MPEG-4 Natural Video Coding

Michael Wollborn, Iole Moccagatta, Ulrish Benzler
Presented by: Michael Closson

April 2, 2003

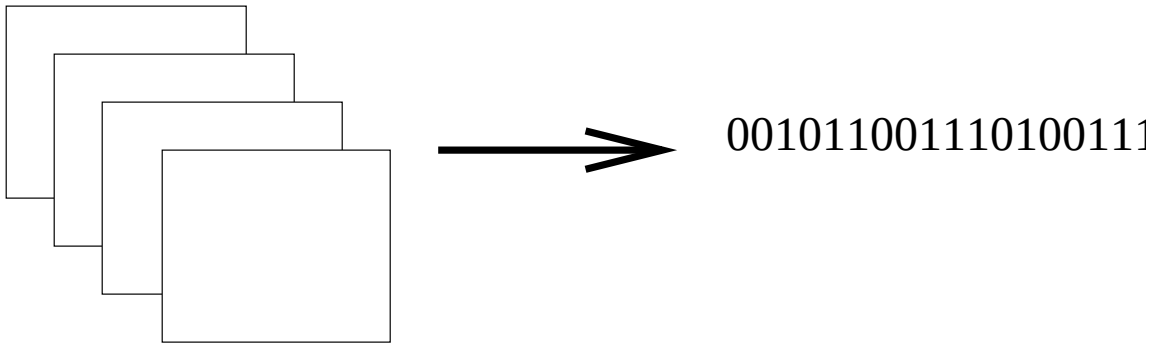
Many Figure were gratefully stolen from:

1. The MPEG-4 Book: Chapter 8 Natural Video Coding
   Michael Wollborn, Iole Moccagatta and Ulrich Benzler

2. Motion Estimation Algorithms for Video Compression
   Borko Furht, Joshua Greenberg, Raymond Westwater

Introduction

Video coding means to represent a sequence of images in a form suitable for storage and transmission.



- Traditionally, the primary goal was compression.
- Uncompressed video data uses a huge amount of disk space, even with modern storage technologies like DVD, compressions is required.
- Other goals of video coding include error resiliency, authentication, encryption, scalability in terms of resolution and bitrate.

Introduction

- Traditionally, video coding codes only rectangular frames.
- Some recent approaches are *object-based* or *region-based* coding.
- The goal of these recent approaches is still compression.
- But they broaden the potential functionalities and applications of coded video.
- MPEG4 is based on combining these two approaches.

General Overview

- The MPEG4 standard consists of a set of tools for multimedia.
- MPEG4 tools can code both sequences of images (video) and still images (visual texture).
- Video coding in MPEG4 is based on Motion-Compensated Hybrid DCT coding. (Similar to MPEG1/2.)
- Still image coding is based on wavelet transform and zero-tree encoding. (Similar to JPEG 2000.)

MPEG4 supports all of the functionalities of MPEG1/2 as well as the following.

- Coding of arbitrarily shaped objects.
- Efficient compression of video sequences and still images over a wide range of bit rates.
- Spatial, temporal and quality scalability.
- Robust transmission in error-prone environments.

The MPEG-4 standard doesn't specify the encoder

- As in previous MPEG standards, the encoder is not standardized. Only the bitstream and the decoder.

- This innovative approach encourages researchers to competitively develop novel approached to encoding.

- The algorithms and data structures to produce a bitstream are not specified in the MPEG4 standard.

- Reference implementations are provided.

Profiles and Levels

- The scope of the MPEG4 standard is large and contains many tools.

- It is not always desirable for a decoder to support all of the tools.

- If you just want an MPEG4 video player for your PC, then some tools like coding of arbitrary shapes may not be useful to you and a smaller decoder optimized for only playing rectangular-shaped video.

Profiles and Levels

To facilitate this notion, MPEG4 defines *profiles* and *levels*.

- Each profile has a set of tools associated with it.
- A decoder can claim adherence to a certain profile by implementing only the tools specific by the profile.

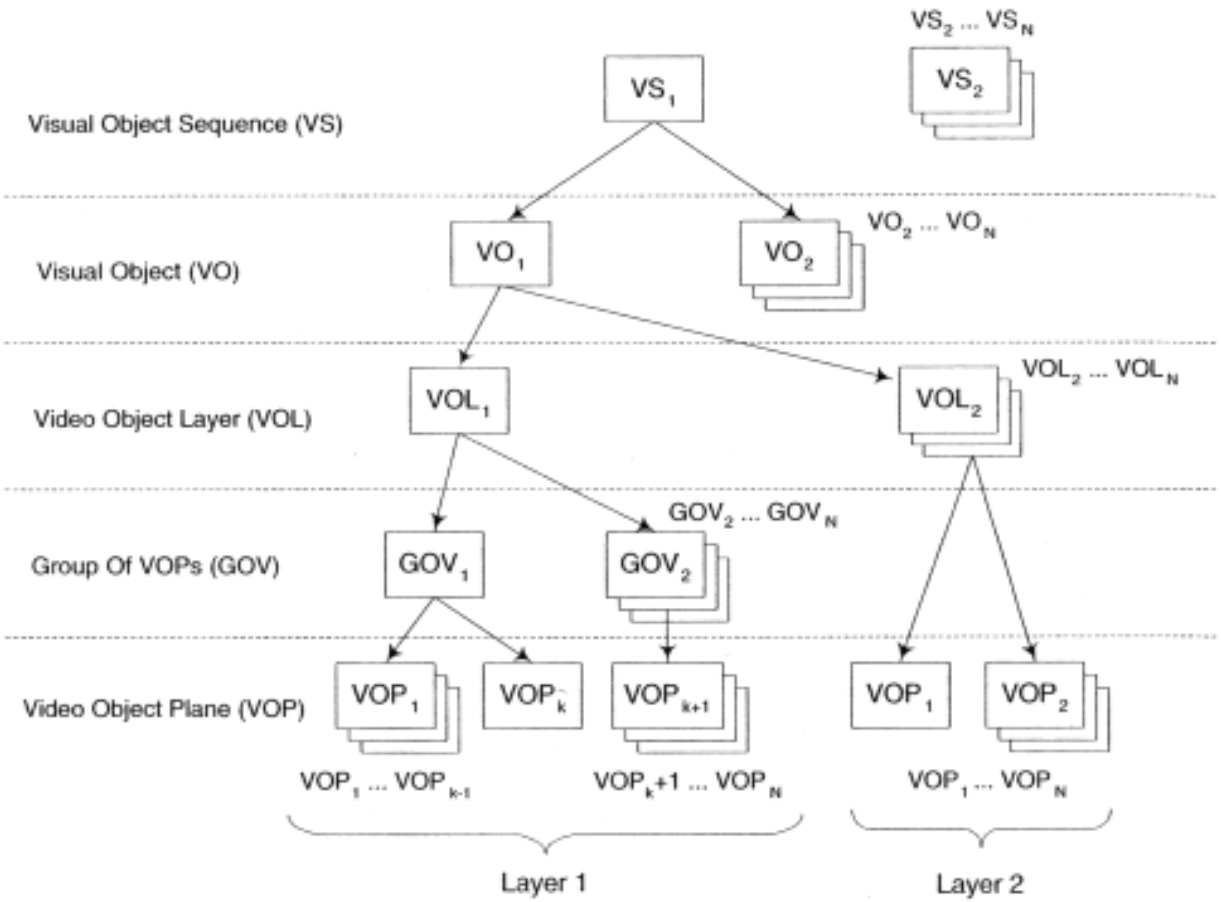
Profiles and Levels

Profiles are further divided into levels.

- Levels are based on complexity bounds.
- For example: memory required, computational complexity, bit rates handled.

In the literature, or if you use an MPEG encoder, you may see things like `MP@ML`. This means Main Profile and Main Level.

This Figure shows the MPEG-4 Object Hierarchy



MPEG-4 Object Hierarchy

MPEG4 video bitstreams can contain many structures. These structures have a hierarchical representation.

Video Object Sequence (VS) This is just an ordered collection of Video Objects.

Video Object (VO) A VO could be for example a video.

Video Object Layer (VOL) If your MPEG4 video contains multiple resolutions or bitrates then it may have one VOL for each.

MPEG-4 Object Hierarchy

Group of VOPs (GOV) In the context of video coding, a GOV is analogous to a GOP in MPEG1/2, that is its a group of pictures. Generally 8 to 10 pictures for a group. The purpose of GOV's is to facilitate random access in the video, among other things.

Video Object Plane (VOP) Again, in the context of video coding, a VOP is analogous to a Frame in MPEG1/2. One difference between a VOP and a frame, is a VOP can contain both data (YUV data) and shape (alpha plane). Where as a frame contains just data.

Coding of Rectangular Video Objects

Based on same techniques as MPEG1/2, with significant enhancements.

First, an introduction to some key MPEG concepts.

- Motion Compensation
- Texture Coding
- Intra, Predictive and Bidirectional (I,P,B) Frames

Motion Compensation (MC)

- What is Motion Compensation?
- Block Matching
- Prediction error
- Full-pel and Half-pel Motion Vector Resolution

What is Motion Compensation?

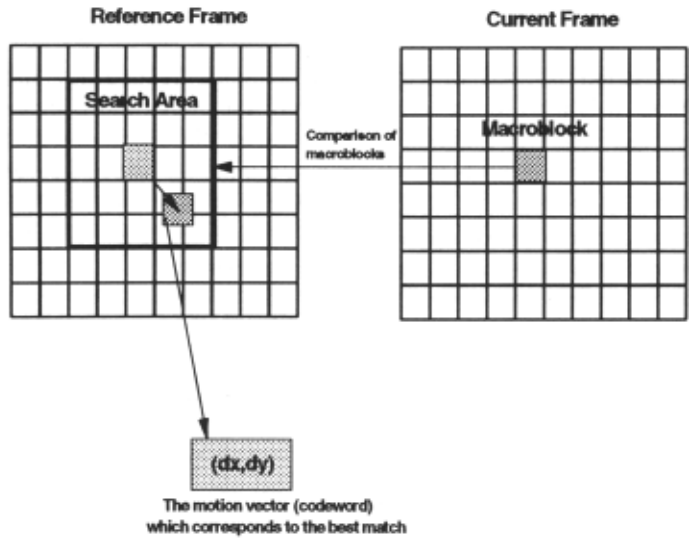
In two consecutive video frames, there usually isn't much difference between them.

What is Motion Compensation?

It doesn't make sense to code them separately since much redundant information will be stored. MC tries to reduce this redundancy, known as temporal redundancy.

Block Matching

MC in MPEG (1/2/4) accomplishes this reduction of temporal redundancy by matching a macroblock of one frame to some block of other frame(s) (known as the reference frame(s)). When it finds the closest matching block, it stores the location of the block (known as the Motion Vector (MV)) and any differences between the two blocks (known as the prediction error).



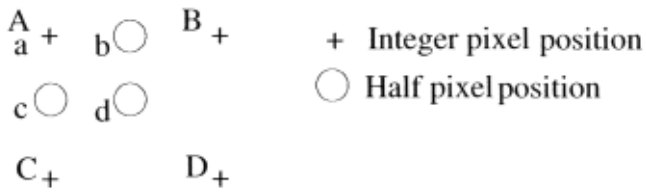
Intra-mode and Inter-mode coding

If a MB is motion compensated, it is coded in terms of other frames. This coding mode is known as *inter-mode*.

If MC is not applied, and therefore the MB can be decoded without referring to other frames, this coding mode is known as *intra-mode*.

Half-pel MC

The granularity of search in the reference frame can be either at the full-pel level or at the half-pel level. In half-pel mode, the value of a pel is computed using *bilinear interpolation*.



The formula for calculating the value of b is

$$b = \frac{A + B + 1}{2}$$

The formula for calculating d is

$$d = \frac{A + B + C + D + 2}{4}$$

Quarter-pel MC

This method can be extended to calculate quarter-pel MC, but the results do not improve the coding efficiency.

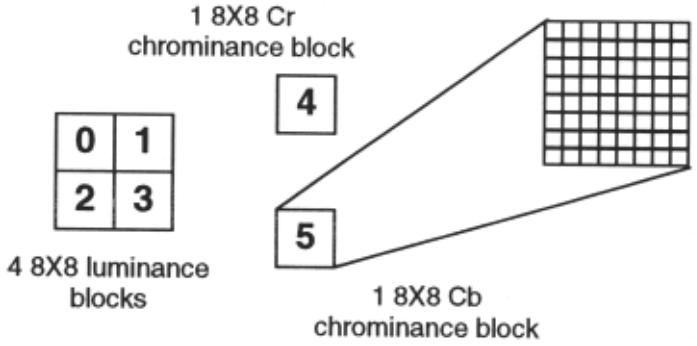
MPEG4 uses an improved method of calculating half-pel MV, which then makes the computation of quarter-pel MVs give facilitate better coding efficiency.

Texture Coding

- Discrete Cosine Transform (DCT)
- Quantization
- Zig-zag scan (Matrix to vector conversion)
- Variable Length Coding (Huffman Coding)

Texture Coding

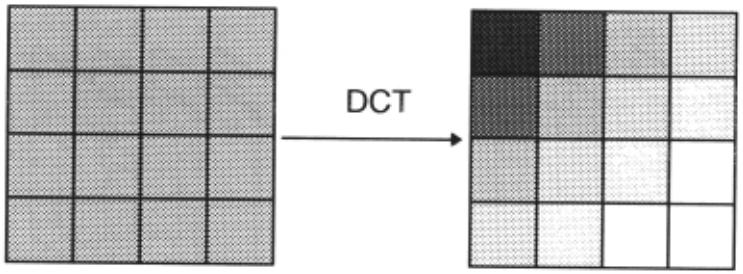
- Texture coding if basically JPEG compression.
- MBs are split in to 4 8x8 blocks.
- Texture in RGB format is converted to YCrCb format.
- The chromanence values (CrCb) are sub-sampled as in JPEG.



Discrete Cosine Transform (DCT)

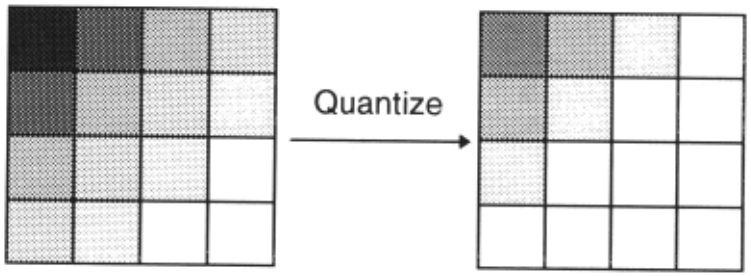
The DCT orders a 2D MB by frequencies.

> The objective of the DCT is to make as many transform coefficients small enough so that they are insignificant and can be discarded in the quantization step.
>
> — from Digital Pictures second edition: Netravali, Haskell



Quantization

The purpose of the quantization step is to change the DCT coefficients in such a way that they can be coded in a compact way, but still preserve the original image as much a possible. Quantization is the lossy step of texture coding. Here information is discarded that can never be restored.



Quantization (cont)

- There are different quantization matrices for intra and inter MBs.

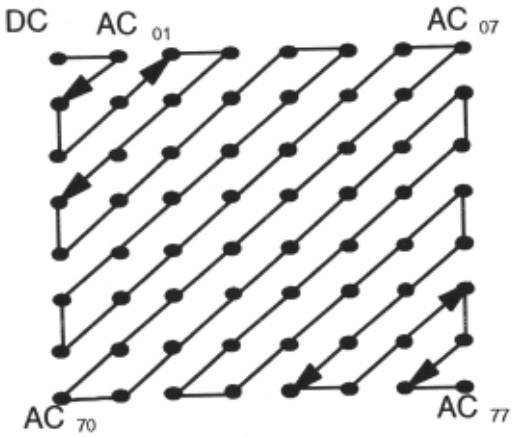
- Because intra MBs store picture data where inter MBs store the prediction error.

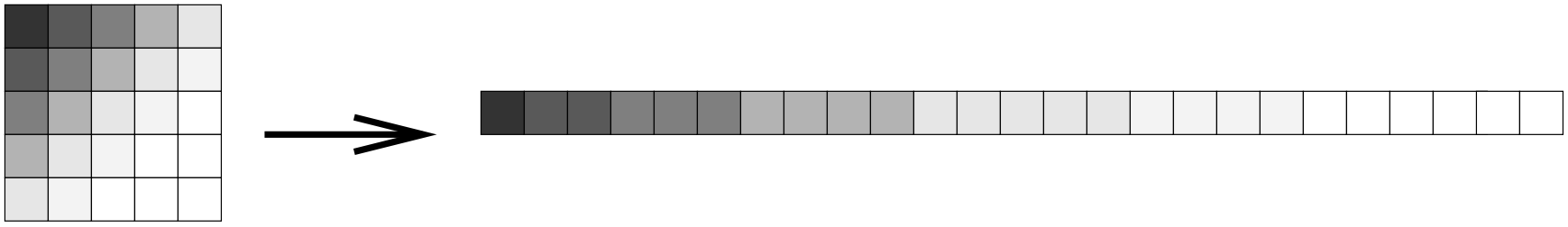
- MPEG defines some standard quantization matrices, but allows for the specification of custom matrices in the mpeg header.

Zig-zag scan (Matrix to vector conversion)

- This purpose of the zig-zag scan is to reorder the 2D matrix representation of a block, in a 1D vector representation that facilitates efficient VLC coding.

- This step transforms this 2d matrix to a 1d vector in such a way that long runs of zeros appear frequently

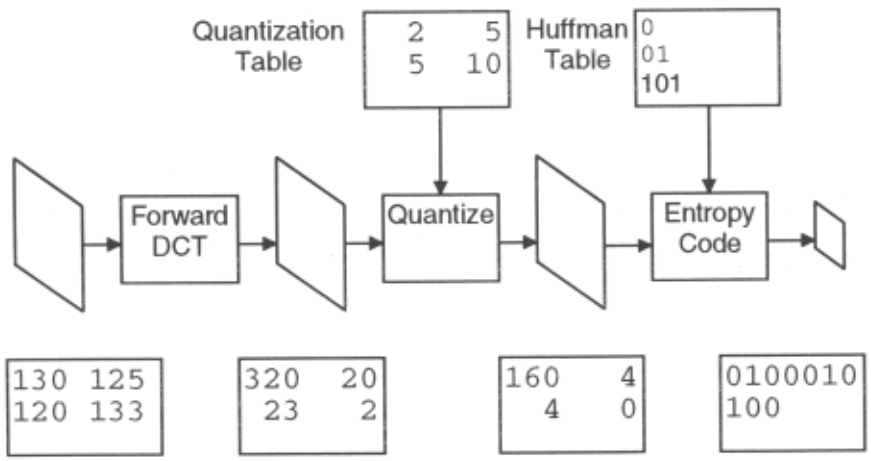




Variable Length Code (VLC); Huffman Code

- The 1D vector produced from the previous step is coded using a Huffman variable length code (Huffman VLC).

- This allows a very compact representation of the original 8x8 macroblock.

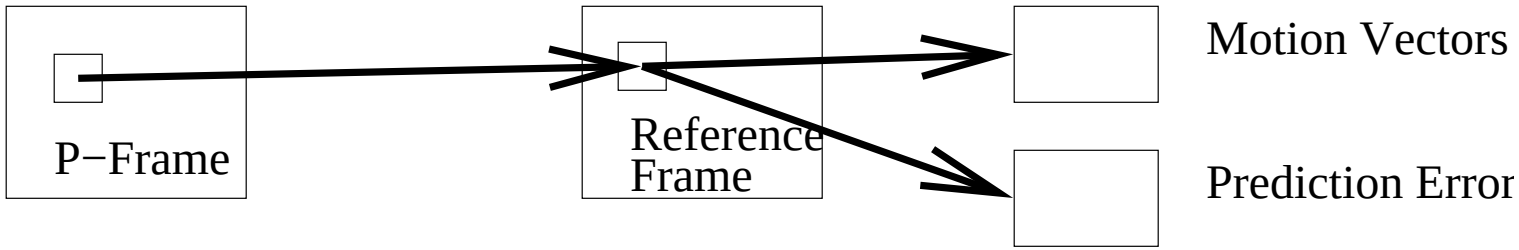


MPEG Frame Types

I-frame Intra-frame, motion prediction is not applied to these frames. Therefore, I-frames can be decoded without reference to any other frame. However, the compression ratio for I-frames is not as good as for the other frame types.

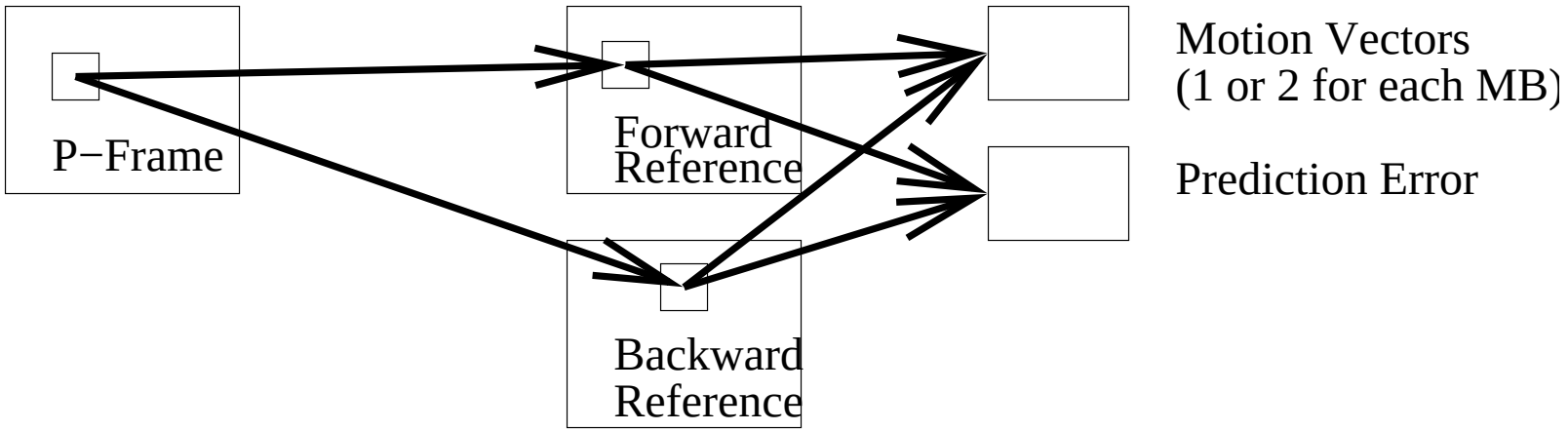
MPEG Frame Types

P-frame Predictive-frame, motion prediction is applied to this frame using a previous I or P-frame.



MPEG Frame Types

B-frame Bidirectional-frame, motion prediction is applied to the frame using either a previous I/P-frame, a future I/P-frame or both. In MPEG-1/2 three different prediction modes are defined. (MPEG4 adds a forth called Direct mode).



B-Frame Modes

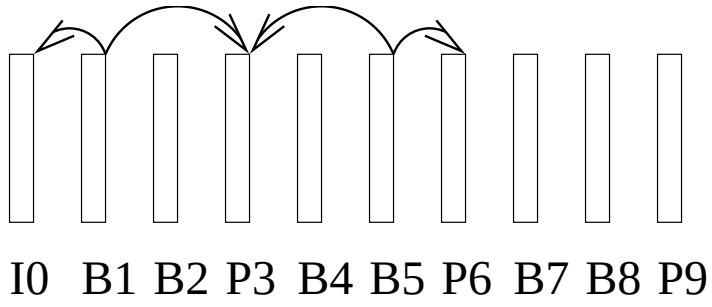
Forward mode only the forward reference MV is used in motion compensation. Only one MV is sent.

Backward mode only the backward reference MV is used in motion compensation. Only one MV is sent.

Interpolative mode Each MB in the frame has 2 MVs, one references a future frame and one a backward frame. The resulting MB is an interpolation of the two referenced MBs.

Frame Transmission Order

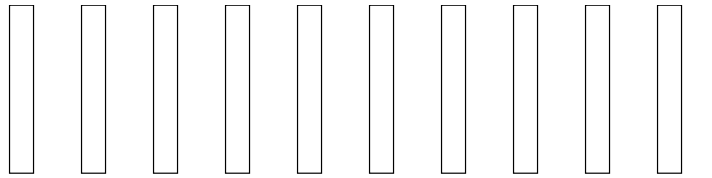
Since B-frames could potentially reference a frame that has not been transmitted yet, the transmission order of frames is different from the display order. For example, if frames in a GOP are arranged like this:



MPEG1/2 Encoding

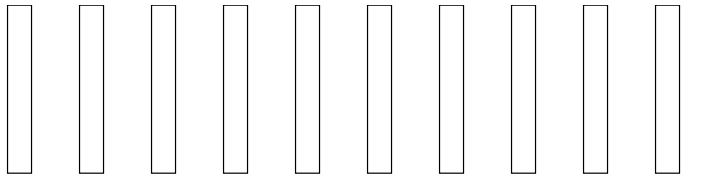
- A sequence of frames are split up into groups of typically 6 to 10 frames.
- These groups are known as GOPs (Group of Pictures).
- The frames in a GOP are assigned a frame type.

For example the frames in a GOP may be arranged like this



MPEG1/2 Encoding

- Compress frame 0 without MC.
- Compress frame 3 using frame 0 as a reference for MC.
- Compress frames 1 and 2 using frames 0 and 3 as potential references for Bidirectional MC.
- Compress frame 6 using frame 3 as a reference for MC.
- Compress frames 4 and 5 using frames 3 and 6 as potential references for bidirectional MC.



MPEG1/2 Encoding, I-frames

The details for compressing a I frame are as follows:

- The frame is split into 16x16 MBs, the MBs are further split into 8x8 blocks,
- Texture compression is applied to the blocks.
  - DCT
  - Quantization
  - Zigzag scan
  - Huffman VLC

MPEG1/2 Encoding, P-frames

The details of P-frame compression are as follows,

- The frame is split into 16x16 MBs.
- Motion compensation is applied to each MB using the appropriate frame as a reference. The resulting MV is stored.
- The difference between the MB and the predicted MB is taken.
- This difference is split into 8x8 blocks and texture compression is applied.

MPEG1/2 Encoding, B-frames

B-frame compression is similar to P-frame except that the encoder can use whatever mode of B-frame MC that results in the closest block match.

The entire video is compressed by processing it one GOP at a time.

1MV, 4MV mode

In MPEG-4, two motion modes are defined:

- 1MV mode - one MV per MB is transmitted.
- 4MV mode - the MB is further split into 8x8 blocks and one MV for each block is sent.

New Motion-Compensation Tools

MPEG-4 introduces 3 new tools to improve motion compensation.

- Quarter-Pel Motion Compensation
- Global Motion Compensation
- Direct mode in bidirectional prediction.

Quarter-Pel MC

- MPEG-4 increases the resolution of MVs through quarter-pel MC.
- This increases the resolution of MVs 16 times over full-pel MC.
- Rather then use bilinear interpolation, MPEG4 uses an 8-tap FIR filter (finite impulse response).
- As opposed to bilinear interpolation, the FIR filter takes into account aliasing due to discretization.

Using a FIR filter to improve Half-pel MC

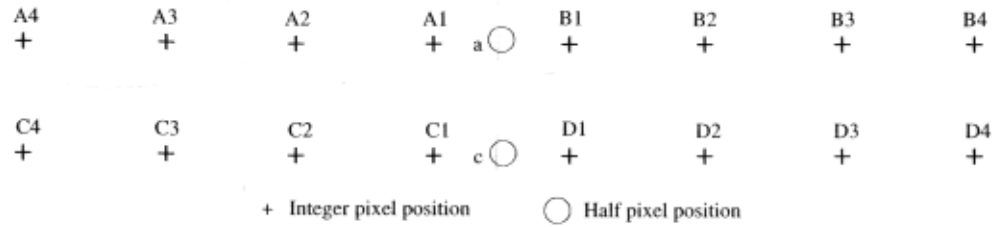
The filter values are defined in the MPEG-4 standard, these are those values

$$[\frac{-8}{256}, \frac{24}{256}, \frac{-48}{256}, \frac{160}{256}, \frac{160}{256}, \frac{-48}{256}, \frac{24}{256}, \frac{-8}{256}]$$

MPEG 4 uses the FIR filter to compute half-pel samples, it then uses bilinear interpolation to calculate the quarter-pel samples.

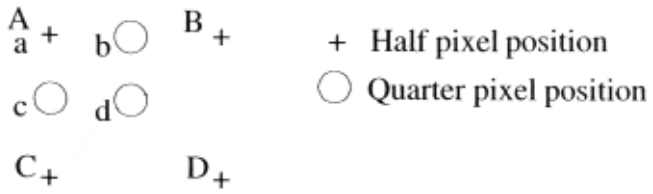
Quarter-Pel MC

Half-pel samples are computed as follows



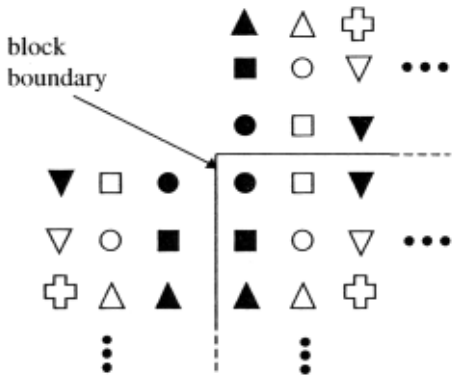
$$a = \frac{-8 \cdot A4 + 24 \cdot A3 - 48 \cdot A2 + 160 \cdot A1 + 160 \cdot B1 - 48 \cdot B2 + 24 \cdot B3 - 8 \cdot B4}{256}$$

In the case of the central pel, the filtering is applied in the horizontal direction first, and then in the vertical direction.



Quarter-Pel MC

What happens at the boundary of a MB or at the edge of a VOP? MPEG4 uses a technique known as boundary mirroring.



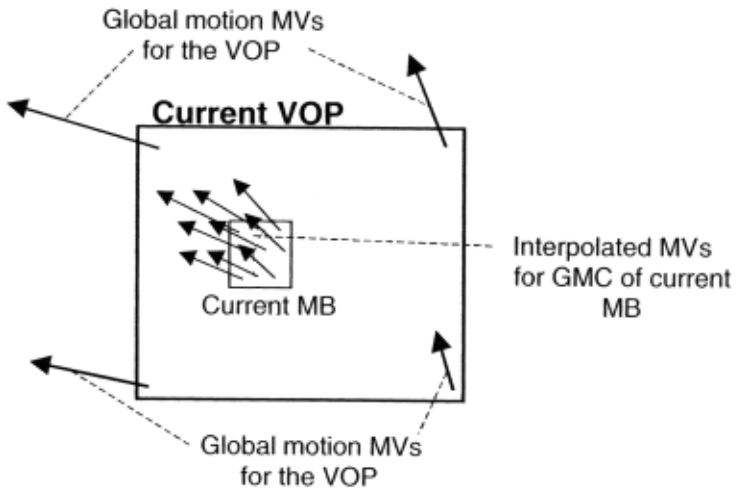
- Even though pels are available at non-VOP boundaries, these pels are not used to decrease the requires memory bandwidth of coding/decoding.
- With half-pel MC (8+1)x(8+1) = 81 pels are read from the reference block.
- Without boundry mirroring, (8+7)x(8+7) = 225 pels from the reference block must be read.
- Resulting in an increase of 2 3/4 times the memory.
- With boundry mirroring, only 81 pels are read from the reference VOP.

Global Motion Compensation

- Some types of motion, for example panning, zooming or rotation; could be described using one set of motion parameters for the entire VOP.
- For example for panning, each MB could potentially have the exact same MV.
- Global MC allows the encoder to pass one set of motion parameters in the VOP header to describe the motion of all MBs.
- Additionally MPEG4 allows each MB to specify its own MV to be used in place of the global MV.

Global Motion Compensation

Global MC parameters are a set of 4 MVs.

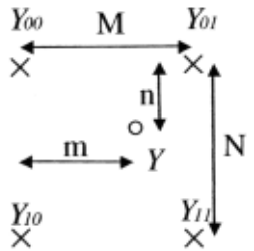


In addition to specifying the position on a reference frame, the tail of the MV, on the current frame can also be specified. In the figure above, the MVs are at the corners of the VOP.

Then the MVs for a MB can be derived from the GMC MVs through interpolation. The resolution of the MVs (full, half or quarter-pel) must be specified in the bitstream to ensure identical results at both the encoder and decoder. This process is also known as *warping*.

Global Motion Compensation

For sub-pel resolution, the samples are calculated using bilinear interpolation. For example,



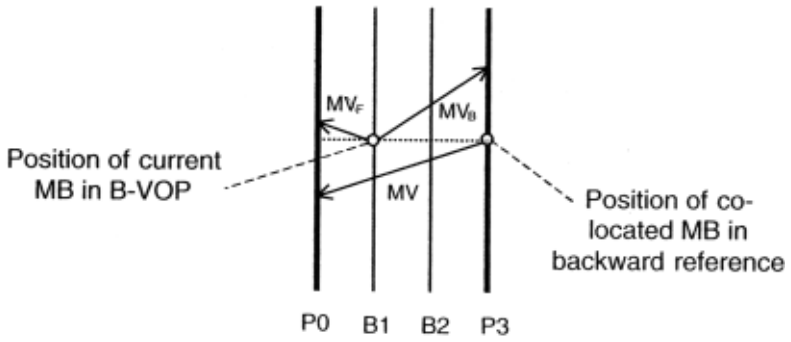
$$Y = \frac{N-n}{N} \cdot \frac{M-m}{M} \cdot Y00 + \frac{N-n}{N} \cdot \frac{m}{M} \cdot Y01 + \frac{n}{N} \cdot \frac{M-m}{M} \cdot Y10 + \frac{n}{N} \cdot \frac{m}{M} \cdot Y11$$

Direct mode in bidirectional prediction

Recall, interpolative mode bidirectional MC. In this mode, two MV are coded referencing two separate reference VOPs.

In direct mode, these two MVs are derived from the VOPs relative temporal position between the two reference frames and parameter called the delta vector which is encoded with the MB.

Direct mode in bidirectional prediction



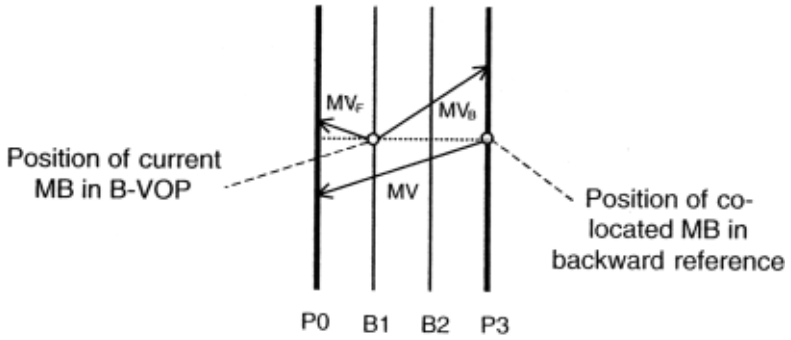
The figure above shows an example of how the forward and backward referencing MVs are computed in direct mode bidirectional MC.

First two parameters are computed, these parameters are the relative temporal position of the VOP from the two reference VOPs, they are

$$TRB = display\_time(currentVOP) - display\_time(forward\_ref)$$

$$TRD = display\_time(backward\_ref) - display\_time(forward\_ref)$$

Direct mode in bidirectional prediction



Then the MVs are calculated using the following formulas

$$MV_f = \frac{TRB}{TRD} MV + MV_d$$

$$MV_b = \frac{TRB - TRD}{TRD} MV + MV_d$$

MV is the MV of the co-located MB of the backward reference VOP.

A couple of caveats regarding direct mode

The MPEG4 standard states that only 1MV mode may be used with bidirectional MC. However, it is possible that the backward reference P-VOPs that a direct mode coded B-VOP references was coded in 4MV mode. If this is the case, then 4MVs are used for the B-VOP. Therefore, Direct mode is the only option for coding a B-VOP in 4MV mode.

A couple of caveats regarding direct mode

If a MB in a B-VOP is coded in *skipped mode* (ie. no MVs or prediction error sent). Then the MB is processed in direct mode with a zero delta vector and no prediction error will be applied.

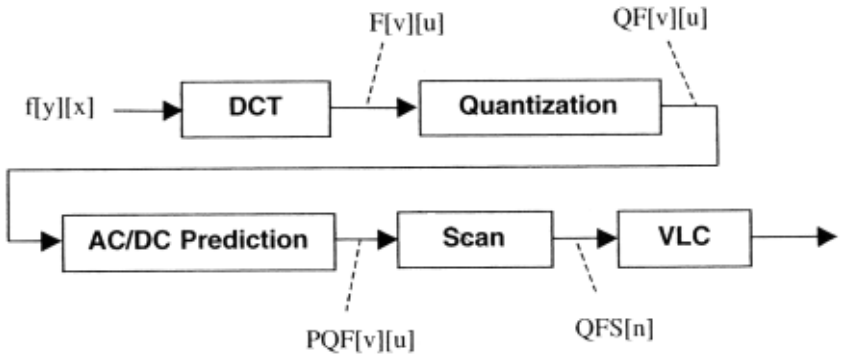
If the backward reference VOP MB was also coded in skipped mode, then the current B-VOP MB is coded in what is called *implicit skipped mode*. In this case the B-VOP MB is processed using forward mode MC with a zero MV.

New Texture Compression Tools

In MPEG-2, texture resolution for intra frames is 8 bits. For Inter frames it is 9 bits. MPEG-4 allows using anywhere from 4 to 12 bits for either intra or inter frame texture coding. Coding of grey-scale alpha values (for coding of arbitrarily shaped objects) is fixed at 8 bits.

MPEG-4 introduces 3 notable changes to the texture coding process described previously.

Two methods to quantize DCT coefficients



MPEG-4 allows for two methods of DCT coefficient quantization.

- This first is the MPEG2 method of quantization.
- This first is the H.263 method of quantization.

quantizer_scale parameter

The parameter quantizer_scale controls how much information is discarded during the quantization process.

- quantizer_scale parameter can take values from 1 to 31 in the case of 8-bit textures

- and 1 to $2^{quant\_precision} - 1$ in the case of no 8-bit textures.

A different quantizer_scale can be used for each VOP.

Intra DC Coefficient Quantization

The F[0][0] entry of a luminance or chromanance block is known as the DC coefficient. It has a special name because it represents the mean luminance or chrominance value, it is quantized using a fixed quantizer step.

$$Quantization : QF[0][0] = \frac{F[0][0]}{dc\_scaler}$$

$$De - Quantization : F[0][0] = QF[0][0] \cdot dc\_scaler$$

dc_scaler depends on the quantizer_scale parameter and is determined from the following table.

`quantizer_scale`(Q_p)	1-4	5-8	9-24	25-31
`dc_scaler` (luninance)	8	2Qp	Qp + 8	2Qp - 16
`dc_scaler` (chrominance)	8	$\frac{Q_p+13}{2}$	$\frac{Q_p+13}{2}$	$Q_p - 6$

First Quantizer method: MPEG Quantization

MPEG quantization introduces a weighting factor into the process. The purpose of this is to exploit properties of the human visual system. Since human eyes are less sensitive to some frequencies, these frequencies can be quantized with a coarser step-size then more important frequencies. This results in a more compactly coded bit-stream with minimal distortion to the picture.

MPEG Quantization uses different matrices for intra and inter blocks.

Example matrices are shown on the next slide, these are the default matrices defined in the standard.

First Quantizer method: MPEG Quantization

8	17	18	19	21	23	25	27
17	18	19	21	23	25	27	28
20	21	22	23	24	26	28	30
21	22	23	24	26	28	30	32
22	23	24	26	28	30	32	35
23	24	26	28	30	32	35	38
25	26	28	30	32	35	38	41
27	28	30	32	35	38	41	45

Default weighting matrix for *intra coded* MBs

16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

Default weighting matrix for *inter coded* MBs

First Quantizer method: MPEG Quantization

The equations for MPEG quantization are as follows:

$$QF[v][u] = \frac{\frac{F[v][u] \cdot 16}{W[v][u] - k \cdot quantizer\_scale}}{2 \cdot quantizer\_scale}$$

where

$$k = \begin{cases} 0 & \text{for intra coded blocks} \\ sign(QF[v][u]) & \text{for inter coded blocks} \end{cases}$$

Inverse quantization is done using the following formula:

$$F''[v][u] = \begin{cases} 0 & \text{if } QF[v][u] = 0 \\ \frac{(2 \cdot QF[v][u] + k) \cdot W[v][u] \cdot quantizer\_scale}{16} & \text{if } QF[v][u] \neq 0 \end{cases}$$

k is defined similarly.

Second Quantization Method: H.263 Quantization

MPEG coding / decoding is computationally asymmetrical. That is, the encoding process is very costly, and the decoding process is relatively simple. H.263, in comparison, is more symmetrical. This is because H.263 is designed for video teleconferencing, where encoding and decoding happen in real-time for all participants of the conference.

The H.263 method differs in that is doesn't use weighting matrices.

Second Quantization Method: H.263 Quantization

Quantization is performed as follows:

$$|QF[v][u]| = \begin{cases} \frac{|F[v][u]|}{2 \cdot quantizer\_scale} & \text{for intra coded blocks} \\ \frac{|F[v][u]| - quantizer\_scale}{2 \cdot quantizer\_scale} & \text{for inter coded blocks} \end{cases}$$

Inverse Quantization is performed as follows:

$$|F''[v][u]| = \begin{cases} 0 & \text{if } QF[v][u] = 0 \\ (2 \cdot |QF[v][u]| + 1) \cdot quantizer\_scale & \text{if } QF[v][u] \neq 0, \text{ quantizer\_scale is odd} \\ (2 \cdot |QF[v][u]| + 1) \cdot quantizer\_scale - 1 & \text{if } QF[v][u] \neq 0, \text{ quantizer\_scale is even} \end{cases}$$

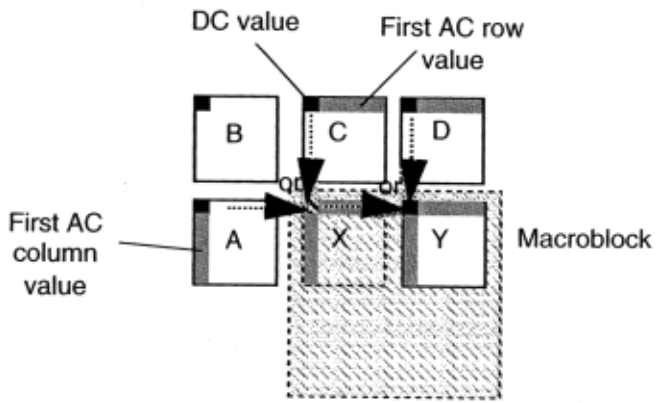
The sign of $QF[v][u]$ is determined by:

$$F''[v][u] = sign(QF[v][u] \cdot |F''[v][u]|$$

AC/DC Prediction for Intra Macroblocks

For some of the AC and DC coefficients, there exists a statistical dependency between them. That is, the value of one can by predicted from a neighboring coefficient. This is exploited in MPEG4 by AC/DC prediction.

AC/DC Prediction for Intra Macroblocks



The figure above shows some 8x8 luminance blocks. Blocks X and Y belong to the same macroblock.

The DC coefficient is the coefficient at $X[0][0]$. Prediction is only carried out on the first row and column of AC coefficients and the DC coefficient.

In the figure, to predict the DC coefficient of X, the gradient of DC coefficients of B to C and B to A is compared, whichever gradient is lower, is used for prediction.

Alternative Scan Modes

After DCT coefficients are quantized, and AC/DC prediction for intra coded blocks has been carried out, the 2D matrix of CDT coefficients is transformed into a 1D vector, and then it is entropy coded using a variable length code table. It is desirable when transforming from matrix to vector, to arrange the vector so that similar coefficients are placed together. In particular, it is desirable to have long runs of zeros. Since high energy coefficients are usually concentrated toward the [0][0] entry of the matrix, a zig-zag scan has traditionally been used.

MPEG 4 defined two additional alternative scanning matrices for use.

Alternative Scan Modes

0	1	2	3	10	11	12	13
4	5	8	9	17	16	15	14
6	7	19	18	26	27	28	29
20	21	24	25	30	31	32	33
22	23	34	35	42	43	44	45
36	37	40	41	46	47	48	49
38	39	50	51	56	57	58	59
52	53	54	55	60	61	62	63

Alternate horizontal scan

0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

Alternate vertical scan

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Zigzag scan

Alternative Scan Modes

The transform can be described using the following c-code.

```
for( v = 0; v < 8; v++ )
  for( u = 0; u < 8; u++ )
    QFS[ scan_pattern[v][u] ] = PQF[v][u];
```

Images with a preference for horizontal or vertical frequencies will be better encoded using one of the alternate scanning matrices.

After the matrix to vector transform, high energy coefficients will appear at the front of the vector, and low energy coefficients (many of which will be zero) will appear at the end of the vector. Entropy encoding can then efficiently compress the vector for storage or transmission.