

Multimedia Communications

Directions &
Innovations



Jerry D. Gibson, Editor

ADOBE - Exhibit 1039, page 1

Multimedia Communications

Directions and Innovations

JERRY D. GIBSON, EDITOR

*Department of Electrical Engineering
Southern Methodist University
Dallas, Texas*



ACADEMIC PRESS

A Harcourt Science and Technology Company

SAN DIEGO / SAN FRANCISCO / NEW YORK / BOSTON / LONDON / SYDNEY / TOKYO

This book is printed on acid-free paper.

Copyright © 2001 by Academic Press

The chapter "Multimedia Conferencing Standards" by David Lindbergh is reprinted from *Digital Compression for Multimedia: Principles and Standards* by Jerry Gibson, Toby Berger, Tom Lookabaugh, and David Lindbergh. Copyright © 1998 by Morgan Kaufmann Publishers, San Francisco. <http://www.mkp.com>. Reprinted with permission.

All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher.

Requests for permission to make copies of any part of the work should be mailed to: Permissions Department, Harcourt, Inc., 6277 Sea Harbor Drive, Orlando, Florida, 32887-6777.

ACADEMIC PRESS

A Harcourt Science and Technology Company

525 B Street, Suite 1900, San Diego, CA 92101-4495, USA

<http://www.academicpress.com>

Academic Press

Harcourt Place, 32 Jamestown Road, London NW1 7BY, UK

Library of Congress Catalog Card Number: 00-104377

International Standard Book Number: 0-12-282160-2

Printed in the United States of America

00 01 02 03 04 QW 9 8 7 6 5 4 3 2 1

MPEG-1 and -2 Compression

TOM LOOKABAUGH

7.1 INTRODUCTION

The Moving Pictures Experts Group (MPEG), an international standards committee, has produced a set of standards for audio and video compression that are good examples of sophisticated compression strategies and that are also commercially important. Starting from an initial focus on video compression at approximately 1.2 Mbit/s on compact disc read-only memory (CD-ROM), the MPEG standards have expanded to become the lingua franca of high quality audio-video compression in the 1990s.

The focus of this chapter is the completed work embodied in the MPEG-1 and -2 standards. As of this writing, MPEG continues to work to produce new standards in several important areas, including advanced audio compression (AAC), digital storage media command and control, MPEG-4, and MPEG-7. This chapter is not an exhaustive restatement of the MPEG-1 and MPEG-2 standards (which run to several hundred pages); rather, it describes the major methods by which MPEG achieves compression and organizes information, to illuminate the potential effectiveness and applications of the standards.

7.2 THE MPEG MODEL

A key to understanding MPEG is understanding both the problems that MPEG set out to address in developing MPEG-1 and -2 (although it is likely MPEG will be applied in many unanticipated places as well) and the fundamental models that underlie the algorithms and that are used to foster interoperability.

7.2.1 Key Applications and Problems

Some of the most important applications to drive the development of MPEG include disk-storage-based multimedia, broadcast of digital video, switched digital video, high-definition television, and networked multimedia.

MPEG-1 had its genesis as the solution to a very specific compression problem: how to best compress an audio-video source to fit into the data rate of a medium (CD-ROM) originally designed to handle uncompressed audio alone. At the time MPEG started, this was considered a difficult goal. Using an uncompressed video rate for 8-bit active video samples [Comité Consultatif International des Radiocommunications or International Telecommunications Union—Radio (CCIR)-601 chroma sampling] of approximately 210 Mbit/s, this requires a rather aggressive 200:1 compression ratio to achieve the slightly greater than 1 Mbit/s or so available after forward error correction and compressed audio on a typical CD-ROM.

Aside from the substantial compression requirement, another requirement of many videos on CD-ROM applications is a reasonable random access capability, that is, the ability to start displaying compressed material at any point in a sequence with predictable and small delay. This is a key attribute, for example, of many interactive game and training materials.

More recently, a new optical disk format with much higher capacity than CD-ROM has been developed. Called digital versatile disc (DVD) (renamed from the original “digital video disc” to include data-storage applications), its higher rates, combined with the use of variable-rate encoding, can provide video quality that surpasses that currently available to consumers through other video media [(e.g., Video Home System (VHS) tape, laserdisc, cable and off-air analog television)], as well as cinematic audio and a number of unique access and format features.

7.2.2 Strategy for Standardization

A key purpose of a standard is to facilitate interoperability. A good standard achieves this while maximizing support for current and as-yet-unknown applications and the development of a variety of implementations. For MPEG, this is achieved by focusing standardization on two related questions:

- What is a legal MPEG bit stream?
- What should be done with it to generate displayable audio-video material?

The first question is answered by a careful specification of the legal syntax of an MPEG bit stream, that is, rules that must be followed in constructing a bit stream. The second is answered by explaining how an idealized decoder would process a legal bit stream to produce decoded audio and video.

This particular strategy for standardization allows a great deal of latitude in the design of encoding systems because the range between truly efficient bit streams (in terms of quality preserved versus bits spent) and inefficient but legal bit streams can be large. There is less latitude in decoder design, since the decoder must not deviate from the idealized decoder in terms of the bit streams it can process, but there is still room for different and clever implementations designed to reduce cost in specific applications, improve robustness to slightly damaged bit streams, and provide different levels of quality in post-MPEG processing designed to prepare a signal for display (e.g., interpolation, digital-to-analog conversion, and composite modulation).

The actual details of what constitutes a legal bit stream are largely conveyed through the specification of the syntax and semantics of such a bit stream.

7.3 MPEG VIDEO

The MPEG video algorithm is a highly refined version of a popular and effective class of video compression algorithms called Motion-Compensated Discrete Cosine Transform (MC-DCT) algorithms. While not universally used for video compression, these algorithms have served as a basis for many proprietary and standard video compression solutions developed in the 1980s and 1990s. The algorithms use the same basic building blocks, including:

- temporal prediction: to exploit redundancy between video pictures
- frequency domain decomposition: the use of the DCT to decompose spatial blocks of image data to exploit statistical and perceptual spatial redundancy
- quantization: selective reduction in precision with which information is transmitted to reduce bit rate while minimizing loss of perceptual quality
- variable-length coding: to exploit statistical redundancy in the symbol sequence resulting from quantization as well as in various types of side information

These basic building blocks provide the bulk of the compression efficiency achieved by the MPEG video algorithm. They are enhanced, however, by a number of detailed special techniques designed to absolutely maximize efficiency and flexibility.

7.3.1 The Basic Algorithm

Figure 7.1 shows the basic encoding and decoding algorithms. The incoming video sequence is preprocessed (interpolated, filtered), then motion estimation is used to help form an effective predictor for the current picture from previously transmitted pictures. The motion vectors from

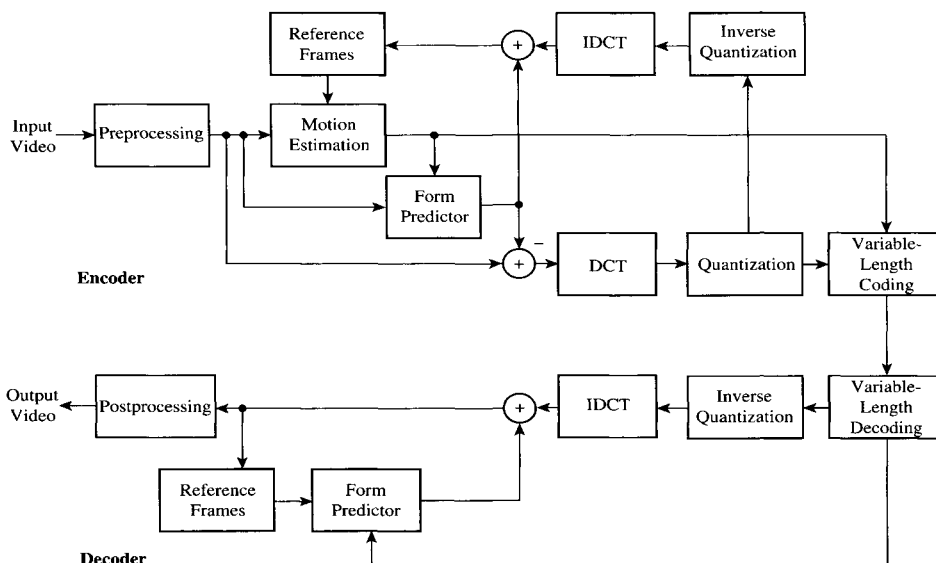


FIGURE 7.1

Block diagram of MPEG encoding and decoding.

motion estimation are sent as side information if used. The predictor for each block is subtracted, and the resulting prediction residual undergoes a Discrete Cosine Transform (DCT). The DCT coefficients are quantized, and the quantized coefficients are variable-length-coded for transmission. The quantized coefficients also undergo reconstruction, inverse DCT, and combination with the predictor, just as they will in the decoder, before forming reference pictures for future motion estimation and prediction.

The decoder decodes the variable-length codes, performs reconstruction of DCT coefficients, inverse DCT, formation of the predictor from previous reconstructed pictures, and summing to form the current reconstructed picture (which may itself serve to predict future received pictures). Postprocessing interpolates and filters the resulting video pictures for display.

7.3.1.1 Representation of Video

The video that MPEG expects to process is composed of a sequence of frames or fields of luma and chroma.

Frame-Based Representation MPEG-1 is restricted to representing video as a sequence of frames. Each frame consists of three rectangular arrays of pixels, one for the luma (Y, black and white) component, and one each for the chroma (Cr and Cb, color difference) components. The luma and chroma definitions are taken from the CCIR-601 standard for representation of uncompressed digital video.

The chroma arrays in MPEG-1 are subsampled by a factor of two both vertically and horizontally relative to the luma array. While MPEG does not specify exactly how the subsampling is to be performed, it does make clear that the decoder will assume subsampling was designed so as to spatially locate the subsampled pixels according to Figure 7.2 and will perform its interpolation of chroma samples accordingly.

Typically, MPEG-2 expects chroma subsampling to be consistent with CCIR-601 prescribed horizontal subsampling. Spatially, this implies the chroma subsampling pattern shown in Figure 7.3, termed 4:2:0 sampling.

Field-Based Representation MPEG-2 is optimized for a wider class of video representations, including, most importantly, field-based sequences. *Fields* are created by dividing each frame into a set of two interlaced fields, with odd lines from the frame belonging to one field and even lines to the other. The fields are transmitted in interlaced video one after the other, separated by

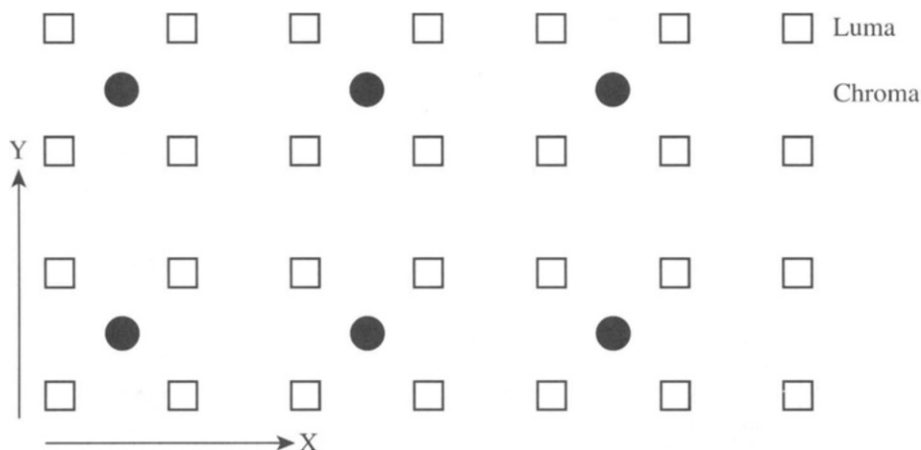
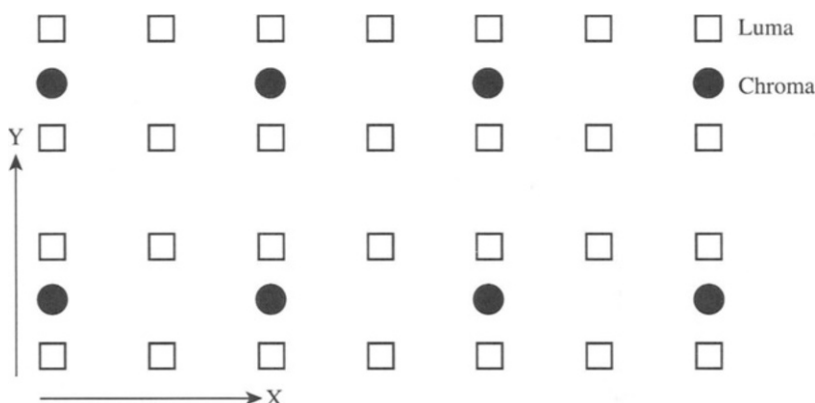


FIGURE 7.2

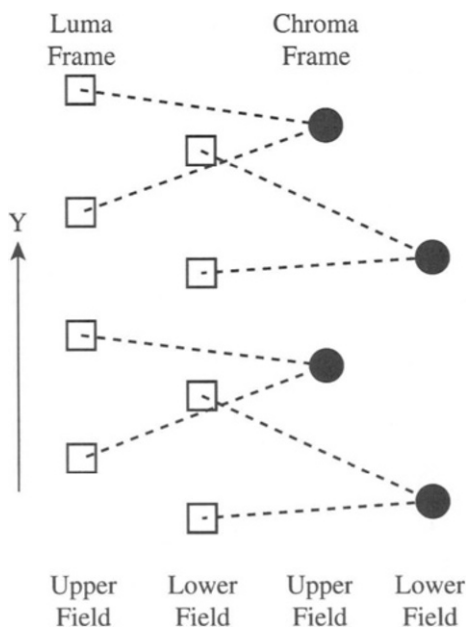
Relationship between luma and chroma subsampling for MPEG-1.

**FIGURE 7.3**

Relationship between luma and chroma subsampling for MPEG-2.

half a frame time. Interlacing of video is in fact a simple form of compression by subsampling. It exploits the fact that the human visual system is least sensitive to scene content that has both high spatial and temporal frequencies (such as a fast-moving item with much detail). An interlaced source cannot represent such scenes effectively, but can build up the full detail of a frame (within two field times) and can also update low-resolution items that are changing every field time; these latter types of material are the most visually important.

For field-based sequences, MPEG-2 expects the chroma associated with each field to be vertically subsampled within the field, yet maintain an expected alignment consistent with frame-based sequences. This leads to a vertical resampling pattern as shown in Figure 7.4.

**FIGURE 7.4**

Relationship between luma and chroma samples vertically and in time for field-based video material.

7.3.2 Temporal Prediction

Temporal prediction exploits the intrinsic redundancy between video pictures. For instance, the background of a scene may be unchanged, with only disturbances in the foreground. In such a case, application of the classic Differential Pulse Code Modulation (DPCM) compression strategy can provide substantial compression, with the prediction being applied to entire video pictures. The typical application of this strategy uses the last picture to predict the current picture and is called *picture differencing*.

We can go beyond picture differencing, however, by noting that when there are changes in the current picture, many times they are caused by the motion of objects. Another common occurrence is motion of the entire picture caused by panning of a camera. Such observations motivate the inclusion of a motion model in the prediction to further reduce the amount of spatial information that must be encoded as a prediction residual. Motion compensation, then, is the application of a motion model to prediction.

7.3.2.1 Motion Compensation

The motion model used by MPEG is blockwise translation. For simplicity, the blocks are chosen to be a single fixed size: 16×16 pixels in the luma component. Since for typical 4:2:0 chroma subsampling, the chroma components are vertically subsampled by a factor of 2 in both dimensions relative to the luma, each chroma component includes an 8×8 block of pixels corresponding to the same spatial region as a 16×16 block from the luma picture. The collection of the 16×16 region from the luma and the two corresponding 8×8 regions from the chroma components is called a *macroblock*.

Motion compensation consists of taking a macroblock from the current picture and determining a spatial offset in the reference picture at which a good prediction of the current macroblock can be found. The offset is called a *motion vector*. The mechanics of motion compensation are to use the vector to extract the predicting block from the reference picture, subtract it, and pass the difference on for further compression. The vector chosen for a macroblock is applied directly to determine the luma predictor, but is scaled by half in both dimensions (corresponding to their subsampled size) before being applied to find the chroma predictors.

Motion estimation consists of finding the best vector to be used in predicting the current macroblock. This is typically the most computationally expensive activity in an MPEG encoder, but can have a substantial impact on compression efficiency.

The effectiveness of motion compensation can be enhanced by allowing the search for an effective prediction region in the reference picture to include not only positions at integral pixel offsets but also fractional pixel offsets. For a fractional pixel offset, the predicting macroblock is constructed by linearly interpolating pixel values relative to the nearest actual pixels. An example of a $1/2$, $1/2$ pixel offset prediction block is shown in Figure 7.5. The dark circle locations have pixel values x calculated as

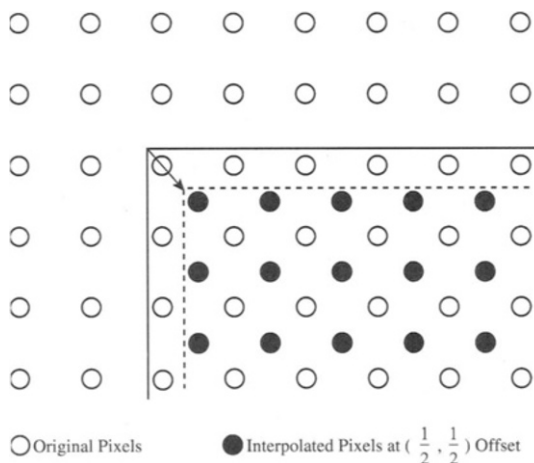
$$x = (a + b + c + d) / 4 \quad (1)$$

where a , b , c , and d are the closest pixels in the original reference picture.

MPEG allows half-pixel interpolation vertically, horizontally, and both (the previous example being both).

7.3.2.2 Picture and Macroblock Prediction Types

One of the key requirements of MPEG is reasonable support for random access. Random access into an arbitrary motion-compensated sequence is difficult for the same reason that starting

**FIGURE 7.5**

Example of half-pixel interpolation in motion compensation.

decoding in a delta modulation or Differential Pulse Code Modulation (DPCM) sequence is difficult: the received sequence consists of a lot of information about changes from one picture to another, but not necessarily the starting point or original reference. This is a dangerous situation in any case from the point of view of dealing with transmission imperfections; the effect of such imperfections on the decoded images can persist indefinitely. These are usually managed through one of two main strategies: refresh and leaky prediction.

Refresh involves periodically sending an entire picture (picture-based refresh) or portion of a picture (region-based refresh) without any prediction, allowing the decoder to resynchronize at that point. Region-based refresh is achieved by guaranteeing that a particular subset of macroblocks are encoded without prediction and rotating this subset through subsequent pictures used in prediction until all macroblocks have been forced to be encoded without prediction. This mechanism reduces the burst of bits required for a picture compressed entirely without prediction, but does not provide a guaranteed random-access entry point or a deterministic time to complete refresh (since nonrefreshed material may be shifted into a just-refreshed area by motion compensation before the refresh cycle is complete).

Leaky prediction involves slightly reducing the effectiveness of the prediction in order that the decoder will progressively “forget” any perturbations in the decoded sequence caused by transmission errors. Both encoder and decoder predictors are multiplied by an agreed upon leak factor less than 1. Since this reduces the accuracy of the predictor, it causes an increase in the bits required to transmit the compressed residual. However, any discrepancy between decoder and encoder predictor memories is reduced during each iteration of prediction, resulting in an exponential decay in the difference.

Only picture-based refresh aids in random access, however, as it allows a predictable location to begin decoding and a bounded time to decode to any target picture.

Given the likelihood that streams would be generated with the regular use of nonpredicted pictures, the MPEG committee decided to investigate some new prediction strategies that might further improve compression efficiency and developed the concept of forward and backward prediction. We first define an *anchor picture* as one that can be used for prediction. Then we define four different prediction strategies that can be used for a particular macroblock, as shown in Table 7.1. Each picture has a declared type that limits the type of macroblock allowed in that picture, as shown in Table 7.2.

Table 7.1 Allowed Prediction Modes by Macroblock Type

Macroblock Type	Prediction
Nonpredicted macroblock	None
Backward-predicted macroblock	References temporally nearest subsequent anchor picture
Forward-predicted macroblock	References temporally nearest previous anchor picture
Bidirectionally predicted macroblock	Averages predictions from temporally nearest previous and subsequent anchor pictures

Table 7.2 Allowed Macroblock Types and Anchor Picture Definition by Picture Type

Picture Type	Anchor Picture	Macroblock Types
I	Yes	Nonpredicted
P	Yes	Nonpredicted
		Forward predicted
B	No	Nonpredicted
		Forward predicted
		Backward predicted
		Bidirectionally predicted

Figure 7.6 uses arrows to show which pictures can be referenced for motion estimation in a typical sequence of pictures. In fact, while not required by the standard, the set of sequences in which there is a fixed spacing between I-pictures and between anchor pictures (I- and P-pictures) is widely used, so widely that a pair of parameters is commonly used to describe these spacings: *N* is the number of pictures from one I-picture (inclusive) to the next (exclusive), and *M* is the number of pictures from one anchor picture (inclusive) to the next (exclusive). So, the pattern shown in Figure 7.6 would be called an *N* = 9, *M* = 3 pattern.

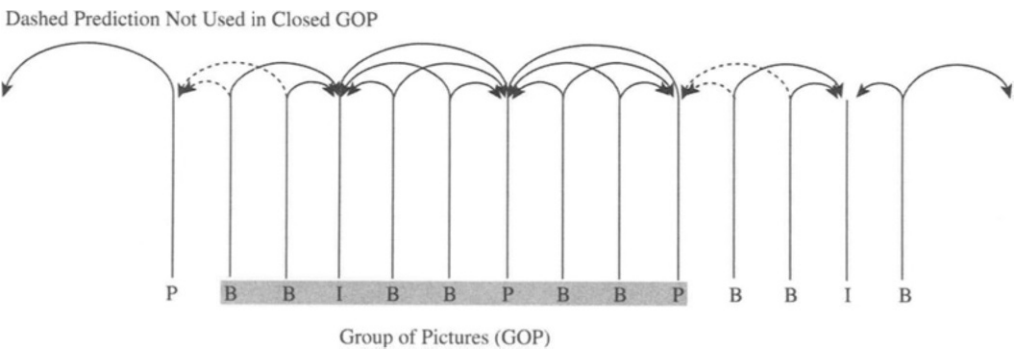


FIGURE 7.6
Typical group of pictures and allowed predictors for pictures.

There are several implications to this set of possible predictions:

- Using bidirectionally predicted blocks allows effective prediction of uncovered background, areas of the current picture that were not visible in the past but are visible in the future.
- Bidirectional prediction can provide for interpolation equivalent to an even finer degree than the half-pixel interpolation for motion compensation already allowed.
- Bidirectional prediction can reduce noise in the predictor.
- Bidirectional prediction increases motion estimation complexity in two ways: First, it obviously requires twice as much work to perform motion estimation in two different anchor pictures. More significantly, though, since anchor pictures are now possibly separated by several picture times, we need a larger overall motion estimation range in order to track objects of a given velocity.
- Since B-pictures cannot themselves be used to predict other pictures, the quality of compression is solely determined by the visual acceptability of the result. For P- and I-pictures, since the reconstruction will also be used for other predictions, quality must reflect both perceptual fidelity and prediction efficiency. This allows a substantial reduction in the bits allocated to B-pictures relative to I- and P-pictures. A ratio of 5:3:1 in bits spent on I-, P-, and B-pictures is not uncommon. These ratios can and should vary dynamically, though.

7.3.2.3 *Display and Transmit Order*

Combining the use of B-pictures with a desire to economize on the use of memory in high-volume decoders leads to a difference in the order in which pictures are transmitted versus the order in which they are displayed. The reordering is intended to ensure that, regardless of the spacing between anchor pictures, a decoder need only have enough memory to store two anchor pictures and the scratch memory for the picture currently being decoded. This is achieved by transmitting the anchor picture's compressed version before those of the B-pictures that will precede it in display.

7.3.2.4 *Field and Frame Prediction*

While MPEG-1 does not consider the possibility of field-based video representations, MPEG-2 specifically does allow for increased compression efficiency on interlaced material. One way in which this is accommodated is by allowing either field- or frame-based prediction for each macroblock (often termed *adaptive field/frame compression*). In the case of field-based prediction, the macroblock is divided into two field macroblocks. The top field macroblock is predicted from the top fields in one or two anchor pictures using the modes appropriate for the picture type. The lower field macroblock is predicted from the lower fields in one or two anchor pictures. Note that the subdivision of chroma here corresponds to the chroma subsampling constraints for field representation described previously.

Field-based prediction can be highly effective in instances of substantial horizontal acceleration. In these cases, an object captured in two different fields appears to have serrated vertical edges when viewed as a frame, with the depth of the serration varying with time due to acceleration. Frame-based motion estimation is ineffective in this case and leaves a good deal of difficult-to-compress high-frequency serrations left over. Field-based prediction can perform quite well. On the other hand, for more benign motion cases, frame-based prediction can more effectively exploit vertical redundancy. Allowing a choice between the modes to occur on a

macroblock basis gives the best of both worlds at the cost of a slight increase in overhead (the net effect being in favor of allowing the choice).

7.3.3 Frequency Domain Decomposition

After motion compensation has been performed, the residual information (or original picture information in the case of a nonpredicted block) is transformed using the DCT. To prepare the macroblock for transformation, six 8×8 blocks are extracted from the macroblock, four from the luma picture and one each from each of the chroma pictures. The luma blocks for MPEG-1 and the frame DCT mode of MPEG-2 are shown in Figure 7.7; the field DCT mode of MPEG-2 is shown in Figure 7.8.

The field DCT mode is useful for much the same reason that field motion compensation is useful: Material with a high degree of horizontal motion, especially acceleration, captured using

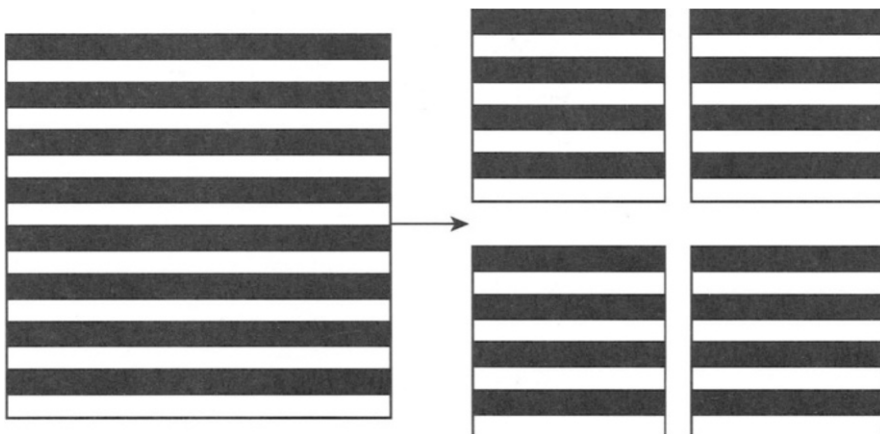


FIGURE 7.7

Mapping from 16×16 block to 8×8 blocks for frame-organized data.

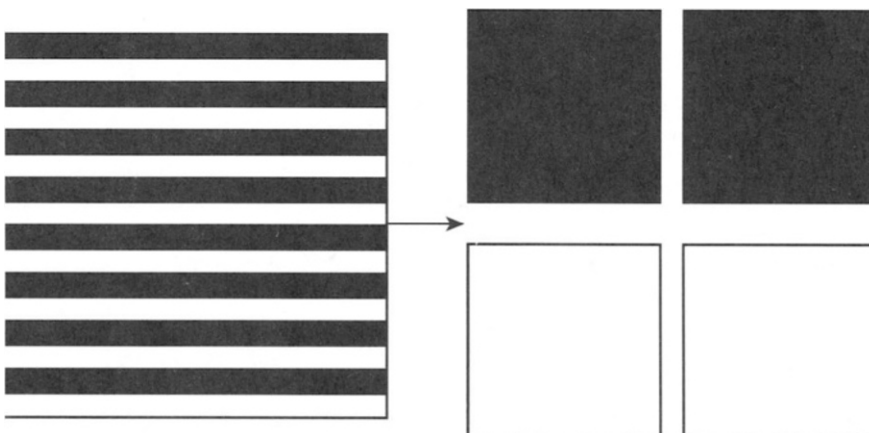


FIGURE 7.8

Mapping from 16×16 block to 8×8 blocks for field-organized data.

an interlaced video standard, tends to produce difficult-to-compress high frequencies when transformed using a frame-based DCT. The choice of frame versus field mode for motion compensation and DCT is not bound together, and there is a small benefit to be had by making the decision independently.

7.3.3.1 Discrete Cosine Transform

The MPEG standard does not specify exactly how to perform the 8×8 DCT and Inverse Discrete Cosine Transform (IDCT) required. This is too limiting given that many different software and hardware implementations of MPEG are expected. Instead, MPEG calls for the IDCT as implemented to be able to pass a test developed as part of the ITU-T standardization of H.261, a videoconferencing standard. In this test, Institute of Electrical and Electronics Engineers (IEEE) P1180/D2, the proposed IDCT inverse transforms a particular set of blocks, and the results are compared to an idealized set of results. Providing the proposed IDCT has less than a specified number of discrepancies at each of several magnitudes, it is an acceptable IDCT.

Allowing similar but not identical IDCTs in different implementations will cause mismatch in the encoders and decoders of different manufacturers. Since the encoder includes an IDCT for the purpose of reproducing the pictures that the decoder generates, the fact that an encoder and decoder have slightly different DCTs will introduce a gradually increasing difference between these pictures, resulting in inappropriate coding decisions on the encoder's part. This phenomenon is managed by limiting to 132 the number of times a specific macroblock location may be coded as predicted in P-pictures before that macroblock location must be coded without prediction. Moreover, MPEG-2 added some specific quantization rules that were found to further mitigate the problem of mismatch.

7.3.4 Quantization

After transformation, the DCT coefficients typically have 12 bits or more of precision. On the face of it, this is not a very auspicious compression, since the data started with 8 bits per pixel. Quantization is the key step that exploits the efforts provided by motion compensation and the DCT to reduce the bits required, since the information has now been organized so that many coefficients are essentially irrelevant to the final reproduction quality and only a few need be treated with care.

7.3.4.1 Quantizer Step Size

MPEG applies a uniform step-size quantizer to each coefficient. However, the step size of the quantizer may vary from coefficient to coefficient and macroblock to macroblock. In fact, the quantizer step size is determined at the decoder by the following equation:

$$ss = qf[m,n] \times qs \quad (2)$$

The factor $qf[m,n]$ is dependent on the location of the coefficient within a block. The factor qs is the base quantizer step size. This allows emphasis on more perceptually important lower frequencies. MPEG provides two default weighting matrices $qf[m,n]$ for use on predicted blocks and nonpredicted blocks, shown in Figures 7.9 and 7.10. Note that for nonpredicted blocks, the DCT DC coefficient, the upper left-hand coefficient that is proportional to the average value of the space domain block, is not quantized using a weighting matrix value; hence that location in the weighting matrix has an * in Figure 7.9. This unusual treatment reflects the fact that the eye is highly sensitive to errors in the DC level of nonpredicted blocks and tends to quickly recognize these as blocking or tiling distortion in the reconstructed picture.

*	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

FIGURE 7.9
Intraframe quantizer weighting matrix.

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

FIGURE 7.10
Interframe quantizer weighting matrix.

7.3.5 Variable-Length Coding

Both the quantized coefficients and several different types of side information (macroblock prediction type, motion vectors, etc.) exhibit statistical concentration so that the overall average bit rate can be lowered by using variable-length coding. Note that the use of variable-length coding will necessitate buffering when MPEG is to operate over a fixed-rate channel.

7.3.5.1 Modified Huffman Codes

The type of variable-length code used throughout MPEG is a modified Huffman code. Huffman coding provides optimal variable-length coding for the chosen alphabet. However, for large alphabets, storing, encoding, and decoding the Huffman code can be expensive. Hence, the table size is generally restricted to a subset consisting of the most probable symbols. An additional codeword is added to the code, termed “ESCAPE.” When an input symbol is observed that does not belong to the chosen high-probability symbol set, it does not have its own Huffman codeword; instead, the ESCAPE codeword is issued followed by the explicit (i.e., standard indexed value) input symbol. A second modification to the MPEG Huffman codes for video is that the all-zero codeword is not used to avoid emulation of a start code via a legitimate sequence of other codes.

7.3.5.2 Two-Dimension to One-Dimension Conversion

The quantized DCT coefficients form the bulk of the material that needs to be variable-length encoded. Empirically, though, after quantization, the 8 × 8 blocks of DCT coefficients tend to exhibit a substantial number of zeros, particularly in the higher frequencies, as in Figure 7.11.

MPEG exploits this phenomenon by first converting the 8 × 8 array into a one-dimensional array and then applying coding that can exploit long runs of zeros. Both MPEG-1 and MPEG-2 use a zigzag scan for cases, such as in Figure 7.11, where there are a large number of zeros in

	Increasing Horizontal Frequency →							
Increasing Vertical Frequency ↓	−42	12	3	−1	0	0	0	1
	−19	14	−8	0	2	0	0	0
	16	8	−9	3	0	0	0	0
	3	−2	0	0	1	0	0	0
	3	0	0	1	−1	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

FIGURE 7.11
Example of an 8 × 8 array of quantized DCT coefficients.

the lower right-hand quadrant. MPEG-2 provides an alternative, vertical scan, which is often more effective, especially when a field-based DCT has been applied. The scan path of zigzag and vertical scans is shown in Figure 7.12.

The decoder will perform the inverse scanning as part of its reconstruction process.

7.3.5.3 Runlength Amplitude Coding

Once a 64-symbol vector has been created by scanning the 8×8 array for each block, runlength amplitude coding is used. First, the DC quantized coefficient (containing the block average) is treated specially, receiving its own dedicated Huffman code, because the DC quantized coefficient tends to have a unique statistical characteristic relative to other quantized coefficients. Moreover, since there is some redundancy between adjacent DC quantized coefficients in non-predicted blocks, only the difference between these is Huffman coded. The remaining quantized coefficients are parsed into a sequence of runs, where a run is defined as zero or more 0s followed by a single nonzero value. This becomes a new symbol for Huffman coding, consisting of a runlength-amplitude pair.

7.3.5.4 Side Information Coding

There are a variety of different types of side information, designed in general to reduce the number of blocks that actually have to undergo runlength-amplitude coding and to reduce the amount of information in those that do. The most important of these are:

- Macroblock address
- Macroblock type
- Coded pattern
- Motion vectors

7.3.6 Rate Control

One of the most important encoder design issues is how to include an effective rate control system into a compressor. As mentioned earlier, once variable-length coding has been applied, some

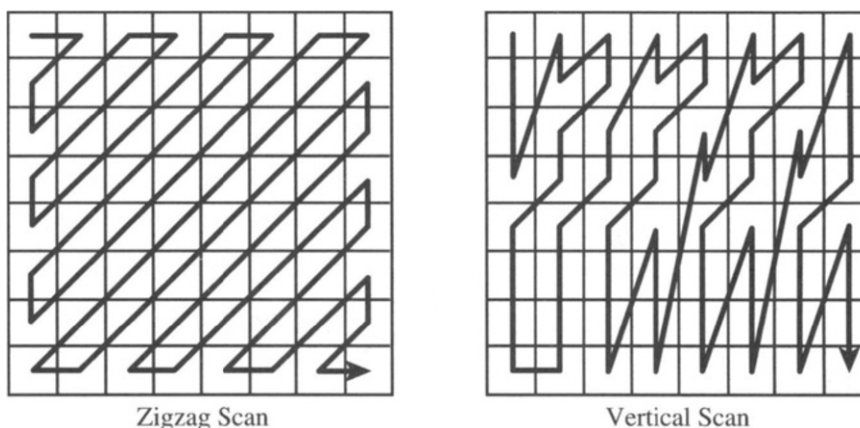


FIGURE 7.12

Coefficient scan pattern for zigzag and vertical scans.

buffering is required to absorb instantaneous fluctuations in bit rate if the stream is to be delivered over a fixed-rate channel. Such buffering must be available at both encoder and decoder. A tenet of MPEG is that once the decoder buffer size has been specified, it must be possible to decode a legal bit stream in that buffer without overflowing or underflowing (except for the low-delay mode of MPEG-2, which does allow underflows).

7.3.6.1 Fixed-Rate Channels

For fixed-rate channels, with a little effort, the requirement on the decoder buffer can be made equivalent to a requirement on the encoder buffer as follows:

- If the encoder and decoder buffer are of the same size,
- and the encoder compresses units at the pace at which they arrive,
- and the decoder decompresses units at the pace at which they are to be displayed,
- and the decoding starts when the decoder buffer's fullness equals the buffer size minus the encoder's buffer fullness,
- and the encoder ensures that the encoder buffer never overflows (by throttling information when necessary) and never underflows (by using stuffing to send filler bits when necessary),
- then the decoder buffer will never underflow or overflow.

7.3.6.2 Variable-Rate Channels

MPEG supports variable-rate channels. The overriding constraint is that the decoder's buffer not overflow or underflow during decoding. Variable-rate channels come in two major classes: slave and master. A slave variable-rate channel has the rate selected by the MPEG system. A master variable-rate channel tells the MPEG system what the current rate is. Variable-rate channels are often also constrained by a maximum and minimum allowable instantaneous bit rate.

7.3.7 Constrained Parameters, Levels, and Profiles

The MPEG video algorithm provides an enormous flexibility in terms of the image sizes, bit rates, and other key parameters that can be supported. It is unreasonable to expect simple applications to bear the cost of decoding at the substantially higher complexity of the worst-case use of the standard, particularly if the simpler applications will be built in highly cost-sensitive high-volume applications. Hence, MPEG has defined compliance points within the standard.

For MPEG-1, the *constrained parameters* case provides the following set of specific restrictions that are reasonable for the kind of moderate-resolution, multimedia applications in the 1–3 Mbit/s range for which MPEG-1 was optimized.

- Horizontal size less than or equal to 720 pixels
- Vertical size less than or equal to 576 pixels
- Total number of macroblocks/picture less than or equal to 396
- Total number of macroblocks per second less than or equal to $396 \times 25 = 330 \times 30$
- Picture rate less than or equal to 30 pictures per second
- Bit rate less than or equal to 1.86 Mbit/s
- Decoder buffer less than or equal to 376,832 bits

Although the first two constraints seem to allow large image sizes (up to the full resolution of broadcast digital CCIR-601), the third and fourth constraints are more limiting in this regard. For instance, the Standard Intermediate Format (SIF) sizes of $352\text{H} \times 240\text{V}$ (NTSC countries) and $352\text{H} \times 288$ (PAL countries) just satisfy these constraints; a CCIR-601 resolution would not. Much of the commodity silicon targeted at MPEG-1 applications is designed to the constrained parameters level.

MPEG-2 is designed to address a much larger number of potential applications, and in order to be cost-effective in each, it has a much larger set of compliance points. These are indexed by *levels*, which set a rough limit on processing power based on image size, and *profiles*, which restrict the algorithmic features available. Table 7.3 shows the grid of compliance points for MPEG-2.

The compliance point addressed by a particular system is stated as “x profile at y level.” The most popular compliance point is, not surprisingly, main profile at main level (or MP ML, for short). Following are brief discussions of each of the levels:

- Low: This sets constraints on image size of approximately half CCIR-601 vertically and horizontally, namely, $352\text{H} \times 288\text{V}$.
- Main: CCIR-601 size images, $720 \times 576\text{V} \times 25\text{Hz}$ or $720\text{H} \times 480\text{V} \times 29.97\text{Hz}$.
- High 1440: HDTV up to $1440\text{H} \times 1152\text{V}$.
- High: HDTV up to $1920\text{H} \times 1152\text{V}$.

Following are brief descriptions of the profiles:

- Simple: No B-pictures. This allows a reduction in the memory required at decoders of as much as 8 Mbits, at some cost in bit-rate efficiency.
- Main: The material in this chapter describes most of the aspects of the main profile.
- SNR: Adds a type of scalability based on transmitting two bit streams. The first, when decoded, will provide a reasonable reproduction. The second provides additional information to refine the reconstructed DCT coefficients, so that when both are decoded together, an excellent reconstruction results. Such scalability allows simpler and more complex decoders to share the same bit stream.
- Spatial: Adds a different type of scalability, in which the first stream provides a low-resolution reproduction and the combination of both streams provides a full-resolution reproduction.
- High: Allows SNR, spatial, and an additional type of scalability based on the first stream providing a low picture-rate reconstruction and both streams providing full picture rate, as well as a number of other high-complexity options.

Table 7.3 Defined Levels and Profiles for MPEG-2

Level	Profile				
	Simple	Main	SNR	Spatial	High
High		X			X
High 1440		X		X	X
Main	X	X	X		X
Low		X	X		

7.4 SUMMARY

The MPEG standard suite is a comprehensive set of video and audio decompression algorithm specifications, as well as specifications of how the compressed bit streams can be multiplexed together and how the resulting decoded media can be synchronized. MPEG is used in a number of substantial commercial applications, including playback of media from disk storage, digital broadcast of audio-video programming over a variety of channels, point-to-point switched connections for delivery of digital audio-video material, high-definition television, and networked multimedia, as well as in a large number of specialized commercial and noncommercial applications.

MPEG compression relies on a number of fundamental techniques described in this chapter and in other specific applications. For video, these include temporal prediction from video frame to video frame based on a simple model of motion, the application of the DCT to prediction residuals, quantization to select the extent to which information about individual DCT coefficients will be retained, and the application of both runlength and Huffman lossless variable-rate codes to extract remaining statistical redundancy in the quantized DCT coefficients.