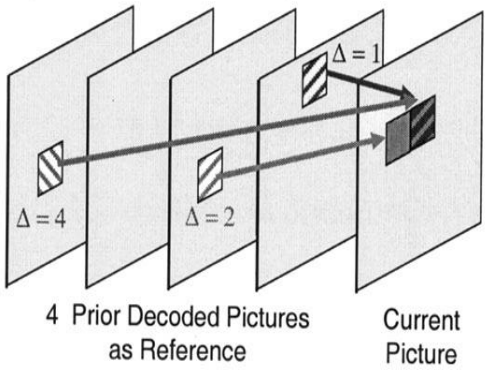

**Mode 6**
Eight 8x4 blocks
Eight motion vectors




**Mode 7**
Sixteen 4x4 blocks
Sixteen motion vectors



# Multiple reference picture motion compensation



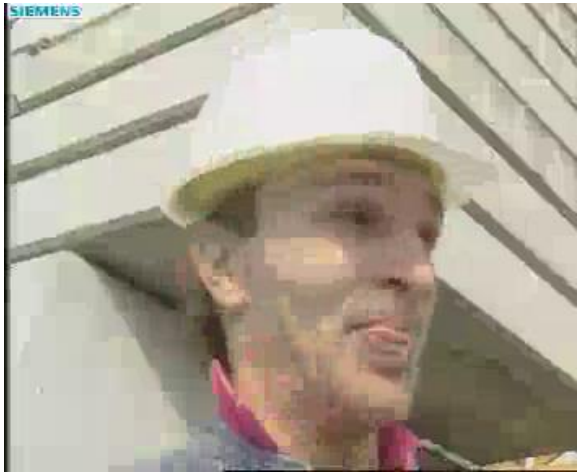

- More than one prior coded picture can be used as reference for MC prediction
- Reference index parameter is transmitted for each MC 16×16, 16×8, 8×16 or 8×8
- For smaller blocks within the 8×8 use 1 reference index
- P macroblock can also be coded in P-Skip type

# Multiple reference picture motion compensation

- Utilize two distinct lists of reference pictures
- Four different types of inter-picture predict
  - List 0, list 1, bi-predictive, and direct prediction
- Bi-predictive
  - weighted average of MC list 0 and list 1
- Direct prediction
  - Inferred from previously transmitted syntax
  - Either list 0 or list 1 prediction or bi-predictive
- Similar macroblock partitioning as P slices is utilized
- B_Skip mode is supported

# Adaptive Deblocking Filter

- Deblocking Filter
  - There are severe blocking artifacts
    - 4*4 transforms and block-based motion compensation
  - Result in bit rate savings of around 6~9%
  - Improve subjective quality and PSNR of the decoded picture




Without Filter


With AVC Deblocking Filter

# Small block-size transform

- Transformation is applied on 4×4 blocks
- Close to 4×4 DCT transform
- Inverse-transform mismatches are avoided
- The transform matrix is given as

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

# Short word-length transform

- Post-scaling matrix in forward transform
- Pre-scaling matrix in inverse transform
- Only integer operations and shifting are needed in transformation and quantization

$$\mathbf{Y} = \mathbf{C}_f \mathbf{X} \mathbf{C}_f^T \otimes \mathbf{E}_f = \left( \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} & & & \\ & \mathbf{X} & & \\ & & & \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}$$

$$\mathbf{X}' = \mathbf{C}_i^T (\mathbf{Y} \otimes \mathbf{E}_i) \mathbf{C}_i = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \left( \begin{bmatrix} & & & \\ & \mathbf{Y} & & \\ & & & \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}$$

$$a = \frac{1}{2} \qquad b = \sqrt{\frac{2}{5}} \qquad d = \frac{1}{2}$$

# Hierarchical block transform

- For macroblock coded in 16×16 Intra mode and chrominance blocks
- DC coefficients are further grouped and transformed
- Hadamard transform is used for chrominance block
- Intended for coding of smooth areas