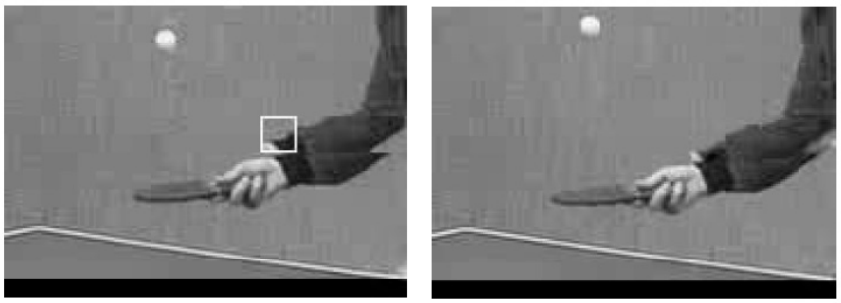


(a) Spatial error propagation in the third frame (b) Temporal error propagation in the sixth frame

Figure 9.2: Spatial and temporal error propagation due to a single-bit error in QSIF TABLE TENNIS H.263 encoded at 10 frames/s (46 kbits/s)

Figure 9.2 shows an example of spatial and temporal error propagation in the QSIF TABLE TENNIS sequence H.263 encoded[1] at a frame rate of 10 frames/s (about 46 kbits/s). Figure 9.2(a) shows the third frame of the sequence, where a single bit error hits the macroblock in the position shown. This error converts the VLC codeword representing the vertical vector difference to another valid codeword of the same length. This causes an error in the compensation of this particular macroblock. In addition, because of the predictive coding of motion vectors, this error propagates spatially to all macroblocks to the right and up to the border of the frame. Figure 9.2(b) shows how motion-compensated prediction caused the errors in the third frame to propagate temporally to the sixth frame. This example shows how serious even a single bit error can be and clearly highlights the need for error detection and control techniques.

9.5 Error Detection

Before being able to combat the effects of errors, it is first necessary to detect whether and where errors have occurred. Error detection can be performed by the channel decoder and/or the source decoder.

One method for error detection is the use of header information. This can be used by both the channel decoder and the source decoder. For example, in a packet-based network like ATM, each packet contains a header with a

[1]Telenor H.263 implementation was used. The luma component was zero padded to 128 lines to be a multiple of 16. The chroma components were also zero padded correspondingly. The optional mode to insert synchronization codewords at the start of each GOB was switched on. All other optional modes were switched off. The initial quantization parameter was set to 10.

sequence number subfield. This sequence number can be used to detect packet losses at the channel decoder. Similarly, the group-number (GN) codeword in an H.263 GOB header can be used to detect errors at the source decoder.

Another method that can be used by both the channel decoder and the source decoder is *forward error correction* (FEC). For example, Annex H of the H.263 standard provides an optional FEC mode. In this mode, 18 parity bits are used to provide error detection and correction for each 493 video bits.

A commonly used method at the source decoder is the detection of syntax and semantic violations. Examples of such violations are:

- An illegal codeword is detected.
- An invalid number of units is decoded. For example, the number of decoded DCT coefficients within a block is invalid, the number of decoded blocks within a MB is invalid, the number of decoded MBs within a GOB is invalid, or the number of decoded GOBs within a frame is invalid.
- A decoded motion vector points outside the permissible range.
- A decoded quantization parameter is out of range.

Another method that can be used at the source decoder is the detection of violations to the general characteristics of natural video signals, for example, the detection of strong discontinuities at the borders of blocks, blocks with highly saturated colours (e.g., pink and green), or blocks where most pels need clipping.

None of these methods guarantee finding all errors within a video bitstream. In fact, the last method may sometimes detect an error-free block as an erroneous one. In practical systems, different combinations of these methods are employed.

Having detected the occurrence of errors and identified their locations, a number of methods can be used to combat the effects of errors on the video bitstream. The following three sections describe three categories of error-resilience techniques: *forward* techniques, *postprocessing* techniques, and *interactive* techniques. The three sections follow closely the classification used in the comprehensive reviews by Wang *et al.* [179, 180].

9.6 Forward Techniques

In forward techniques, the *encoder* plays the primary role. Such techniques work by adding a controlled amount of redundancy to the video bitstream. This means that they sacrifice some coding efficiency to gain in terms of error

resilience. Some techniques are designed to minimize the effects of transmission errors, some are designed to make error handling at the decoder more effective, and others are designed to guarantee a basic level of quality while providing graceful degradation in the presence of transmission errors. Examples of forward techniques are briefly described in the following subsections.

9.6.1 Forward Error Correction (FEC)

Forward error correction works by adding redundant bits to a bitstream to help the decoder detect and correct some transmission errors without the need for retransmission. The name *forward* stems from the fact that the flow of data is always in the forward direction (i.e., from encoder to decoder).

For example, in block codes the transmitted bitstream is divided into blocks of k bits. Each block is then appended with r parity bits to form an n-bit codeword. This is called an (n,k) code.

For example, Annex H of the H.263 standard provides an optional FEC mode. This mode uses a $(511,493)$ BCH (*Bose-Chaudhuri-Hocquenghem*) code. Blocks of $k=493$ bits (consisting of 492 video bits and 1 fill indicator bit) are appended with $r=18$ parity bits to form a codeword of $n=511$ bits. Use of this mode allows the detection of double-bit errors and the correction of single-bit errors within each block.

9.6.2 Robust Waveform Coding

As already discussed, the waveform encoder in a typical video communication system works by removing statistical and psychovisual redundancies present in the input video. Robust waveform coding techniques, however, intentionally keep (or even add) some redundancy to achieve error resilience. Examples of such techniques are given next.

9.6.2.1 Adding Redundant Information

This technique adds auxiliary information or repeats some previously coded information to help error handling at the decoder. For example, as is shown in Section 9.7, a powerful technique for error concealment is temporal concealment. The performance of this technique is highly dependent on the availability of motion information for the damaged blocks. Thus, this technique is usually used for concealing INTER macroblocks. In MPEG-2, however, the encoder can optionally send auxiliary motion vectors for INTRA macroblocks. In the presence of errors, such vectors can be used to temporally conceal damaged macroblocks.

Another example is the header extension code (HEC) included by MPEG-4 in packet headers. If this bit is set to "1," then some data, like timing

information and VOP coding type, is repeated from the VOP header. This helps error detection and resynchronization. A similar example is the picture header repetition allowed by the optional additional supplemental enhancement information mode (annex W) of H.263++.

9.6.2.2 Using INTRA Refresh

An effective way to stop temporal error propagation is to periodically encode pictures in INTRA mode. Given the large number of bits consumed by INTRA pictures, this leads to a significant increase in the total bit rate. A more suitable approach for applications like mobile video communication is to use INTRA refresh on the macroblock level. By controlling the *number* and *spatial location* of INTRA MBs, INTRA refresh can be a very efficient and scalable error-resilience tool.

Obviously, the required number of INTRA MBs is highly dependent on the channel quality and capacity. Such information is usually available to the encoder. For example, in mobile networks, antenna parameters can give an indication of the channel quality. In Ref. 181, Haskell and Messerschmitt discuss how to select a suitable number of INTRA MBs.

There are many methods for selecting the spatial location of INTRA MBs within frames. One method is to choose the locations randomly [181, 182]. Another method is to follow a raster scanning order. In Ref. 183 the INTRA MBs are placed adaptively in regions with high activity.

Recently, a very powerful technique for deciding both the number and spatial locations of INTRA MBs has been proposed by Côté *et al.* [182, 184]. In Ref. 182 they propose a rate-distortion optimized mode selection method for packet lossy networks. This method takes into account the channel conditions and the error concealment method used at the decoder. In Ref. 184 they apply the same method to bit-oriented networks.

Obviously, if there is a feedback channel from the decoder, then information regarding the number and locations of damaged MBs can help the encoder to better decide the number and locations of INTRA MBs.

9.6.2.3 Using Restricted Prediction

In this technique, prediction is limited within nonoverlapping spatial and/or temporal regions. This clearly limits temporal and/or spatial error propagation.

For example, in the independent segment decoding mode (annex R) of H.263+, video pictures are divided into segments. Each video picture segment is then encoded with complete independence from all other segments in the same picture, and also with complete independence from all data outside the corresponding segment in the reference picture(s). For example, motion vectors

of blocks outside the current segment cannot be used when calculating the current motion vector predictor. Similarly, motion vectors of blocks outside the current segment cannot be used as remote motion vectors for overlapped block-motion compensation when the advanced prediction mode is in use. In addition, no motion vectors are allowed to reference areas outside the corresponding segment in the reference picture.

9.6.3 Robust Entropy Coding

In this case, redundancy is added at the entropy encoder. Examples of robust entropy coding techniques are discussed next.

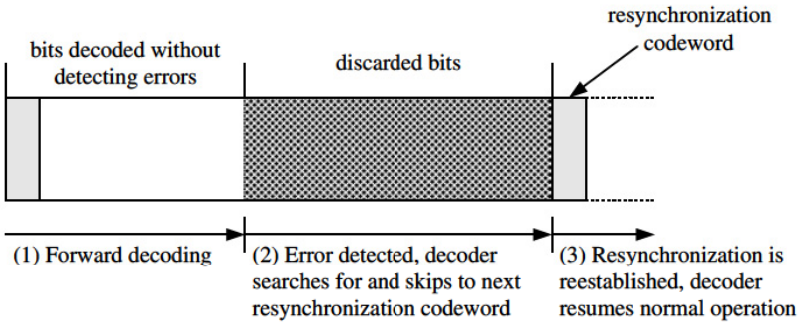
9.6.3.1 Resynchronization Codewords

As already discussed, one of the disadvantages of VLC coding is that errors in the bitstream can cause loss of synchronization between the encoder and the decoder, and this leads to spatial error propagation. One way to reduce this effect is to insert unique markers called *resynchronization codewords* in the bitstream. When an error is detected, the decoder skips the remaining bits until it finds a resynchronization codeword. This reestablishes synchronization with the encoder, and the decoder then proceeds to decode from that point on. This is illustrated in Figure 9.3(a).

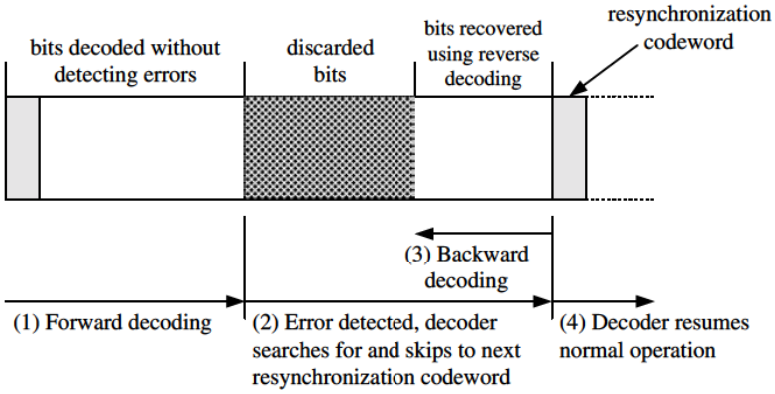
Resynchronization codewords can be inserted at regular intervals in the spatial domain, as illustrated in Figure 9.4(a). For example, version 1 of H.263 adopts a GOB-based resynchronization approach. This means that a resynchronization codeword is inserted every time a fixed number of macroblocks has been encoded. A disadvantage of this approach is that, since the number of bits can vary between macroblocks, the resynchronization codewords will most likely be unevenly spaced throughout the bitstream. Therefore, certain parts of the sequence, such as high-motion areas with high bit content, will be more susceptible to errors and will also be more difficult to conceal.

A more robust approach is to insert resynchronization codewords at regular intervals in the bit domain, as illustrated in Figure 9.4(b). For example, MPEG-4 adopts a packet-based resynchronization approach. In this approach each packet contains approximately the same number of bits. This means that the resynchronization codewords are almost periodic in the bitstream. A similar approach has also been adopted in the slice structured mode (annex K) of H.263+.

Another problem with VLC coding is that errors can emulate the occurrence of resynchronization codewords. To reduce this effect, MPEG-4 provides a second resynchronization approach called *fixed-interval synchronization*. In this approach, resynchronization codewords appear only at legal fixed-interval



(a) Resynchronization with normal VLC coding

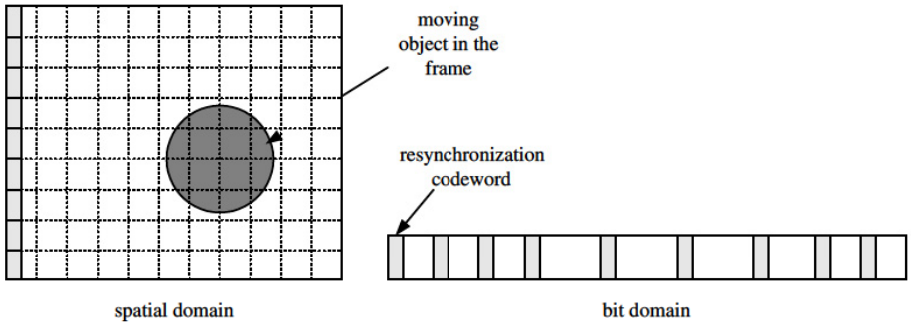


(b) Resynchronization with reversible VLC (RVLC) coding

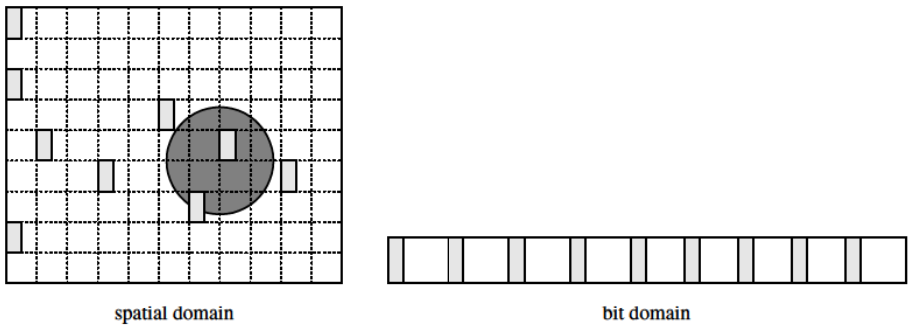
Figure 9.3: Resynchronization using synchronization codewords

locations in the bitstream. Thus, only codewords at those legal locations will be used by the decoder to reestablish synchronization.

As described in Section 9.4, loss of synchronization appears in two forms: loss of codeword synchronization and loss of positional (or coefficient) synchronization. Inserting resynchronization codewords reduces the effect of loss of codeword synchronization. In order to reduce the effect of loss of positional synchronization, resynchronization codewords are usually followed by some positional information, like the address and the temporal reference of the macroblock immediately following the resynchronization codeword. This allows the decoder to resume its normal operation.



(a) Resynchronization codewords at regular intervals in the spatial-domain



(b) Resynchronization codewords at regular intervals in the bit-domain

Figure 9.4: Resynchronisation codewords at regular intervals

9.6.3.2 The Error-Resilience Entropy Code (EREC)

An interesting alternative to inserting resynchronisation codewords is the *error resilience entropy code* (EREC) [177, 185]. The EREC takes variable-length blocks of data and rearranges them into fixed-length slots. For example, assume that there are N variable-length blocks with lengths b_i, $i = 1 \ldots N$. The encoder first chooses a total data size $T \geqslant \sum b_i$, which is sufficient to encode all the data. This total data size is split into N slots of fixed lengths s_i, $i = 1 \ldots N$. An N-stage algorithm is then used to place the data from the variable-length blocks into the fixed-length slots. At each stage n, a block i with data left unplaced searches slot $j = i + \phi_n \pmod N$ for space to place some or all of the remaining data. Here, ϕ_n is an offset sequence that is usually pseudo-random.

Figure 9.5 shows an example of the EREC algorithm. In this case, there are $N = 6$ variable-length blocks, with lengths 11, 9, 4, 3, 9, and 6 bits. The total data size is chosen as $T = 42$ and is divided into $N = 6$ slots, with a length of $s_i = 7$ bits each. The offset sequence is $\phi_n = \{0, 1, 2, 3, 4, 5, 6\}$. In stage 1 of

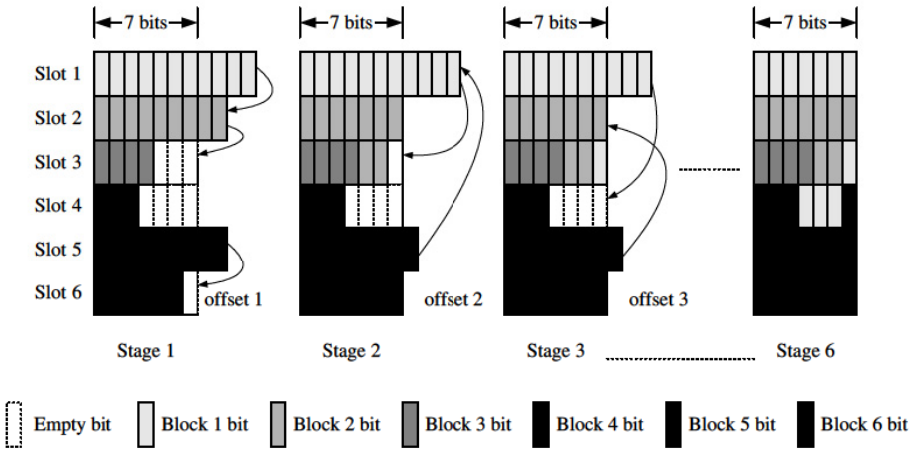


Figure 9.5: Example of the EREC algorithm

the algorithm, blocks 3, 4, and 6 are completely placed into the corresponding slots, with some leftover space in those slots. Blocks 1, 2, and 5, however, are only partially placed in the corresponding slots and have some bits left to be placed in empty spaces in other slots. According to the offset sequence, block 1 searches slot 2 for empty space, block 2 searches slot 3, and block 5 searches slot 6. Both blocks 2 and 5 find empty spaces. Thus, in stage 2, all the remaining bits from block 2 are placed in slot 3, whereas some of the remaining bits of block 5 are placed in slot 6. Since block 1 did not find empty spaces in slot 2, then, according to the offset sequence, it searches slot 3, and so on. By the end of stage 6, all data bits are placed in the slots. The decoder operates in a similar manner. Thus bits in a slot are decoded and placed in a block until an end-of-block codeword is encountered.

In the presence of errors, the resilience provided by the EREC algorithm is due to two factors. First, each block starts at a known position in the bitstream (i.e., the start of the corresponding slot). Thus, in the case of loss of synchronization, the decoder simply jumps to the start of the next slot without the need for resynchronization codewords. Second, subjectively less important data (e.g., high-frequency DCT coefficients) are usually placed in later stages of the algorithm. With the EREC algorithm, most error propagation effects (due, for example, to missing or falsely detecting end-of-block codewords) hit data placed at later stages of the algorithm rather than the more important data at the start of the slots.

9.6.3.3 Reversible Variable-Length Coding (RVLC)

Reversible VLC codewords are designed to be decoded both in the forward and backward directions. As already described, when an error is detected in

the bitstream, the decoder discards all bits until the next resynchronization codeword, where synchronization is reestablished and the decoder resumes its decoding process. The discarded bits may well be correctly received but cannot be decoded correctly due to loss of synchronization. In the case of RVLCs, when the decoder identifies the next resynchronization codeword, instead of discarding all preceding bits, the decoder starts decoding in the reverse direction to recover and utilize some of those bits. This is illustrated in Figure 9.3(b).

Reversible variable-length coding has been adopted in most recent standardization efforts. For example, the modified unrestricted motion vector mode (modified annex D) of H.263+ uses RVLC to encode motion vector differences, the data partitioned slice mode (annex V) of H.263++ uses RVLC to encode header and motion information, and MPEG-4 uses RVLC to encode texture information.

9.6.4 Layered Coding with Prioritization

In layered coding, video is encoded into a base layer and one or more enhancement layers. The base layer is separately decodable and provides a basic level of perceived quality. The enhancement layers can be decoded to incrementally improve this quality.

Layered coding can be useful when applied over heterogenous networks with varying bandwidth capacity. However, to be used as an error-resilience tool, layered coding must be combined with prioritized transmission or what is commonly known as *unequal error protection*. In this case, the base layer is transmitted with higher priority or a higher degree of error protection. For example, in Ref. 186 Ghanbari introduced the concept of layered coding with prioritized transmission to increase the robustness of video against cell loss in ATM networks. In this technique, the encoder generates two bitstreams. The base-layer bitstream contains the most vital video information, whereas the enhancement-layer bitstream contains residual information to improve the quality of the base layer. The base layer is then transmitted using high-priority ATM cells, whereas the enhancement layer is transmitted using low-priority cells. When traffic congestion occurs, low-priority cells are discarded first. Another example is the power control method proposed in Ref. 187. In this method, when video is transmitted over a wireless network, more power is used to transmit the base layer, whereas less power is used to transmit the enhancement layers.

There are many ways to encode video into more than one layer. For example, the base layer can include a low-frame-rate version of video, whereas the enhancement layers can contain frames used to increase the frame rate. This is usually referred to as *temporal scalability*. Another method is when the base

layer contains a coarsely quantized version of video, whereas the enhancement layers carry the error between the original version and this coarsely quantized version. This is known as *SNR scalability*. Another form is *spatial scalability*. This is very similar to SNR scalability. The only difference is that pictures in the base layer are subsampled to a smaller size. Yet another form of layered coding is known as *data partitioning*. In this case, the base layer contains vital video information like headers, motion vectors, and low-frequency DCT coefficients. Other information, like high-frequency DCT coefficients, is included in the enhancement layers.

Note that all these forms of layered coding are supported in recent standardization efforts. For example, MPEG-4 supports temporal and spatial scalability in addition to data partitioning. H.263+ supports temporal, SNR, and spatial scalability in annex O, and H.263++ supports data partitioning in annex V.

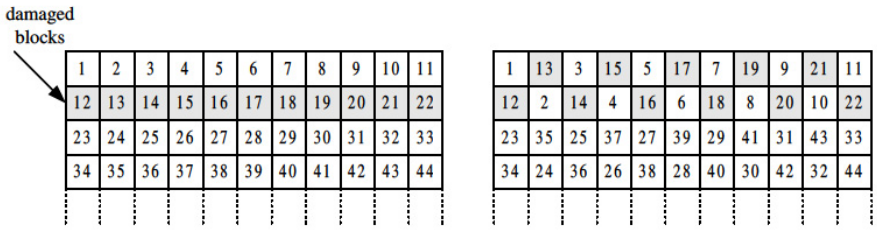
9.6.5 Multiple Description Coding

This technique assumes that there are multiple channels between the encoder and the decoder. These multiple channels can be physically distinct paths or they can be a single path divided into multiple virtual channels using, for example, time or frequency division. The technique further assumes that the error events of these multiple channels are independent. This means that the probability that all channels simultaneously experience errors is very small.

Similar to layered coding, multiple description coding encodes video into multiple streams known as *descriptions*. In this case, however, the descriptions are correlated and have equal importance. The requirement that all descriptions have equal importance means that the descriptions must share some fundamental information about the input video. As a consequence of this information sharing, the descriptions are correlated.

At the encoder, each description is transmitted on a different channel. As already mentioned, the error events of the channels are independent. As a result, at least one description will be received at the decoder without errors. This description carries some fundamental information about the transmitted video and can, therefore, be used to provide a basic level of quality. Since the descriptions are correlated, missing descriptions can be estimated from correctly received descriptions and the quality can be improved.

There are a number of methods to achieve the required decomposition into descriptions. For example, in Ref. 188, the input signal is decomposed and encoded into two streams. The two streams are obtained by transmitting two quantization indices for each quantized level. The index assignment is designed such that when both indices are received, the reconstruction quality is equivalent to that of a fine quantizer. When, however, only one index is received, the reconstruction quality is equivalent to that of a coarse quantizer.



(a) without interleaving (b) with interleaving

Figure 9.6: Coding and transmission order with and without interleaving

There are also other multiple description techniques, as detailed in Refs. 179 and 180.

9.6.6 Interleaved Coding

In normal coding, the blocks of a given frame are encoded in raster scan order, as illustrated in Figure 9.6(a). In this case, when an error occurs in one block, spatial error propagation results in the loss of a contiguous set of blocks. In the example shown, an error in block 12 results in the loss of all blocks to its right.[2] As is discussed later, the concealment of a damaged block depends heavily on the availability of its four neighboring blocks. In this case, a damaged block will have only its top and bottom neighbors intact.

Interleaved coding attempts to separate the information of neighboring blocks as far as possible. As a result, an error in a block will propagate to nonadjacent blocks. Figure 9.6(b) shows the even/odd interleaving scheme adopted in Ref. 189. The numbers here indicate the encoding and transmission order. Thus, the first block in the first row (block 1) is encoded and transmitted first, followed by the second block in the second row (block 2), and so on. Note that in this case, when an error occurs in block 12, the lost set of blocks is not contiguous. Thus, a damaged block will have all its four neighbors intact and this will help the error concealment process considerably.

9.7 Postprocessing (or Concealment) Techniques

The second category of error-resilience techniques are postprocessing (or concealment) techniques. In postprocessing techniques, the *decoder* plays the

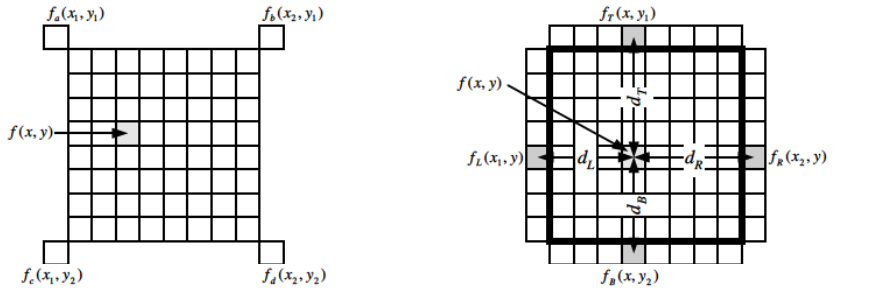
[2]This example assumes that resynchronization codewords are inserted at the beginning of each row of blocks.

primary role. Thus, the decoder attempts to *conceal* the effects of errors by providing a *subjectively* acceptable approximation to the original data. This is achieved by exploiting the limitations of the human visual system and the high temporal and/or spatial correlation of video sequences. Error concealment is an ill-posed problem since it does not have a unique solution. Thus, error concealment techniques exploit *a priori* knowledge of the characteristics of video signals to restrict the otherwise large number of possible solutions. Depending on the information used for concealment, postprocessing techniques can be divided into three main categories: *spatial* techniques, *temporal* techniques, and *hybrid* techniques.

9.7.1 Spatial Error Concealment

Spatial techniques exploit the high spatial correlation of video signals and conceal damaged pels in a frame using information from correctly received and/or previously concealed neighboring pels within the same frame. Such techniques apply primarily to intracoded blocks but may also be used to conceal intercoded blocks with missing motion information or to recover the DFD signal.

In Ref. 190 a damaged pel within a block is interpolated from the four corner pels outside the block, as illustrated in Figure 9.7(a). Interpolation from the four nearest pels outside the block boundaries, as illustrated in Figure 9.7(b), is proposed in Ref. 191. Interpolation in the frequency domain has also been used. For example, in Ref. 192 the DC coefficient of a damaged block is recovered as the average or the median of the DC



$$f(x,y) = (1-x_n)(1-y_n)f_a(x_1,y_1) + x_n(1-y_n)f_b(x_2,y_1) + (1-x_n)y_n f_c(x_1,y_2) + x_n y_n f_d(x_2,y_2)$$

where $x_n = \frac{x-x_1}{x_2-x_1}$, $y_n = \frac{y-y_1}{y_2-y_1}$

(a) using corner pels

$$f(x,y) = \frac{1}{d_L + d_R + d_T + d_B} \times [d_L f_R(x_2,y) + d_R f_L(x_1,y) + d_T f_B(x,y_2) + d_B f_T(x,y_1)]$$

(b) using nearest neighboring pels

Figure 9.7: Error concealment using spatial interpolation

coefficients of the four or eight neighboring blocks. Another approach is to form a partial DC value at each boundary by taking the average of a one-, two-, or four-pels-wide neighborhood. The recovered DC coefficient is then the average or the median of the four partial DC values.

In Ref. 193 the lost DCT coefficients of an intracoded block are recovered by minimizing the intersample variation within the block and across the block boundaries. This is based on the smoothness property of image and video sequences. In Ref. 189 the same method is extended by adding a temporal smoothness measure.

Another property that is used in error concealment is edge continuity. Thus, if the direction of an edge in a neighboring block indicates that the edge passes through the damaged block, then the concealment process must conserve the continuity of this edge. For example, in Ref. 194 an edge classifier is applied to the neighboring blocks to determine which directions characterize the strongest edges passing through the damaged block. For each of these classified directions, directional spatial interpolation along the respective direction is used to create a block from the neighboring pels. The blocks are then mixed together in such a way that all the strong edge features are preserved and combined in a single block used for concealment.

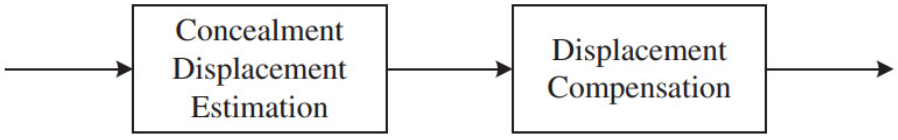
Statistical correlation is another *a priori* assumption utilized in error concealment. For example, in Ref. 195 the pel values of a frame are modeled as a Markov random field (MRF). Maximum *a posteriori* probability (MAP) estimation is then used to spatially interpolate the damaged blocks.

9.7.2 Temporal Error Concealment

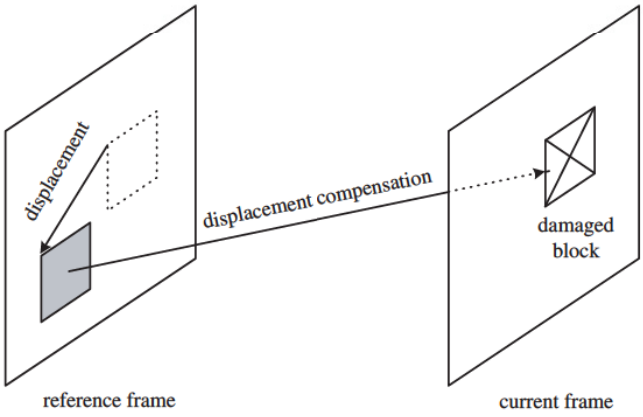
Temporal techniques exploit the high temporal correlation of video signals and conceal damaged pels in a frame using information from correctly received and/or previously concealed pels within a reference frame. Such techniques apply primarily to intercoded blocks. They may work for some intracoded blocks but will completely fail in cases like scene changes and uncovered background.

As in motion-compensated prediction, the process of temporal concealment involves two stages: *concealment displacement estimation* and *displacement compensation*, as shown in Figure 9.8(a). For this reason, temporal concealment is sometimes referred to as *motion-compensated concealment*.

Conventional temporal techniques estimate one concealment displacement for the whole damaged block and then use translational displacement compensation to conceal the block, as shown in Figure 9.8(b). Such techniques perform very well when the original motion vector of the damaged block is available. In this case the first stage of the temporal concealment process,



(a) Stages of temporal concealment



(b) Conventional temporal concealment

Figure 9.8: Temporal error concealment

i.e., displacement estimation, is bypassed and the concealment displacement is simply set to the original motion vector.

In practice, however, the motion vector of a damaged block is usually lost or erroneously received. This is due mainly to spatial error propagation. For example, an erroneous codeword will usually lead to loss of synchronization at the decoder and all blocks, including their motion information, up to the next synchronization point will be undecodable and completely lost.[3] In such cases, the displacement estimation stage at the decoder is extremely important. In fact, the only difference between the various conventional temporal techniques reported in the literature is in their displacement estimation algorithm. This stage is also known as *motion information recovery*, because it attempts to recover or provide an approximation to the original motion information.

The simplest and most commonly used technique is to replace the damaged motion vector with $(0,0)$ [179, 192]. This is based on the center-biased property of video block-motion fields, which is also equivalent to the temporal smoothness property of video signals. The technique is usually referred to as

[3] As already discussed, RVLCs and data partitioning into motion and texture data are some of the mechanisms that can be used to reduce this effect.

temporal replacement (TR) because it effectively replaces the damaged block by its corresponding block in the reference frame. This method works well for stationary and quasi-stationary areas, e.g., background, but will fail for fast-moving areas.

Another technique is to exploit the high-correlation property of video block-motion fields and replace the damaged motion vector with the *average* (AV) [179, 190, 189, 191, 192] or the median [179, 192] of neighboring vectors. This technique works well for areas with smooth motion but will fail for areas with unsmooth motion, e.g., at the boundaries of objects moving in different directions.

A boundary matching (BM) technique has also been used to select a suitable replacement from a set of candidate motion vectors [196, 197, 198]. Assume that a set of M neighboring motion vectors $\mathscr{V}=\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$ is to be used for the concealment of a damaged block $\mathscr{D}$ of size $N \times N$ with its top-left corner at (x_o, y_o). Each candidate vector $\mathbf{v}_i = (v_i^x, v_i^y)$ in $\mathscr{V}$ is used to conceal the damaged block $\mathscr{D}$. The quality of this concealment is assessed using the continuity across the concealed block boundaries. This continuity is measured using the *side-match distortion* (SMD) measure, defined as

$$\mathrm{SMD}_i = \mathrm{SMD}_i^L + \mathrm{SMD}_i^R + \mathrm{SMD}_i^T + \mathrm{SMD}_i^B, \tag{9.1}$$

where SMD_i^L is the sum of absolute, or squared, differences across the *left* boundary of block $\mathscr{D}$ when concealed using candidate vector $\mathbf{v}_i$. Thus

$$\mathrm{SMD}_i^L = \sum_{k=0}^{N-1} g[f_t(x_o - 1, y_o + k) - f_{t-\Delta t}(x_o + v_i^x, y_o + v_i^y + k)], \tag{9.2}$$

where f_t and $f_{t-\Delta t}$ are the current and reference frames, respectively, $g = (\cdot)^2$ for the SSD, and $g = |\cdot|$ for the SAD. Similarly, SMD_i^R, SMD_i^T and SMD_i^B are the side-match distortions across the *right*, *top*, and *bottom* boundaries, respectively. Based on the smoothness property of video signals, the candidate motion vector that achieves the minimum SMD is chosen as the recovered motion vector. Thus

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}_i \in \mathscr{V}} \mathrm{SMD}_i. \tag{9.3}$$

The main advantage of this method is that displacement estimation is based on a distortion measure. The method will fail for areas with unsmooth motion and also for areas with low spatial correlation, e.g., at the boundaries of objects.

Similar to spatial concealment, Bayesian statistical approaches have also been used for motion vector recovery, e.g., Ref. 195.

9.7.3 Hybrid Error Concealment

Hybrid techniques exploit both spatial and temporal correlations of video signals. A straightforward technique is to use spatial concealment for intracoded blocks and temporal concealment for intercoded blocks. More sophisticated combinations are also possible. For example, in Ref. 199 temporal concealment is first used to get an initial estimate of the damaged block. This initial estimate is then refined using spatial concealment.

9.7.4 Coding-Mode Recovery

As already discussed, each of the preceding concealment techniques applies to a particular type of macroblocks. More specifically, spatial concealment is more applicable to intracoded blocks, whereas temporal concealment is more suitable for intercoded blocks. Provided that the coding mode of a damaged block is known, the appropriate type of concealment is applied. In many cases, however, the coding-mode information of a damaged block is also damaged. Thus, coding-mode information needs to be recovered first before being able to choose the appropriate concealment method.

In Ref. 189, when the coding mode is damaged it is simply set to INTRA and the corresponding block is concealed using spatial techniques.

Usually, there is a high correlation between the coding modes of adjacent blocks. Thus, the coding mode of a damaged block can be estimated from the coding modes of neighboring blocks. In Ref. 200, the coding mode of a damaged MB in an MPEG-2 coded video is estimated from the coding modes of its top and bottom neighboring MBs. For example, the coding mode of a damaged MB in a P-frame is set to INTRA only if its top and bottom neighboring MBs are both INTRA coded; otherwise, a FORWARD INTER mode is assumed.

9.8 Interactive Techniques

The third type of error-resilience methods are interactive techniques. In this case, the *encoder* and *decoder* cooperate to minimize the effects of transmission errors. In such techniques, the decoder uses a feedback channel to inform the encoder about which parts of the transmitted video have been received in error. Based on this feedback information, the encoder adjusts its operation to combat the effects of such errors. The following subsections discuss some examples of interactive (or feedback-based) techniques. A more comprehensive review of such techniques can be found in Ref. 201.

9.8.1 Automatic Repeat Request (ARQ)

In this technique, when an error is detected, the decoder automatically requests the encoder to retransmit the damaged data. When this ARQ is received, the encoder retransmits the requested data. Usually, this retransmission is repeated until either the requested data is correctly received or a predetermined number of retransmissions is exceeded.

Typically, when a decoder sends an ARQ, it waits for the arrival of the requested data before resuming normal operation. This introduces delays that may not be acceptable in real-time applications like mobile video communication. To overcome such delays, Wang and Zhu [179] proposed a technique called *retransmission without waiting*. In this technique, instead of waiting for the arrival of the requested data, the damaged video part is concealed and normal decoding operation is then resumed. A trace of the affected pels and their associated coding information is recorded until the arrival of the requested data. This error trace, along with the received data, is then used to correct the affected pels. Another technique proposed in Ref. 179 is the *multicopy retransmission*. In this technique, multiple copies of the damaged data are sent in each single retransmission trial. This reduces the required number of retransmissions and, consequently, reduces delays.

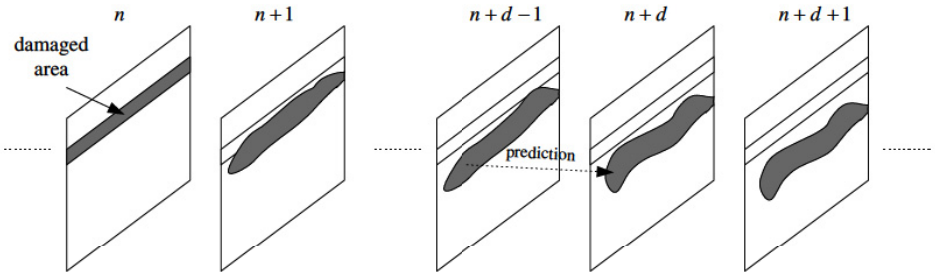
9.8.2 Error Tracking

When feedback information is received, the encoder can reconstruct the error propagation process. In other words, the encoder can *track* the error propagation from the original occurrence up to the current frame. A number of techniques can then be used to utilize this error trace, as discussed next.

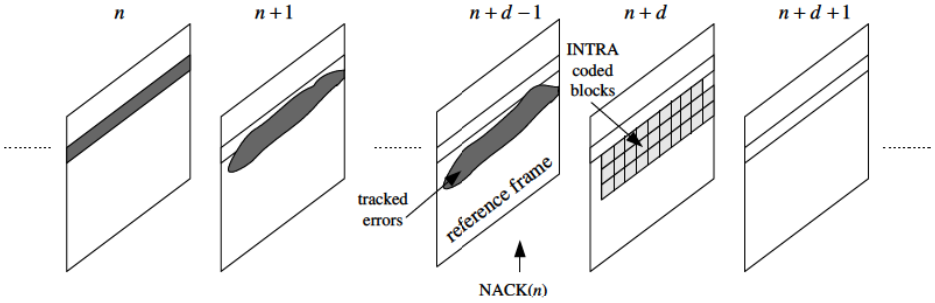
9.8.2.1 INTRA Refresh Based on Feedback

Based on the error trace, areas in the current frame that would have been predicted from affected pels in the reference frame are INTRA encoded. This is illustrated in Figure 9.9. Figure 9.9(a) shows the spatial and temporal propagation in a sequence of frames due to an error in frame n. In Figure 9.9(b) a feedback message arrives at the encoder before the time to encode frame $n + d$. The encoder tracks this error and the affected pels from frame n up to frame $n + d - 1$. During the encoding process of the current frame, $n + d$, blocks that would have been predicted from affected pels in the reference frame, $n + d - 1$, are encoded in INTRA mode to stop error propagation to the next frame, $n + d + 1$.

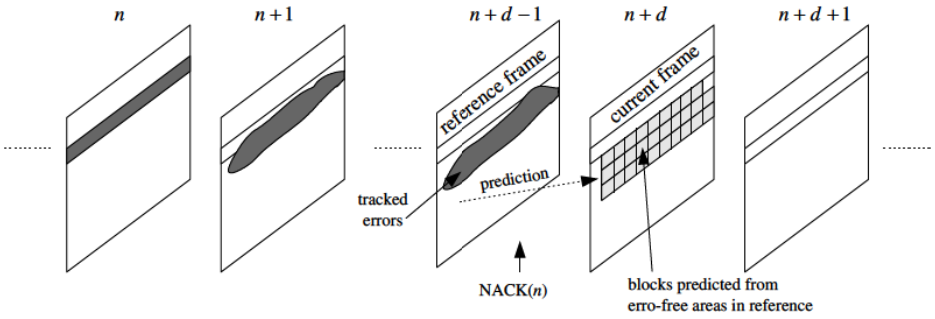
There are two main drawbacks to this approach. First, a perfect reconstruction of error propagation is a computationally complex process. Second, in cases of high error rates, INTRA refresh can result in a significant loss in



(a) Error propagation in a sequence of frames



(b) INTRA refresh based on error tracking and feedback information



(c) Restricted prediction based on error tracking and feedback information

Figure 9.9: Error tracking techniques

coding efficiency. In Ref. 202 Steinbach *et al.* propose a reduced-complexity error-tracking algorithm that can provide a sufficiently accurate estimate of the true error propagation. In order to reduce the loss of coding efficiency, they INTRA refresh only severely affected blocks. Thus, if the process of error

concealment is successful and the error of a given block is sufficiently small, then the encoder may decide against INTRA encoding. Note that this method requires the encoder to perform the same error concealment process that was used at the decoder.

9.8.2.2 Restricted Prediction Based on Feedback

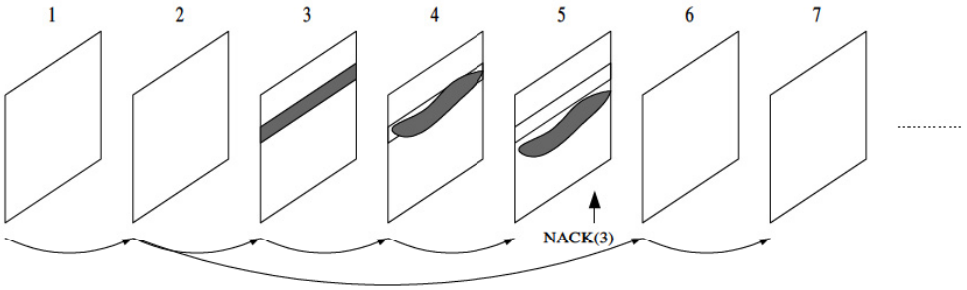
Based on the error trace, prediction of the current frame is restricted to use only error-free areas in the reference frame. For example, in Figure 9.9(c) the affected pels in the reference frame, $n+d-1$, are not used for predicting the current frame, $n+d$. This stops error propagation to the next frame, $n+d+1$. This restricted prediction based on feedback and error tracking was proposed by Wada in the *selective recovery* technique [203].

Again, this technique can also benefit from the reduced-complexity error-tracking algorithm of Steinbach *et al.* [202], and the coding efficiency can also be improved by performing error concealment in the encoder so that both encoder and decoder use the same reference frames for prediction.

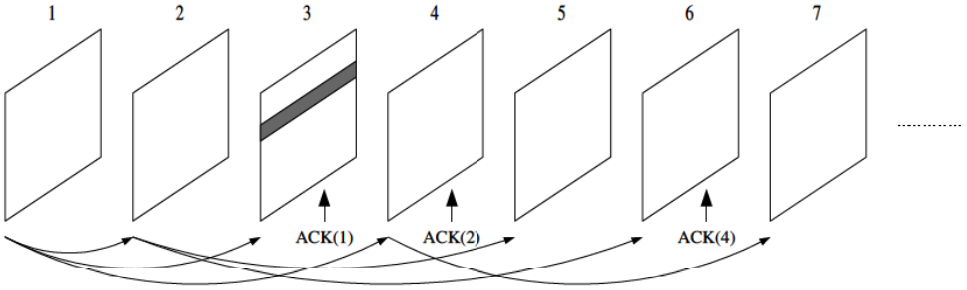
9.8.3 Reference Picture Selection

In reference picture selection (RPS), both the encoder and decoder store multiple previous frames to be used as reference frames. When the encoder learns, through feedback messages from the decoder, that the most recent reference frame contains errors, the encoder switches to use another older reference frame that is known to be error free. Provided the alternative reference frame is not too far away from the current frame, the loss in coding efficiency is not significant. In particular, this technique is more efficient than the INTRA refresh technique. The RPS technique has been adopted by H.263+ in annex N, and an enhanced version of the technique has been included in annex U of H.263++.

Figure 9.10 shows the RPS technique with two types of feedback messages. In the negative acknowledgment mode, illustrated in Figure 9.10(a), the decoder sends a *negative acknowledgment* (NACK) message whenever errors are detected in a frame. In the example shown, the decoder detects an error in frame 3 and sends a NACK(3) message to the encoder. At the encoder, the encoding operation proceeds in the normal way (i.e., using the most recent reference frame for prediction) until the NACK(3) message arrives before encoding frame 6. Based on this message, the encoder knows that errors occurred in frame 3 and propagated up to the most recent reference frame 5. To stop this error propagation, the encoder uses the older error-free reference frame 2 instead of the most recent reference frame 5 to encode the current frame 6.



(a) Reference picture selection with negative acknowledgment messages



(b) Reference picture selection with positive acknowledgment messages

Figure 9.10: Reference picture selection based on feedback

In the positive acknowledgment mode, illustrated in Figure 9.10(b), the decoder sends an *acknowledgment* (ACK) message whenever a frame is received error-free. At the encoder, only acknowledged frames are used as references. In the example shown, the encoder continues to use frame 1 for prediction until it receives the acknowledgment for frame 2. The encoder then starts using the acknowledged frame 2 for prediction until the acknowledgment of the next error-free reference frame is received. Note that since the erroneous frame 3 is not acknowledged, it is never used for prediction and its errors do not propagate to subsequent frames.

Note that during error-free transmission, the NACK mode is more efficient than the ACK mode since the most recent reference frame is used for prediction. During erroneous transmission, however, the NACK mode results in longer periods of error propagation than the ACK mode. Thus, the NACK mode is more suitable if errors occur only rarely after long periods of error-free transmission, whereas the ACK mode is preferred for highly error-prone transmissions.

9.9 Discussion

When transmitted over a mobile channel, compressed video can suffer severe degradation. Thus, error resilience is one of the main requirements for mobile video communication.

Due to the use of predictive and VLC coding, transmission (both random and erasure) errors cause temporal and spatial error propagation in compressed video.

Before being able to combat these effects, it is first necessary to detect whether and where errors have occurred. Different techniques can be used to achieve this error detection.

Error control techniques can be broadly classified into three categories: forward, postprocessing, and interactive techniques. In forward techniques, the encoder plays the primary role. Such techniques work by adding a controlled amount of redundancy to the video bitstream. In postprocessing techniques, the decoder plays the primary role. Thus, the decoder attempts to conceal the effects of errors by providing a subjectively acceptable approximation to the original data. This is achieved by exploiting the limitations of the human visual system and the high temporal and/or spatial correlation of video sequences. In interactive techniques, the encoder and decoder cooperate to minimize the effects of transmission errors. In such techniques, the decoder uses a feedback channel to inform the encoder about which parts of the transmitted video have been received in error. Based on this feedback information, the encoder adjusts its operation to combat the effects of such errors.

It should be emphasized that the three categories of techniques are not mutually exclusive, and different combinations can be employed in practical systems.

Chapter 10

Error Concealment Using Motion Field Interpolation

10.1 Overview

Chapter 9 discussed three categories of error-resilience techniques: forward, postprocessing (or concealment), and interactive techniques. Almost all forward techniques increase the bit rate because they work by adding redundancy to the data, e.g., FEC. Some of them may also require modifications to the encoder, e.g., layered coding, and others may not be suitable for some applications, e.g., multiple description coding assumes several parallel channels between transmitter and receiver. Most interactive techniques depend on a feedback channel between the encoder and decoder. Such a channel may not be available in some applications, e.g., multipoint broadcasting. Most interactive techniques will also introduce some delay and may, therefore, be unsuitable for real-time applications like mobile video communication. On the other hand, concealment techniques do not increase the bit rate, do not require any modifications to the encoder, do not introduce any delay, and can be applied in almost any application. This makes them a very attractive choice for mobile video communication, where bit rate and delay are very critical issues.

A very successful class of error concealment is *temporal error concealment*. Conventional temporal concealment techniques estimate one concealment displacement for the whole damaged block and then use translational displacement compensation to conceal the block from a reference frame. The main problem with such techniques is that incorrect estimation of the concealment displacement can lead to poor concealment of the whole or most of the block.

This chapter describes the design of two novel temporal concealment techniques. In the first technique, *motion field interpolation* (MFI) is used to estimate one concealment displacement per pel of the damaged block. Each

pel is then concealed individually. In this case, incorrect estimation of a concealment displacement will affect only the corresponding pel. On a block level, this may affect few pels rather than the entire block. In the second technique, *multihypothesis motion compensation* (MHMC) is used to combine the first technique with a *boundary matching* (BM) temporal concealment technique to obtain a more robust performance.

The chapter also investigates the performance of different temporal error concealment techniques when incorporated within a multiple-reference video codec. In particular, the chapter finds a combination of techniques that best recovers the spatial-temporal components of a damaged multiple-reference motion vector. In addition, the chapter describes the design of a novel multihypothesis temporal concealment technique that can be used with multiple-reference systems.

The rest of the chapter is organized as follows. Section 10.2 describes the MFI temporal concealment technique, whereas Section 10.3 presents the combined BM-MFI technique. Section 10.4 presents some simulation results. Section 10.5 investigates the performance of temporal error concealment within multiple-reference video codecs. It also describes the multihypothesis multiple-reference temporal concealment technique. The chapter concludes with a discussion in Section 10.6.

Preliminary results of this chapter have appeared in Refs. 204, 205, 206, 207, and 208.

10.2 Temporal Error Concealment Using Motion Field Interpolation (MFI)

10.2.1 Motivation

As described earlier, conventional temporal concealment techniques estimate one concealment displacement for the whole damaged block and then use translational displacement compensation to conceal the block from a reference frame. As already discussed in Section 9.7.2, there are many cases where conventional temporal concealment techniques can fail and the concealment displacement can be incorrectly estimated. The main problem with such techniques is that incorrect estimation of the concealment displacement can lead to poor concealment of the entire or most of the block. This section describes a new temporal error concealment technique. This technique estimates one concealment displacement per pel of the damaged block and then conceals each pel individually. In this case, incorrect estimation of a concealment displacement will affect only the corresponding pel. On a block level, this may affect few pels rather than the entire block.

The described technique uses *motion field interpolation* (MFI) in its displacement estimation stage. In MFI, motion information needs to be available only at a number of nodal or control points within the motion field. The motion vector at any other point within the field can be approximated by interpolating the motion vectors of the surrounding control points. Thus, motion information recovery is inherent in MFI. As discussed in Chapter 5, MFI is used in warping-based motion compensation. Its main advantage over conventional translational compensation is that it provides a smoothly varying motion field that reduces blocking artefacts and compensates for more types of motion. These two features, i.e., *inherent motion information recovery* and *better motion compensation*, can improve both stages of the temporal concealment process, i.e., estimation and compensation, respectively. This makes MFI a very attractive choice for temporal error concealment.

10.2.2 Description of the Technique

Let $f_t(x, y)$ be the value of the current frame at pel location (x, y) and $f_{t-\Delta t}$ be a previously reconstructed and concealed frame. Further, let $\mathscr{D} = \{f_t(x, y): x \in [x_l, x_h], y \in [y_l, y_h]\}$ be a damaged block within the current frame and $\mathbf{v}_L$, $\mathbf{v}_R$, $\mathbf{v}_T$, and $\mathbf{v}_B$ be the motion vectors of the blocks to the left of, to the right of, above, and below the damaged block, respectively. The concealment displacement, $\hat{\mathbf{v}}(x, y) = (\hat{v}_x(x, y), \hat{v}_y(x, y))$, at any pel (x, y) within the damaged block $\mathscr{D}$ can be estimated by interpolating the neighboring motion vectors as follows:

$$\hat{\mathbf{v}}(x, y) = \frac{h_\gamma(x_n)\mathbf{v}_L + (1 - h_\gamma(x_n))\mathbf{v}_R + h_\gamma(y_n)\mathbf{v}_T + (1 - h_\gamma(y_n))\mathbf{v}_B}{2}, \tag{10.1}$$

$$y_n = \frac{y - y_l}{y_h - y_l}, \qquad x_n = \frac{x - x_l}{x_h - x_l}, \tag{10.2}$$

where (x_n, y_n) are the normalized spatial coordinates of pel (x, y) within the damaged block, ranging from $(0, 0)$ at the top-left corner to $(1, 1)$ at the bottom-right corner, and $h_\gamma(\cdot)$ is a suitable *interpolation kernel*.

Thus, the estimated displacement is a weighted sum of the neighboring motion vectors. The interpolation kernel, $h_\gamma(\cdot)$, is used to adjust the weights according to the spatial location of the pel within the damaged block. Intuitively, a pel on the left border should have a high contribution from the left vector, $\mathbf{v}_L$, and a low contribution from the right vector, $\mathbf{v}_R$, and so on. To achieve this, the following interpolation kernel [209] was used:

$$h_\gamma(a) = \frac{k(\gamma(2a - 1)) - k(\gamma)}{k(-\gamma) - k(\gamma)}, \quad 0 \le a \le 1 \text{ and } \gamma \ge 1, \tag{10.3}$$

where

$$k(b) = \frac{1}{1 + e^{b}}. \tag{10.4}$$

The parameter γ in Equation (10.3) is used to control the smoothness of the interpolation kernel. As γ varies from 1 to ∞, the interpolation kernel varies from an approximately linear shape to a brickwall shape, as illustrated in Figure 10.1.

Once the concealment displacement is estimated, then the damaged pel is concealed as follows:

$$\hat{f}_t(x, y) = f_{t-\Delta t}(x + \hat{v}_x(x, y), y + \hat{v}_y(x, y)). \tag{10.5}$$

In the case where the estimation process produces a subpel accurate displacement, the compensation process will require accessing a pel at a nonsampling location within the reference frame. Interpolation (e.g., bilinear) of the pels at surrounding sampling locations can be employed to provide an approximation to the required pel.

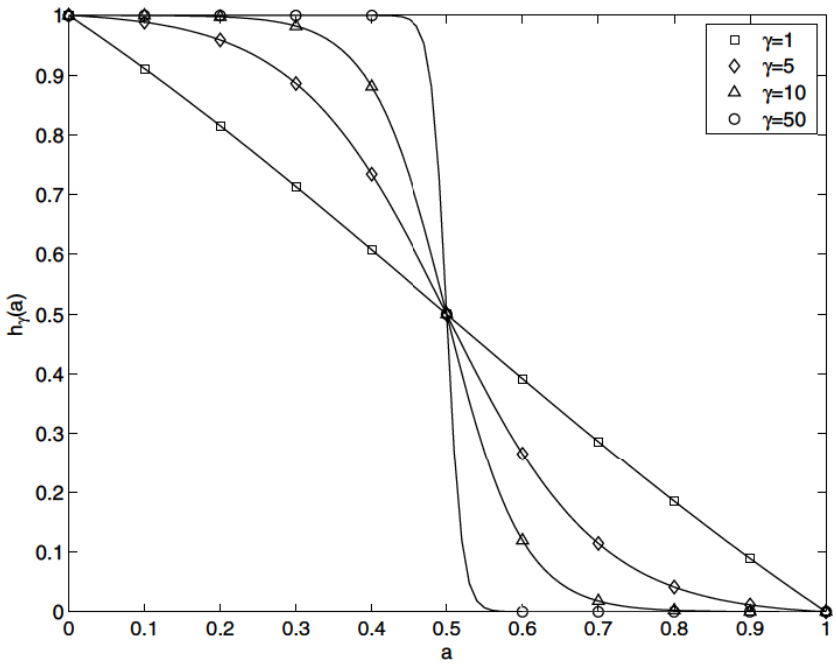


Figure 10.1: Interpolation kernel $h_\gamma(\cdot)$ with different values of the smoothness parameter γ

Table 10.1: Computational complexity of the displacement estimation stage of different temporal concealment techniques with a block of 16×16 pels

	Add/subtract	Multiply/divide	Magnitude
TR	—	—	—
AV	6	2	—
BM	496	—	256
MFI	**516**	**6**	—

10.2.3 Reduced-Complexity MFI

One of the main disadvantages of MFI is its high computational complexity. In the case of a linear interpolation kernel, Equation (10.1) reduces to

$$\hat{\mathbf{v}}(x, y) = \frac{(1 - x_n)\mathbf{v}_L + x_n\mathbf{v}_R + (1 - y_n)\mathbf{v}_T + y_n\mathbf{v}_B}{2}. \tag{10.6}$$

A direct implementation of Equations (10.6) and (10.2) requires $10N^2$ additions/subtractions and $12N^2$ multiplications/divisions for an $N \times N$ block. This complexity can be reduced using a number of methods. One method is to calculate the weights off-line and store them in a lookup table. This reduces the complexity to $6N^2$ additions/subtractions and $8N^2$ multiplications/divisions. Another method is to use a *line-scanning* technique. That is, once $\hat{\mathbf{v}}(x, y)$ is calculated, the displacement of the next pel in the row and the next pel in the column can be calculated as follows:

$$\hat{\mathbf{v}}(x + 1, y) = \hat{\mathbf{v}}(x, y) + \frac{\mathbf{v}_R - \mathbf{v}_L}{2N} \quad \text{and} \quad \hat{\mathbf{v}}(x, y + 1) = \hat{\mathbf{v}}(x, y) + \frac{\mathbf{v}_B - \mathbf{v}_T}{2N}. \tag{10.7}$$

It is very simple to derive Equations (10.7) from Equation (10.6). Note that the second term in both of Equations (10.7) is a constant and needs to be calculated only once per block. This line-scanning technique further reduces the complexity to $(2N^2 + 4)$ additions/subtractions and six multiplications/divisions.

Table 10.1 compares the computational complexity of different temporal concealment techniques for a 16×16 block. The figures in the table refer to the complexity of the displacement estimation stage and do not include the complexity of the displacement compensation stage.[1] The figures for BM are based on four candidate motion vectors and SAD as the SMD measure. They do not include the complexity of sorting the SMDs and choosing the vector with the minimum SMD. Although the MFI technique has the highest number of multiplications/divisions, this increased complexity can be justified by

[1]For MFI, the displacement compensation stage is more complex, since it may involve interpolation.

improved concealment quality, as will be shown later. A point to note here is that MFI will be used only for damaged blocks. Thus, provided that the error rate is relatively low, this will not increase the complexity of the decoder considerably.

10.3 Temporal Error Concealment Using a Combined BM-MFI Technique

10.3.1 Motivation

In this section, *multihypothesis motion compensation* (MHMC) [106] is used to further improve the second stage, i.e., compensation, of the temporal concealment process. In MHMC, a block is compensated using a weighted average of several motion-compensated predictions (hypotheses). This is a general term that can be used to describe techniques like overlapped motion compensation, bidirectional motion compensation, and any other technique that compensates individual pels using more than one motion vector. When applied to temporal error concealment, this means that each pel of the damaged block will be concealed using more than one concealment displacement. In the described technique, *two* concealment displacements are used per pel: one is estimated using BM, as described in Section 9.7.2, and the other is estimated using MFI, as described in Section 10.2. The BM technique was chosen because it is one of the best conventional temporal error concealment techniques. A similar combination between BM and overlapped motion compensation has also been reported in Ref. 198.

In addition to improving the second stage of the temporal concealment process, the combination of BM and MFI can provide a more robust performance. This can be explained as follows. There are many cases where the BM technique will fail but the MFI technique will not, and vice versa. In such cases, a combination may be more robust because it may average out the concealment distortion.

10.3.2 Description of the Technique

Let $\hat{\mathbf{m}}(x, y) = (\hat{m}_x(x, y), \hat{m}_y(x, y))$ be the displacement estimated using MFI to conceal pel (x, y) of the damaged block $\mathscr{D}$, and let $\hat{\mathbf{b}} = (\hat{b}_x, \hat{b}_y)$ be the displacement estimated using BM to conceal the whole block. Then pel (x, y) is concealed as follows:

$$\begin{aligned}\hat{f}_t(x, y) = {} & w_\delta(x_n, y_n) f_{t-\Delta t}(x + \hat{m}_x(x, y), y + \hat{m}_y(x, y)) \\ & + (1 - w_\delta(x_n, y_n)) f_{t-\Delta t}(x + \hat{b}_x, y + \hat{b}_y). \end{aligned} \tag{10.8}$$

Thus, the concealed pel is a weighted sum of two predictions. The function $w_\delta(\cdot,\cdot)$ is used to adjust the weights given to MFI and BM according to the spatial location of the pel within the damaged block. Knowledge of the way both BM and MFI work can provide some insights into designing a suitable $w_\delta(\cdot,\cdot)$. For example, the SMD measure of the BM technique involves the border pels of the damaged block. It is expected, therefore, that BM will perform well at those pels. Therefore, BM must be given high weights at the borders of the block and low weights at the center. To achieve this, the following function was used:

$$w_\delta(x_n, y_n) = \frac{g_\delta(x_n) g_\delta(y_n) + 1}{2}, \tag{10.9}$$

where

$$g_\delta(a) = \begin{cases} 1 - \frac{k(\delta(4a-1)) - k(\delta)}{k(-\delta) - k(\delta)}, & 0 \le a \le \frac{1}{2} \\ g_\delta(1-a), & \frac{1}{2} < a \le 1 \end{cases} \tag{10.10}$$

and $k(\cdot)$ is defined by Equation (10.4). The parameter δ is used to control the smoothness of $w_\delta(\cdot,\cdot)$, as illustrated in Figure 10.2.

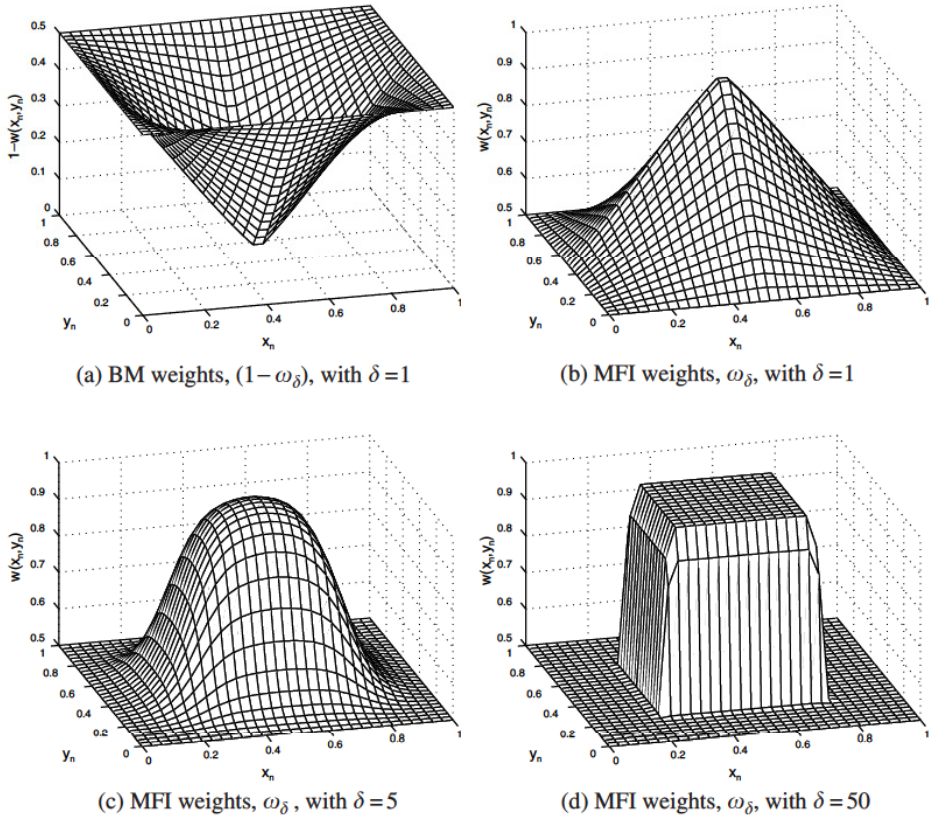
Before proceeding to present simulation results, it is valuable at this point to highlight the main differences between the two novel algorithms, MFI and BM-MFI, and conventional temporal error concealment techniques. These are summarized in Table 10.2.

10.4 Simulation Results

10.4.1 Results Within an Isolated Error Environment

It is very important to evaluate the performance of the techniques in isolation from any external effects, like temporal and spatial error propagation and the choice of the error detection algorithm. This is particularly important for a fair comparison, since such error mechanisms and algorithm choices may randomly affect one technique more than another. Thus, in this set of simulations, the following assumptions were made

1. There is no temporal error propagation. This was achieved by using original reference frames for the concealment process.
2. There is no spatial error propagation. This is equivalent to using fixed-length codes and no predictive coding.
3. The concealment process is supported by an ideal error detection algorithm that can identify all damaged blocks.



(a) BM weights, $(1-\omega_\delta)$, with $\delta=1$

(b) MFI weights, ω_δ, with $\delta=1$

(c) MFI weights, ω_δ , with $\delta=5$

(d) MFI weights, ω_δ, with $\delta=50$

Figure 10.2: Multihypothesis weights $w_\delta(\cdot,\cdot)$ with different values of the smoothness parameter δ

Table 10.2: Comparison between conventional temporal concealment and the MFI and BM-MFI techniques

	Conventional temporal concealment	MFI	BM-MFI
Displacement estimation	One displacement per block using AV, TR, BM, etc.	One displacement per pel using MFI (weighted sum of four neighboring vectors)	Two displacements per pel: one produced by MFI and another produced by BM
Displacement compensation	Translational, same displacement for the whole block.	Translational, but on a pel-by-pel basis	Multihypothesis motion compensation (each pel is a weighted sum of two concealments)

Hereafter, the term *isolated error environment* will be used to refer to this set of test conditions.

All results in this subsection were generated using a full-search block-matching algorithm with blocks of 16×16 pels, a maximum allowed displacement of ± 15 pels, SAD as the distortion measure, restricted motion vectors, and full-pel accuracy. Block losses were introduced randomly. Five temporal error concealment techniques were simulated: temporal replacement (TR), average vector (AV), boundary matching with side-match distortion (BM), motion field interpolation (MFI), and the combination of BM and MFI (BM-MFI). In each technique, the motion vectors of the four neighboring blocks—left, right, above and below—were used in the concealment displacement estimation stage. Whenever a neighboring motion vector was not available, e.g., damaged or does not exist as in border blocks, it was set to $(0, 0)$. For the BM technique, SAD was used in the side-match distortion calculations. Again, to mask any external effects, all quoted PSNRs in this set of simulations were calculated for concealed blocks only and averaged over the whole sequence. All quoted results refer to the luma components of sequences.

10.4.1.1 Choice of Parameters

Before evaluating the performance of MFI and BM-MFI, suitable values for the smoothness parameters γ and δ need to be chosen. Figure 10.3 shows the effect of changing the smoothness parameter γ on the performance of MFI when applied to FOREMAN at 25 frames/s with different block loss rates. In general, the performance is not particularly sensitive to the choice of γ (a change of about 0.3 dB). As γ increases, the performance of MFI deteriorates slightly. The best performance is achieved with $\gamma = 1$. This is approximately a linear kernel. Thus, a linear interpolation kernel will be used in all subsequent simulations. Note that a linear kernel also facilitates the use of a line-scanning technique to reduce complexity, as was shown in Section 10.2.3.

Figure 10.4 shows the effect of changing the smoothness parameter δ on the performance of BM-MFI when applied to FOREMAN at 25 frames/s with different block loss rates. Again, the performance is not very sensitive to changes in δ. As δ increases, the performance of BM-MFI slightly deteriorates. The best performance is achieved with $\delta = 1$. The corresponding multihypothesis weights are those shown in Figures 10.2(a) and 10.2(b). In what follows, this value of δ will be used.

10.4.1.2 Performance Evaluation

Figures 10.5, 10.6, and 10.7 compare the performance of the five techniques when applied to AKIYO, FOREMAN, and TABLE TENNIS, respectively. All results were generated with a frame skip of 1.

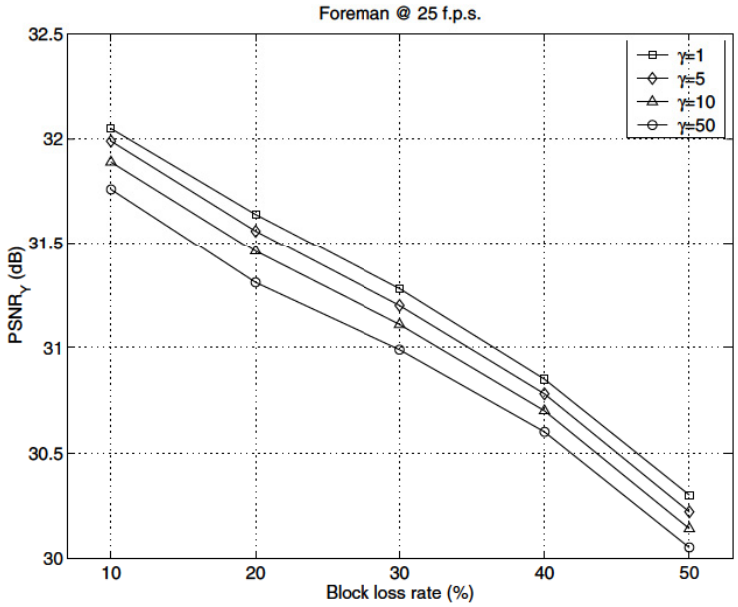


Figure 10.3: Performance of MFI when applied to QSIF FOREMAN at 25 frames/s with different interpolation kernels. PSNRs are for damaged blocks only

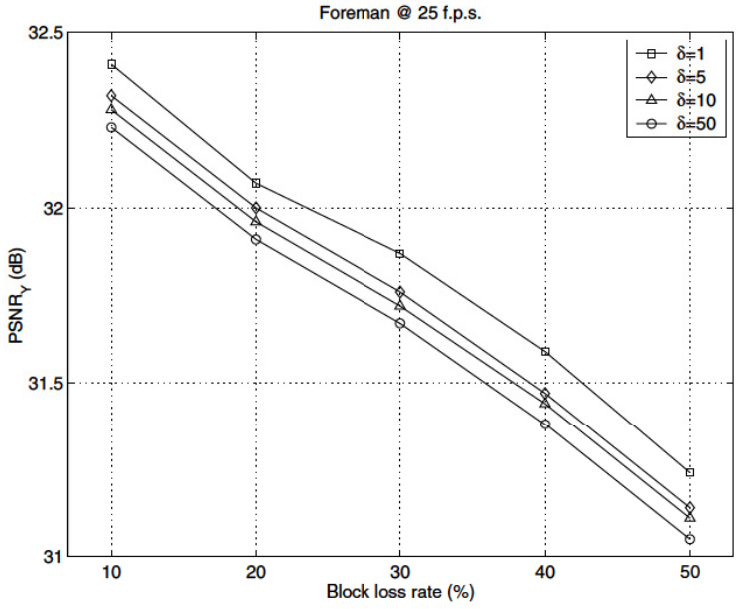


Figure 10.4: Performance of BM-MFI when applied to QSIF FOREMAN at 25 frames/s with different multihypothesis weights. PSNRs are for damaged blocks only

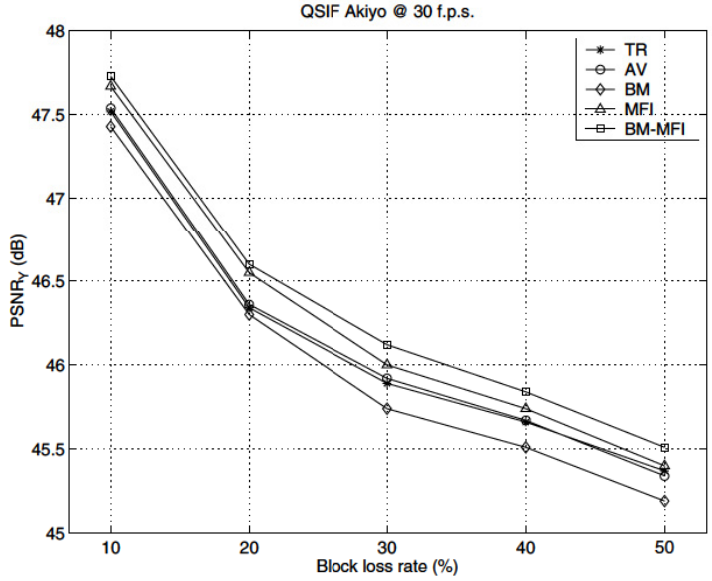


Figure 10.5: Comparison between different temporal concealment techniques when applied to QSIF AKIYO at 30 frames/s. PSNRs are for damaged blocks only

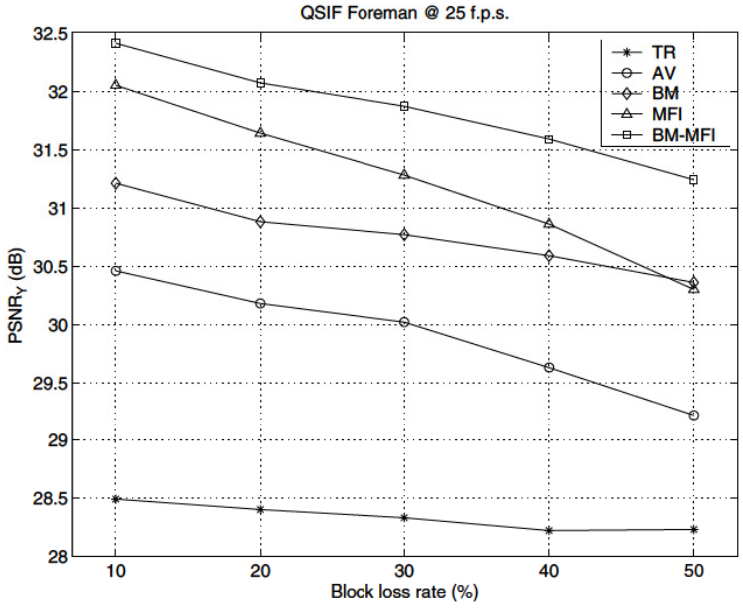


Figure 10.6: Comparison between different temporal concealment techniques when applied to QSIF FOREMAN at 25 frames/s. PSNRs are for damaged blocks only

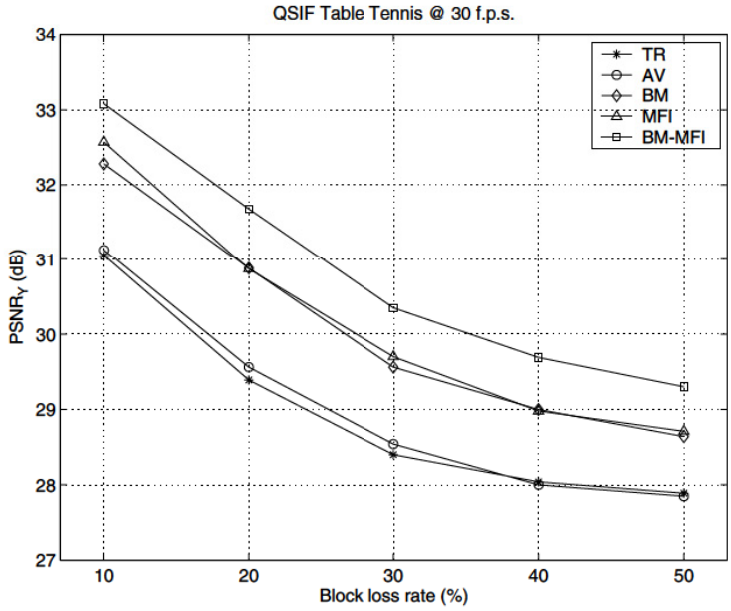
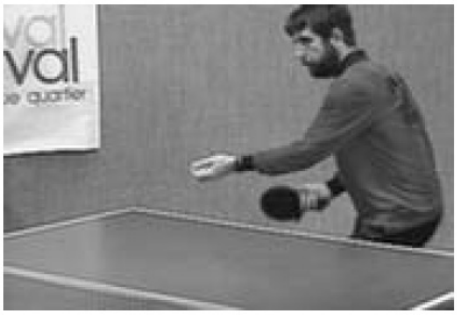


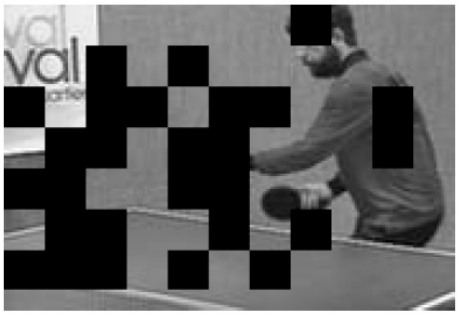
Figure 10.7: Comparison between different temporal concealment techniques when applied to QSIF TABLE TENNIS at 30 frames/s. PSNRs are for damaged blocks only

In general, the best performance was achieved by BM-MFI, followed by MFI, then BM, AV, and TR. As expected, TR performs well for the low-movement AKIYO sequence. The poor performance of BM for AKIYO may be due to an ambiguity problem where neighboring motion vectors give similar SMD measures. A very interesting point to note is that the performance of MFI starts to deteriorate for FOREMAN at high block loss rates. This may be due to the high dependency of MFI on the availability of the neighboring motion vectors. This can be improved using interleaving techniques, as was described in Chapter 9. In all cases, however, the BM-MFI technique maintained its superior performance. This is a clear indication of the robustness of the technique. Over the three sequences and the considered block loss rate range, MFI provides on average 0.3 dB, 0.9 dB, and 1.4 dB improvements over BM, AV, and TR, respectively, whereas BM-MFI provides a further 0.5 dB improvement over MFI. This corresponds to improvements of about 0.8 dB, 1.4 dB, and 1.9 dB over BM, AV, and TR, respectively.

Figure 10.8 shows the subjective quality of the 58th frame of TABLE TENNIS with a block loss rate of 30% when concealed using BM and BM-MFI. The superior performance of the BM-MFI technique is immediately evident from the good concealment of the left hand of the player. Note, however, that some parts of the shirt are less sharp with the BM-MFI technique. This may be due to the low-pass filtering effect of the averaging (weighting) process.



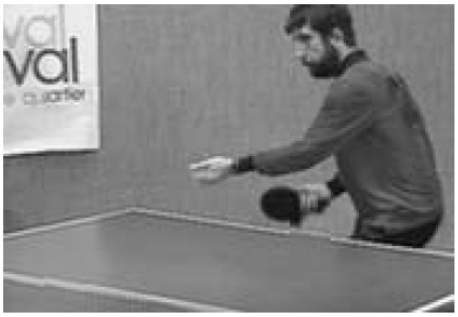
(a) Original 58th frame



(b) Damaged blocks, 30%



(c) Concealed using BM



(d) Concealed using BM-MFI

Figure 10.8: Subjective quality of concealed 58th frame of QSIF TABLE TENNIS at 30 frames/s with a block loss rate of 30%

10.4.2 Results Within an H.263 Decoder

This set of simulations tests the performance of the techniques when incorporated within an H.263 decoder. In this case, the assumptions made in the previous set of simulations will be relaxed. In other words, previously reconstructed, possibly damaged and concealed, frames will be used for both prediction and concealment. This will result in temporal error propagation. In addition, spatial error propagation will also occur, since H.263 uses VLC and predictive coding.

The Telenor implementation [144] of H.263 was used in this simulation. The decoder was modified to perform error detection by detecting syntax and semantic violations, as was described in Section 9.5. When an error is detected, the decoding process is stopped, the decoder searches for the next synchronization codeword, and decoding is resumed. All macroblocks between the point where the error was detected and the synchronization point are marked as damaged macroblocks. In this simulation, the H.263 encoder option to insert synchronization codewords at the start of each GOB was switched on. All

Table 10.3: Comparison between different temporal concealment techniques when applied to three test sequences corrupted with a random bit error rate of 10^{-3}. PSNRs are for whole frames

		Error free	TR	AV	BM	MFI	BM-MFI
AKIYO	$PSNR_{Y'}$	35.15	30.01	29.93	28.19	30.21	30.35
12 kbits/s	$PSNR_{C'_R}$	37.18	34.01	33.67	30.61	34.12	34.23
	$PSNR_{C'_B}$	39.16	36.14	36.03	35.00	36.20	36.32
	PSNR	**35.92**	**31.12**	**31.02**	**29.18**	**31.31**	**31.45**
FOREMAN	$PSNR_{Y'}$	27.93	19.11	19.30	19.59	19.56	20.05
24 kbits/s	$PSNR_{C'_R}$	35.02	30.49	29.01	28.92	30.64	30.67
	$PSNR_{C'_B}$	34.54	29.93	29.37	29.23	30.34	30.40
	PSNR	**29.26**	**20.71**	**20.84**	**21.11**	**21.15**	**21.62**
TABLE TENNIS	$PSNR_{Y'}$	33.21	18.36	18.25	18.58	18.68	18.95
48 kbits/s	$PSNR_{C'_R}$	38.21	23.86	22.50	22.40	23.91	23.93
	$PSNR_{C'_B}$	36.79	21.79	21.32	21.42	22.22	22.34
	PSNR	**34.22**	**19.39**	**19.16**	**19.43**	**19.70**	**19.94**

other optional modes were switched off. No INTRA refresh was employed. Thus, only the first frame was INTRA coded.

The H.263 encoder was used to encode the three sequences AKIYO, FOREMAN, and TABLE TENNIS[2] at bit rates of 12 kbits/s, 24 kbits/s, and 48 kbits/s, respectively. Note that the bit rates were chosen according to the amount of spatial detail and movement within each sequence. Note also that all bit rates were chosen within the very-low-bit-rate range, i.e., less than 64 kbits/s.

The compressed bitstreams were corrupted with random bit errors generated according to the MPEG-4 error robustness test specification [210]. The specifications provide an initial period of 1.5 s during which no errors are injected. This allows for the encoder to transmit an initial INTRA frame and for the codec operation to stabilize into a steady state before errors are introduced.

Table 10.3 summarizes the performance of the five techniques when applied to the three test sequences with a frame skip of 1 and a bit error rate (BER) of 10^{-3}. The quoted PSNRs are for whole frames and averaged over the sequence. The quantities $PSNR_{Y'}$, $PSNR_{C'_R}$, and $PSNR_{C'_B}$ represent the PSNRs of the separate luma and two chroma components, respectively, whereas PSNR

[2]The luma components of both AKIYO and TABLE TENNIS were zero-padded vertically to 128 lines because Telenor's H.263 can work only with an integer multiple of 16. The corresponding chroma components were also appropriately padded.

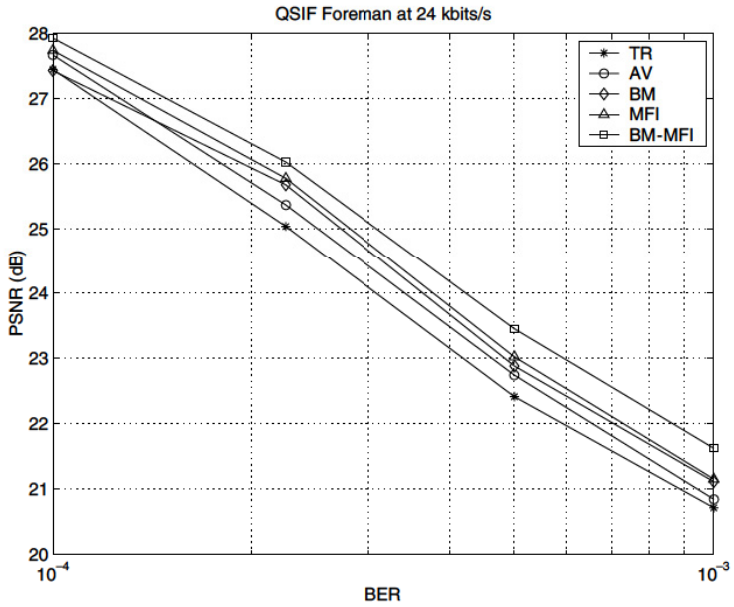


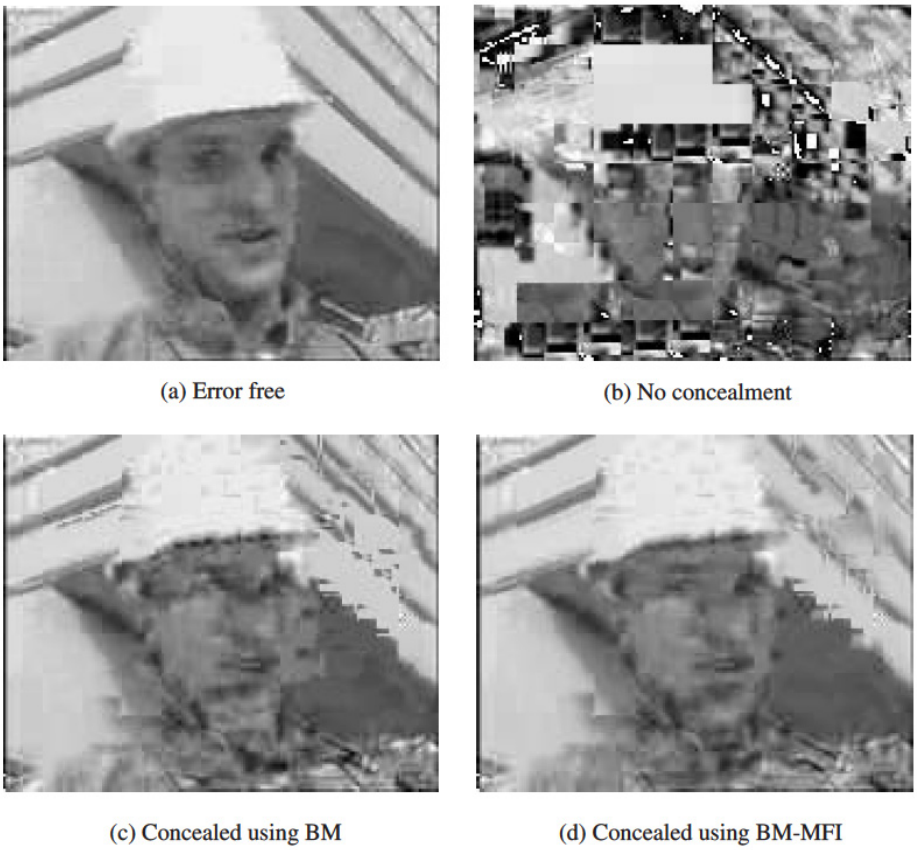
Figure 10.9: Comparison between different temporal concealment techniques when applied to QSIF FOREMAN. The sequence was H.263 encoded at 24 kbits/s and then corrupted with a range of bit error rates

represents the PSNR of the three components together with a 4:2:0 subsampling. Again, the best performance in each case was achieved by BM-MFI, followed by MFI. For example, for the TABLE TENNIS sequence, MFI provides improvements of 0.27 dB, 0.54 dB, and 0.31 dB over BM, AV, and TR, respectively, whereas BM-MFI provides a further 0.24 dB improvement over MFI. This corresponds to improvements of about 0.51 dB, 0.78 dB, and 0.55 dB over BM, AV, and TR, respectively.

Figure 10.9 shows the performance of the five techniques when used to conceal the 24 kbits/s QSIF FOREMAN sequence corrupted with BERs in the range 10^{-4} to 10^{-3}. At low BERs the differences between the techniques are small. However, as the BER increases, the techniques split into three performance levels. The lowest level includes TR and AV, the next level includes BM and MFI, and the highest level includes BM-MFI.

Figure 10.10 shows a frame of the 24 kbits/s QSIF FOREMAN corrupted with a BER of 10^{-3} and then decoded and concealed using BM and BM-MFI. The superior performance of the BM-MFI technique is immediately evident, especially at the eyes and the edges of the face.

It is worth noting here that at a BER of 10^{-3}, the PSNRs of the concealed sequences drop by about 5–9 dB compared to the error-free values, and the subjective quality may not be acceptable. A close inspection of the



(a) Error free (b) No concealment

(c) Concealed using BM (d) Concealed using BM-MFI

Figure 10.10: Subjective quality of decoded and concealed frame of QSIF FOREMAN. The sequence was H.263 encoded at 24 kbits/s and corrupted with a 10^{-3} bit error rate

decoded and concealed sequences revealed that this poor performance is due mainly to the effects of spatial and temporal error propagation and also to the imperfections of the error detection approach. In addition, it was observed that temporal techniques do not perform well for intracoded blocks, scene changes, and uncovered backgrounds. Thus, despite their advantages, temporal error concealment techniques must be combined with spatial error concealment and, more importantly, must be supported by some error containment techniques, such as INTRA refresh.

10.5 Temporal Error Concealment for Multiple-Reference Motion-Compensated Prediction

As already discussed, temporal error concealment is an important tool to combat the effects of errors on transmitted video. A number of temporal error

concealment techniques have been proposed in the literature, and their performances have been extensively studied within typical single-reference video codecs operating over various error-prone channels. There is, however, a *need* to characterize the performance of such techniques within multiple-reference video codecs. This is the main aim of this section.

Temporal error concealment within a multiple-reference video codec can be split into two problems: *spatial-components* (d_x, d_y) *recovery* and *temporal-component* d_t *recovery*. Thus, a multiple-reference temporal error concealment method can be represented by a combination of the form S-T, where S is the technique used to recover the spatial components and T is the technique used to recover the temporal component. In this section, S and T can be chosen from the following list of techniques

ZR The recovered motion component (either spatial or temporal) is set to *zero*. In Chapter 9 this was referred to as temporal replacement (TR).

AV The recovered motion component is set to the *average* of the *corresponding* components of a set of neighboring motion vectors. In this section, four neighboring vectors are used: top, bottom, left, and right.

BM This is a boundary-matching method (refer to Section 9.7.2 for a detailed description). A set of candidate vectors is first chosen. Each candidate is then used to conceal the damaged block. The quality of this concealment is assessed using the side-match distortion (SMD) measure, which is defined as the sum of absolute (or squared) differences across the four boundaries of the block. The candidate with the minimum SMD is chosen. In this section, the set of candidates includes the four neighboring vectors—top, bottom, left, and right—and the SMD is defined as the SAD across the boundaries.

MFI This is the method described in this chapter. It uses motion field interpolation to recover one vector per pel of the damaged block. In this section a linear interpolation kernel is employed.

Since there are four techniques in the list, there are 16 possible combinations of the form S-T. Each combination leads to a different long-term temporal concealment method. For example, assume that $\mathbf{l} = (l_x, l_y, l_t)$, $\mathbf{r} = (r_x, r_y, r_t)$, $\mathbf{t} = (t_x, t_y, t_t)$, and $\mathbf{b} = (b_x, b_y, b_t)$ are, respectively, the motion vectors of the blocks to the left of, to the right of, above, and below the damaged block. A combination of the form AV-BM means that the spatial components (d_x, d_y) are first recovered using the AV method:

$$\hat{d}_x = \frac{l_x + r_x + t_x + b_x}{4} \quad \text{and} \quad \hat{d}_y = \frac{l_y + r_y + t_y + b_y}{4}. \tag{10.11}$$

Then a set $\mathscr{C} = \{\mathbf{d}_1, \ldots, \mathbf{d}_4\}$ of four candidates is formed from the recovered spatial components $(\hat{d}_x, \hat{d}_y)$ and the four temporal components of the neighboring blocks. In other words: $\mathbf{d}_1 = (\hat{d}_x, \hat{d}_y, l_t)$, $\mathbf{d}_2 = (\hat{d}_x, \hat{d}_y, r_t)$, $\mathbf{d}_3 = (\hat{d}_x, \hat{d}_y, t_t)$, and $\mathbf{d}_4 = (\hat{d}_x, \hat{d}_y, b_t)$. The BM technique is then used to recover the temporal component by choosing from this set of candidates. Thus

$$\hat{\mathbf{d}} = \arg \min_{\mathbf{d}_i \in \mathscr{C}} \mathrm{SMD}(\mathbf{d}_i). \tag{10.12}$$

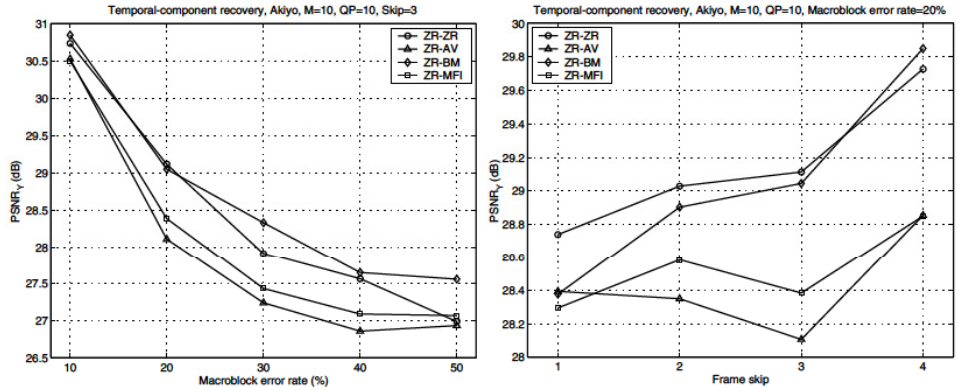
A multiple-reference rate-constrained H.263-like codec was used to generate the results of this section. This codec uses full-pel full-search block matching with macroblocks of 16×16 pels, a maximum allowed spatial displacement of ± 15 pels, SAD as the distortion measure, restricted motion vectors, and reconstructed reference frames. Motion vectors are coded using the median predictor and the VLC table of the H.263 standard. The frame signal (in case of INTRA) and the DFD signal (in case of INTER) are transform encoded according to the H.263 standard. The codec uses rate-constrained motion estimation and mode decision as defined in the high-complexity mode of TMN10. The codec employs a sliding-window control to maintain a long-term memory of size $M = 10$ frames. Only the first frame is INTRA coded, and no INTRA refresh is employed. A fixed quantization parameter of $\mathrm{QP} = 10$ is used. Errors were introduced randomly on a macroblock level. Thus, an error rate of 20% means that 20% of the macroblocks are damaged per frame. It is assumed that the decoder uses an ideal error detection mechanism. All quoted results refer to the luma components of sequences.

10.5.1 Temporal-Component Recovery

This set of experiments investigate the best technique for recovering the temporal component d_t of a damaged long-term motion vector. In this case, the spatial recovery technique S, in the combination S-T, was kept constant at ZR, whereas the temporal recovery technique T was varied over ZR, AV, BM, and MFI. In other words, four S-T combinations were considered: ZR-ZR, ZR-AV, ZR-BM, and ZR-MFI.

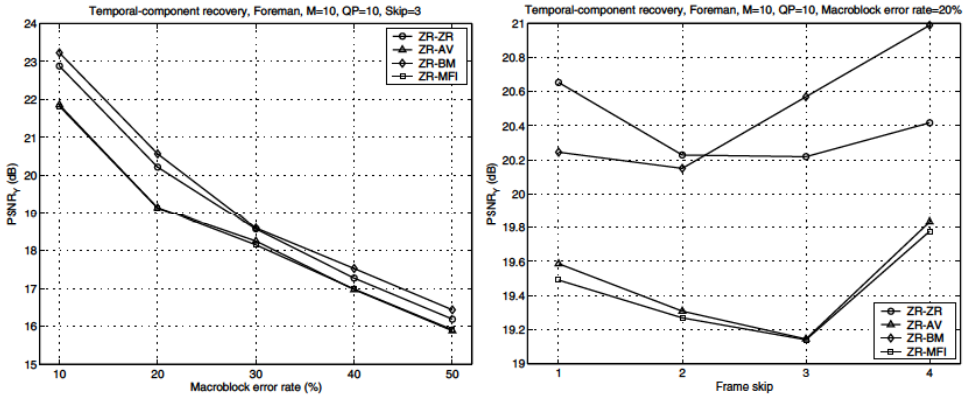
Figures 10.11, 10.12, and 10.13 show the results for the QSIF sequences AKIYO, FOREMAN, and TABLE TENNIS, respectively. Part (a) of each figure shows the performance with a frame skip of 3 over a range of macroblock error rates, whereas part (b) shows the performance with a macroblock error rate of 20% over a range of frame skips.

In general, the best temporal-component recovery is achieved by ZR and BM (i.e., ZR-ZR and ZR-BM). The good performance of ZR is due to the zero-biased distribution of the temporal components (Property 6.3.1.2). In other words, the temporal component $d_t = 0$ has the highest frequency of occurrence



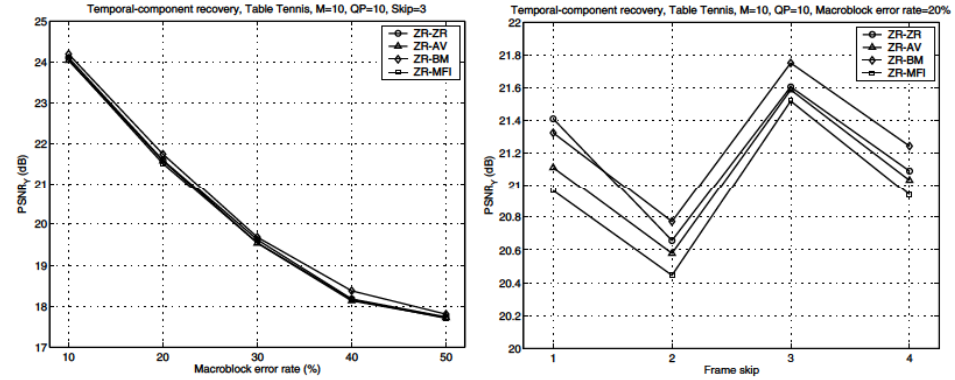
(a) Performance over a range of error rates (b) Performance over a range of frame skips

Figure 10.11: Temporal-component recovery for QSIF AKIYO with $M = 10$ and QP $= 10$



(a) Performance over a range of error rates (b) Performance over a range of frame skips

Figure 10.12: Temporal-component recovery for QSIF FOREMAN with $M = 10$ and QP $= 10$



(a) Performance over a range of error rates (b) Performance over a range of frame skips

Figure 10.13: Temporal-component recovery for QSIF TABLE TENNIS with $M = 10$ and QP $= 10$

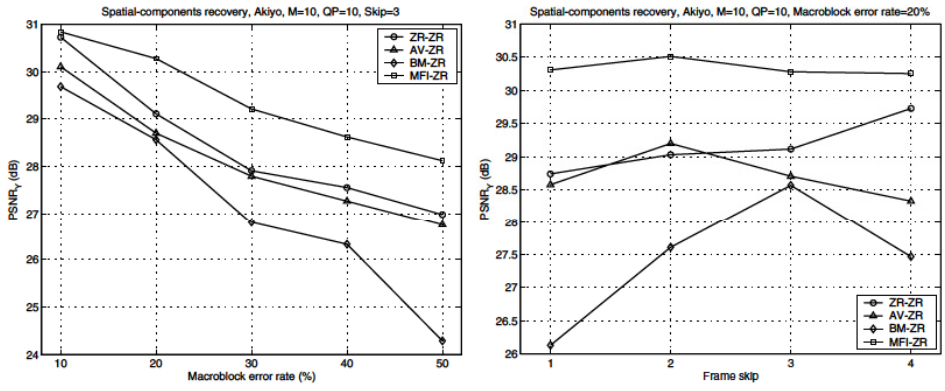
within the long-term memory block-motion field. Note that at low frame skips, this simple ZR method is sufficient, whereas at high frame skips the more complex method of BM has to be employed. This may be due to the fact that at high frame skips, the zero-biased distribution becomes more spread (see Property 6.3.1.2 and Figure 6.3). In other words, $d_t = 0$ becomes less probable, and longer temporal components start to appear more frequently in the motion field. Such components need to be recovered using BM. Both AV and MFI provide poor temporal-component recovery compared to BM and ZR.

10.5.2 Spatial-Components Recovery

This set of experiments investigates the best technique for recovering the spatial components (d_x, d_y) of a damaged long-term motion vector. In this case, the temporal recovery technique T, in the combination S-T, was kept constant at ZR, whereas the spatial recovery technique S was varied over ZR, AV, BM, and MFI. In other words, four S-T combinations were considered: ZR-ZR, AV-ZR, BM-ZR, and MFI-ZR.

Figures 10.14, 10.15, and 10.16 show the results for the QSIF sequences AKIYO, FOREMAN, and TABLE TENNIS, respectively. Part (a) of each figure shows the performance with a frame skip of 3 over a range of macroblock error rates, whereas part (b) shows the performance with a macroblock error rate of 20% over a range of frame skips.

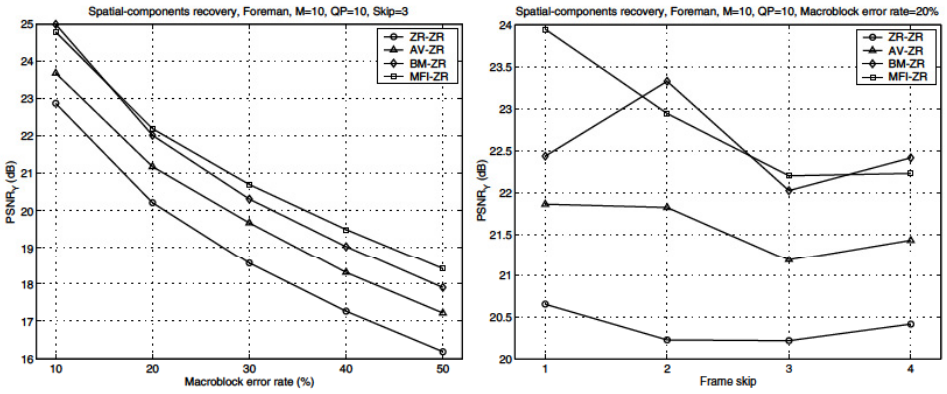
In general, the best spatial-components recovery is achieved by MFI followed by BM. This is similar to the single-reference results reported in Section 10.4. Thus, moving from a single-reference system to a multiple-reference system does not significantly influence the spatial-components



(a) Performance over a range of error rates

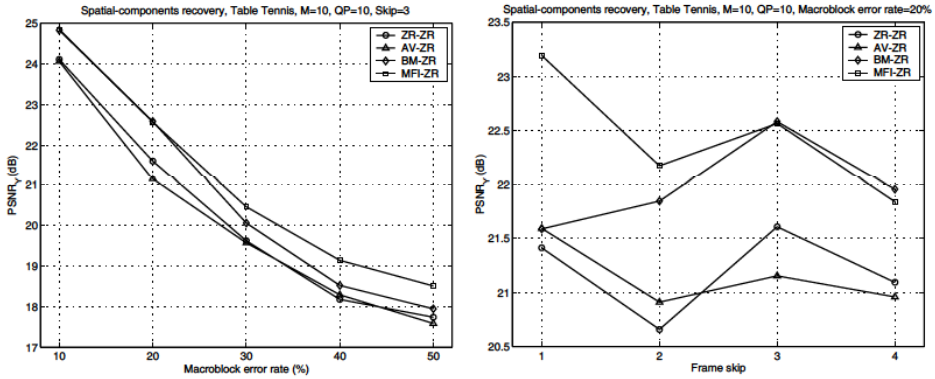
(b) Performance over a range of frame skips

Figure 10.14: Spatial-components recovery for QSIF AKIYO with $M = 10$ and $\mathrm{QP} = 10$



(a) Performance over a range of error rates (b) Performance over a range of frame skips

Figure 10.15: Spatial-components recovery for QSIF FOREMAN with $M = 10$ and QP $= 10$



(a) Performance over a range of error rates (b) Performance over a range of frame skips

Figure 10.16: Spatial-components recovery for QSIF TABLE TENNIS with $M = 10$ and QP $= 10$

recovery process. A very interesting point to note is that the performance of MFI starts to deteriorate at high frame skips. This may be due to the fact that at high frame skips, the spatial components within the motion field become less correlated (see Property 6.3.1.3 and Figures 6.4(a) and 6.4(b)). Since MFI assumes a high correlation between the spatial components, its performance will deteriorate with decreased correlation.

10.5.3 Spatial-Temporal-Components Recovery

Comparing the results of Section 10.5.1 to those of Section 10.5.2 it can be concluded that spatial-components recovery is, in general, more important than

Table 10.4: Spatial-temporal recovery for QSIF AKIYO with $M = 10$, QP = 10, skip = 3, and a macroblock error rate of 30%

		Spatial-components recovery			
		ZR	AV	BM	MFI
Temporal-component recovery	TR	27.91	27.80	26.79	29.22
	AV	27.23	27.49	27.07	28.58
	BM	28.33	28.10	27.38	**29.48**
	MFI	27.42	27.58	26.38	28.69

Table 10.5: Spatial-temporal recovery for QSIF FOREMAN with $M = 10$, QP = 10, skip = 3, and a macroblock error rate of 30%

		Spatial-components recovery			
		ZR	AV	BM	MFI
Temporal-component recovery	TR	18.57	19.68	20.32	20.71
	AV	18.25	19.58	19.72	20.56
	BM	18.59	20.13	20.80	**21.18**
	MFI	18.14	19.51	19.65	20.48

temporal-component recovery. For example, in Figure 10.13(b), at a frame skip of 3, moving from the best technique, ZR-BM, to the worst technique, ZR-MFI, drops the quality by about 0.3 dB, whereas in Figure 10.16(b) moving from the best technique, MFI-ZR, to the worst technique, AV-ZR, drops the quality by about 1 dB. It can be concluded also that spatial-components recovery is, in general, more complex than temporal-component recovery. With temporal-component recovery, a simple technique like ZR can be sufficient, whereas with spatial-components recovery more complex techniques like MFI and BM are essential. Furthermore, the results of Sections 10.5.1 and 10.5.2 indicate that the combination MFI-BM (i.e., spatial recovery using MFI and temporal recovery using BM) may provide the best spatial-temporal recovery. This is confirmed in Tables 10.4, 10.5, and 10.6, which show the performance of all 16 possible combinations with a frame skip of 3 and a macroblock error rate of 30%.

10.5.4 Multihypothesis Temporal Error Concealment

It was demonstrated in Section 10.4 that a more robust performance can be achieved if the concealed block is a weighted average of a number of

Table 10.6: Spatial-temporal recovery for QSIF TABLE TENNIS with $M = 10$, QP = 10, skip = 3, and a macroblock error rate of 30%

		Spatial-components recovery			
		ZR	AV	BM	MFI
Temporal-component recovery	TR	19.62	19.57	20.06	20.46
	AV	19.54	19.57	20.02	20.40
	BM	19.68	19.87	20.09	**20.58**
	MFI	19.55	19.74	20.00	20.40

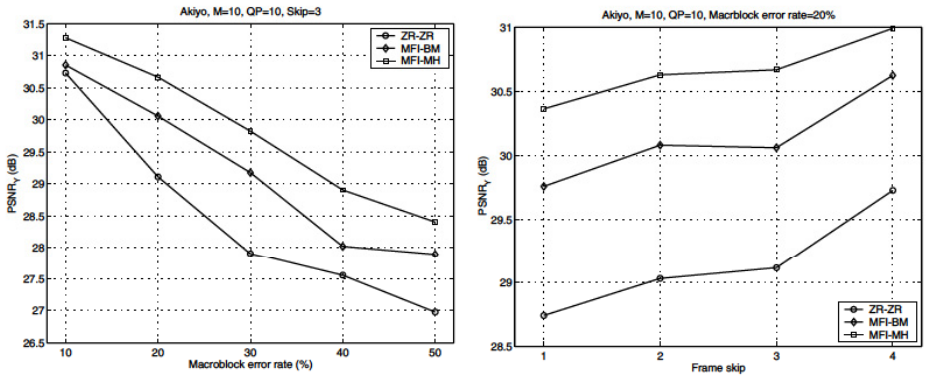
candidate concealments, where each candidate concealment is provided using a different recovered motion vector. This is very similar to multihypothesis motion compensation [106]. Thus, it is termed *multihypothesis temporal error concealment*.

In this subsection a multihypothesis temporal concealment technique to be used with long-term memory motion-compensated prediction is presented. In this case, the candidate concealments are taken from different reference frames. The details of this technique are as follows. The spatial components are first recovered using MFI (as suggested in Section 10.5.2). However, instead of recovering a single temporal component, all four neighboring temporal components are utilized. Combined with the recovered spatial components, each neighboring temporal component provides a candidate concealment from the corresponding reference frame. The four candidate concealments are then averaged and used to conceal the damaged block in the current frame. In other words, a damaged pel (x, y) in the current frame f_c is concealed as follows:

$$\hat{f_c}(x, y) = \frac{1}{4} \sum_{i=1}^{4} f_r(x + \hat{d_x}(x, y), y + \hat{d_y}(x, y), d_{t_i}), \tag{10.13}$$

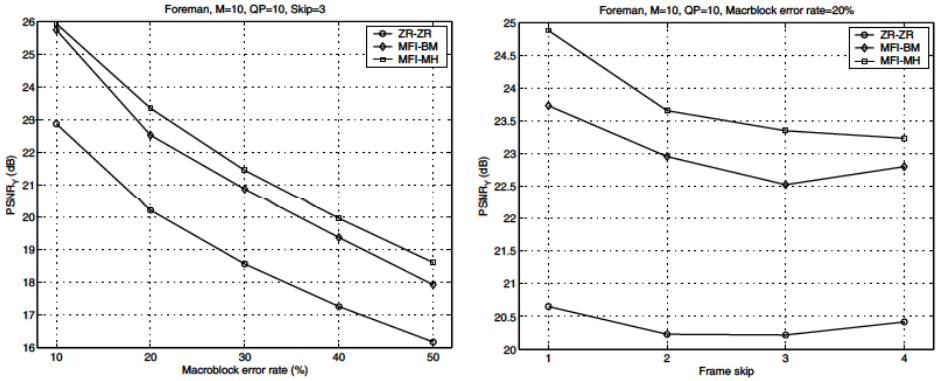
where $f_r(\cdot, \cdot, d_t)$ refers to reference frame d_t in the multiframe memory, $(\hat{d_x}(x, y), \hat{d_y}(x, y))$ are the spatial components recovered at pel (x, y) using MFI, and d_{t_i}, $i = 1, \ldots, 4$ are the temporal components of the four neighboring vectors. In what follows, this approach is designated as MFI-MH.

Figures 10.17, 10.18, and 10.19 compare the performance of the MFI-MH technique to that of MFI-BM (which is the best combination, as suggested in Section 10.5.3) and also to that of ZR-ZR (which is the simplest and most commonly used combination). The figures confirm the superior performance of the suggested combination, MFI-BM, compared to the most commonly used combination, ZR-ZR. In addition, the figures show that further improvements can be achieved using the multihypothesis MFI-MH technique.



(a) Performance over a range of error rates (b) Performance over a range of frame skips

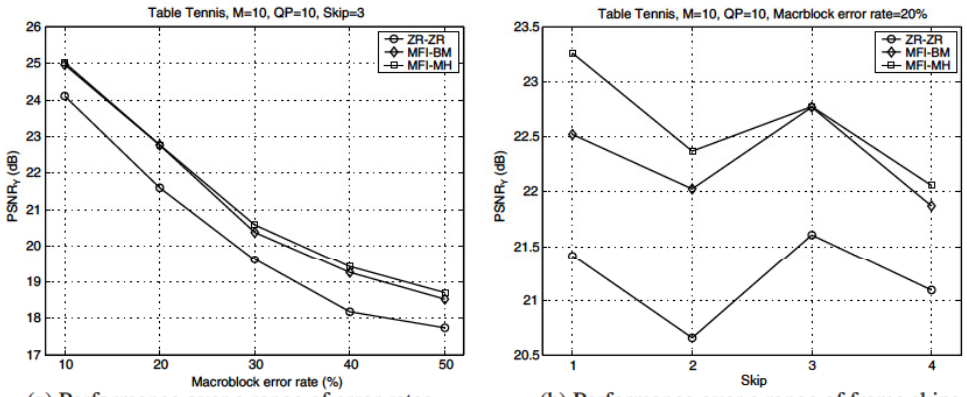
Figure 10.17: Multihypothesis temporal concealment for QSIF AKIYO with $M = 10$ and QP $= 10$



(a) Performance over a range of error rates (b) Performance over a range of frame skips

Figure 10.18: Multihypothesis temporal concealment for QSIF FOREMAN with $M = 10$ and QP $= 10$

This is also confirmed using Figure 10.20, which shows the subjective quality of the 102nd frame of QSIF FOREMAN encoded at 8.33 frames/s with $M = 10$, QP $= 10$, and corrupted with a random macroblock error rate of 20%. Figure 10.20(a) shows the error-free reconstructed frame, whereas Figure 10.20(b) shows the locations of the damaged macroblocks in addition to errors propagated from previous frames. Figures 10.20(c), 10.20(d), and 10.20(e) show the same frame when concealed using ZR-ZR, MFI-BM, and MFI-MH, respectively. The figures clearly show that the suggested MFI-BM combination and the multihypothesis MFI-MH technique both outperform the commonly used ZR-ZR technique. In addition, the figures clearly show the superior subjective quality of the MFI-MH technique (Figure 10.20(e)), even



(a) Performance over a range of error rates (b) Performance over a range of frame skips

Figure 10.19: Multihypothesis temporal concealment for QSIF TABLE TENNIS with $M = 10$ and QP = 10

over that of MFI-BM (Figure 10.20(d)). In particular note the left eye of Foreman (to the right of the viewer) and the diagonal lines in the walls.

10.6 Discussion

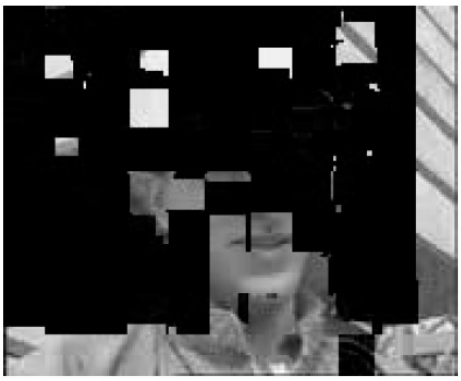
Because of their simplicity, no added redundancy, and minimum delay, error concealment techniques were identified in this chapter as the most suitable techniques for mobile video applications. Thus, it was decided to concentrate on error concealment and in particular on temporal techniques.

Conventional temporal concealment techniques estimate one concealment displacement for the whole damaged block and then use translational displacement compensation to conceal the block from a reference frame. It was realized, therefore, that wrong estimation of the concealment displacement can lead to poor concealment of the entire or most of the block. To overcome this drawback, a novel temporal concealment technique was designed. In this technique, motion field interpolation (MFI) is used to estimate one concealment displacement per pel of the damaged block. Each pel is then concealed individually. In this case, incorrect estimation of a concealment displacement will affect only the corresponding pel rather than the entire block. The inherent motion information recovery and the good motion compensation performance of the MFI technique improve both stages of temporal concealment, i.e., estimation and compensation.

To achieve a more robust performance, a second novel temporal concealment technique was also designed. In this technique, multihypothesis motion compensation (MHMC) is used to combine the MFI technique with a



(a) Error free (32.26 dB)



(b) Locations of errors (with propagation)



(c) Concealed using ZR-ZR (20.62 dB)



(d) Concealed using MFI-BM (22.91 dB)



(e) Concealed using MFI-MH (25.05 dB)

Figure 10.20: Subjective quality of 102nd frame of QSIF FOREMAN encoded at 8.33 frames/s with $M = 10$, QP = 10, and corrupted with a macroblock error rate of 20%

boundary matching (BM) temporal technique. In effect, this improves the second stage of temporal concealment, i.e., compensation.

Simulation results, within both an isolated error environment and an H.263 codec, showed the superior objective and subjective performances of the designed techniques. The MFI technique achieved reasonable improvements over conventional temporal concealment techniques, but it was found that its performance can slightly deteriorate at very high error rates. The combined BM-MFI technique showed a more superior and robust performance at all error rates.

It was also observed that factors like spatial and temporal error propagation, imperfections of the error detection algorithm, scene changes, and uncovered background can severely degrade the performance of temporal concealment techniques. Thus, despite their advantages, such techniques must be combined with spatial techniques and must also be supported by powerful error detection and error containment techniques.

The chapter also investigated the performance of temporal error concealment techniques when incorporated within an LTM-MCP codec. It was found that the best techniques to recover the temporal component are zero replacement (ZR) and boundary matching (BM). The former is sufficient at low frame skips, whereas the latter is preferred at high frame skips. It was also found that the best technique to recover the spatial components is the MFI technique. All these findings were explained in view of the properties of the long-term memory block-motion field. In general, it was concluded that spatial-components recovery is more complex and more important than temporal-component recovery. In addition, a combination of the form MFI-BM (i.e., spatial recovery using MFI and temporal recovery using BM) will provide the best spatial-temporal recovery. In order to achieve a more robust performance, the chapter described the design of a multihypothesis multiple-reference temporal concealment technique. In this technique, a damaged block is concealed using the average of four candidate concealments, probably from different reference frames. Simulation results showed the superior performance of this technique.

Appendix

Fast Block-Matching Algorithms

A.1 Notation and Assumptions

- BDM: a block distortion measure, like the SSD or SAD.
- d_m: maximum allowed motion displacement.
- N: total number of steps in the search. It is an integer number greater than 0.
- s: current search step size.
- (c_x, c_y): current search center.
- (m_x, m_y): current location of minimum distortion.
- (d_x, d_y): final motion vector.
- $\lfloor \cdot \rfloor$: floor operator. It rounds its argument to the nearest integer toward $-\infty$.
- $\lceil \cdot \rceil$: ceil operator. It rounds its argument to the nearest integer toward $+\infty$.
- min: minimize operator. It returns the minimum of a given function.
- max: maximize operator. It returns the maximum of a given function.
- arg: argument operator. It returns the argument of a given function.
- All algorithms presented in this appendix assume full-pel accuracy. Sub-pel accuracy can easily be achieved using very minor modifications.
- If the search procedure attempts to search a location outside the search window, the corresponding BDM is set to a maximum value.

- It is assumed that the search procedure keeps a record of all locations searched so far and their BDM values. This avoids reevaluating the same BDMs in subsequent steps.

A.2 The Two-Dimensional Logarithmic (TDL) Search

The two-dimensional logarithmic (TDL) search was proposed by Jain and Jain in 1981 [54]. It uses a uniform search pattern of five locations (the center and endpoints of a + shape). At each step, the search pattern is centered at the minimum location from the previous step. The step size is halved if the center of the search is the same as that of the previous step. The search is stopped when the step size is 1. In this case, nine locations, rather than five, are searched (the center and endpoints of a $*$ shape) to find the final motion vector. The TDL algorithm is described in the following procedure.

1. Initialize the search step size to
$$s = \max(2, 2^{\lfloor \log_2 d_m \rfloor - 1}).$$
2. Initialize the center of search to the origin of the search window:
$$(c_x, c_y) = (0, 0).$$
3. Evaluate the BDM at the center of the search and its four vertical and horizontal neighbors at a step size of s. Out of this set of five locations, find the one that achieves the minimum BDM:
$$(m_x, m_y) = \arg \min_{(i,j) \in \mathscr{P}_1} \text{BDM}(i, j),$$
where
$$\mathscr{P}_1 = \{(c_x, c_y), (c_x + s, c_y), (c_x - s, c_y), (c_x, c_y + s), (c_x, c_y - s)\}.$$
4. IF the minimum is at the center of the search pattern, i.e., if $(m_x, m_y) = (c_x, c_y)$, THEN
   (a) Halve the search step size:
   $$s = \frac{s}{2}.$$
   (b) IF the search step size is 1, i.e., if $s = 1$, THEN
      i. Evaluate the BDM at the center of the search and its eight immediate neighbors. Out of this set of nine locations, set the motion vector to the one that achieves the minimum BDM:
      $$(d_x, d_y) = \arg \min_{(i,j) \in \mathscr{P}_2} \text{BDM}(i, j),$$

where

$$\mathcal{P}_2 = \{(c_x, c_y), (c_x + 1, c_y), (c_x - 1, c_y), (c_x, c_y + 1), (c_x, c_y - 1), (c_x - 1, c_y - 1), (c_x - 1, c_y + 1), (c_x + 1, c_y - 1), (c_x + 1, c_y + 1)\}.$$

ii. `STOP`

(c) `ELSE` (when the step size is not 1, i.e., $s \neq 1$) `GOTO` step 3.

5. `ELSE` (when the minimum is not in the center, i.e., $(m_x, m_y) \neq (c_x, c_y)$)

(a) Set the center of the search to the new minimum location:

$$(c_x, c_y) = (m_x, m_y).$$

(b) `GOTO` step 3.

A.3 The N-Steps Search (NSS)

This is the general form of the three-steps search (TSS) reported by Koga *et al.* in 1981 [145]. It uses a uniform search pattern of nine locations (the center and endpoints of a $*$ shape). At each step, the step size is halved and the search pattern is centered at the minimum location from the previous step. The search is stopped when the step size is 1. The TSS starts with a step size of ± 4 pels in the first step, then ± 2 pels in the second step and ± 1 pel in the third step. This gives a maximum allowed displacement of $\pm 4 \pm 2 \pm 1 = \pm 7$ pels. For larger search windows the number of steps must be increased. This is called the N-steps search and is described in the following procedure.

1. Find the required number of steps N such that

$$2^{N-1} \leq d_m \leq 2^N.$$

2. Initialize the search step size to

$$s = 2^{N-1}.$$

3. Initialize the center of search to the origin of the search window:

$$(c_x, c_y) = (0, 0).$$

4. Evaluate the BDM at the center of the search and its eight neighbors at a step size of s. Out of this set of nine locations, find the one that achieves the minimum BDM:

$$(m_x, m_y) = \arg \min_{(i,j) \in \mathcal{P}} \text{BDM}(i, j),$$

where

$$\mathcal{P} = \{(c_x, c_y), (c_x + s, c_y), (c_x - s, c_y), (c_x, c_y + s), (c_x, c_y - s), (c_x - s, c_y - s), (c_x - s, c_y + s), (c_x + s, c_y - s), (c_x + s, c_y + s)\}.$$

5. IF the search step size is 1, i.e., if $s = 1$, THEN
   (a) Set the final motion vector to the minimum location found so far:
   $$(d_x, d_y) = (m_x, m_y).$$
   (b) STOP.
6. ELSE (when the step size is not 1, i.e. $s \neq 1$)
   (a) Halve the step size:
   $$s = \frac{s}{2}.$$
   (b) Set the center of the search to the minimum location:
   $$(c_x, c_y) = (m_x, m_y).$$
   (c) GOTO step 4.

A.4 The One-at-a-Time Search (OTS)

The one-at-a-time search (OTS) was proposed by Srinivasan and Rao in 1985 [146]. It uses two fixed-size uniform patterns. The search starts using a horizontal pattern of one center location and its immediate left and right neighbors. At each step, this search pattern is moved horizontally and centered at the minimum location from the previous step. This continues until the minimum is in the center of the pattern (i.e., the minimum is the same as that of the previous step). In this case, the search switches to the vertical direction using a pattern of one center location and its immediate top and bottom neighbors. This is explained in the following procedure.

1. Initialize the center of search to the origin of the search window:
   $$(c_x, c_y) = (0, 0).$$
2. Evaluate the BDM at the center of the search and its immediate left and right neighbors. Out of this set of three locations, find the one that achieves the minimum BDM:
   $$(m_x, m_y) = \arg \min_{(i,j) \in \mathcal{P}_h} \text{BDM}(i, j),$$
   where
   $$\mathcal{P}_h = \{(c_x, c_y), (c_x + 1, c_y), (c_x - 1, c_y)\}.$$
3. IF the minimum is at the center of the search pattern, i.e., $(m_x, m_y) = (c_x, c_y)$, THEN GOTO step 5 (i.e., vertical direction).
4. ELSE
   (a) Move the center of the search to the new minimum location:
   $$(c_x, c_y) = (m_x, m_y).$$
   (b) GOTO step 2 (i.e., continue in the horizontal direction).

5. Evaluate the BDM at the center of the search and its immediate top and bottom neighbors. Out of this set of three locations, find the one that achieves the minimum BDM:

$$(m_x, m_y) = \arg \min_{(i,j) \in \mathcal{P}_v} \text{BDM}(i,j),$$

where

$$\mathcal{P}_v = \{(c_x, c_y), (c_x, c_y + 1), (c_x, c_y - 1)\}.$$

6. `IF` the minimum is at the center of the search pattern, i.e., $(m_x, m_y) = (c_x, c_y)$, `THEN`
   (a) Set the final motion vector to the current minimum:

$$(d_x, d_y) = (m_x, m_y)$$

   (b) `STOP`.
7. `ELSE`
   (a) Move the center of the search to the new minimum location:

$$(c_x, c_y) = (m_x, m_y).$$

   (b) `GOTO` step 5 (i.e., continue in the vertical direction).

A.5 The Cross-Search Algorithm (CSA)

The cross-search algorithm (CSA) was proposed by Ghanbari in 1990 [147]. The search includes an early-termination criterion where, in the first step, a threshold is used to detect if the block is stationary. The search starts with a uniform pattern of five locations (the center and endpoints of an $\times$ shape). At each step, the search step size is halved and the search pattern is centered at the minimum location from the previous step. The search is stopped when the step size is 1. In this case, the search switches to one of two uniform patterns: five locations using either an $\times$ shape or a $+$ shape. This is explained in the following procedure.

1. Evaluate BDM$(0,0)$.
2. `IF` BDM$(0,0) <$ Threshold, `THEN STOP`.
3. Initialize the center of search to the origin of the search window:

$$(c_x, c_y) = (0,0).$$

4. Initialize the search step size to half the maximum allowed displacement:

$$s = \left\lceil \frac{d_m}{2} \right\rceil.$$

5. Evaluate the BDM at the center of the search and its four diagonal neighbors at a step size of s. Out of this set of five locations, find the one that achieves the minimum BDM:

$$(m_x, m_y) = \arg \min_{(i,j) \in \mathscr{P}_1} \text{BDM}(i, j),$$

where

$$\mathscr{P}_1 = \{(c_x, c_y), (c_x - s, c_y - s), (c_x + s, c_y - s), (c_x - s, c_y + s), (c_x + s, c_y + s)\}.$$

6. `IF` search step size is 1, i.e., if $s = 1$, `THEN`

   (a) `IF` the minimum (m_x, m_y) is one of the three locations (c_x, c_y), $(c_x - 1, c_y - 1)$, or $(c_x + 1, c_y + 1)$, `THEN`

      i. Set the center of search to the minimum location:

$$(c_x, c_y) = (m_x, m_y).$$

      ii. Evaluate the BDM at the center of the search and its four horizontal and vertical immediate neighbors. Out of this set of five locations, set the motion vector to the one that achieves the minimum BDM:

$$(d_x, d_y) = \arg \min_{(i,j) \in \mathscr{P}_2} \text{BDM}(i, j),$$

      where

$$\mathscr{P}_2 = \{(c_x, c_y), (c_x - 1, c_y), (c_x + 1, c_y), (c_x, c_y - 1), (c_x, c_y + 1)\}.$$

      iii. `STOP`.

   (b) `ELSE`

      i. Set the center of search to the minimum location:

$$(c_x, c_y) = (m_x, m_y).$$

      ii. Evaluate the BDM at the center of the search and its four diagonal immediate neighbors. Out of this set of five locations, set the motion vector to the one that achieves the minimum BDM:

$$(d_x, d_y) = \arg \min_{(i,j) \in \mathscr{P}_3} \text{BDM}(i, j),$$

      where

$$\mathscr{P}_3 = \{(c_x, c_y), (c_x - 1, c_y - 1), (c_x + 1, c_y - 1), (c_x - 1, c_y + 1), (c_x + 1, c_y + 1)\}.$$

      iii. `STOP`.

7. `ELSE` (when step size is not 1, i.e., $s \neq 1$)

   (a) Halve the step size:

$$s = \left\lceil \frac{s}{2} \right\rceil.$$

   (b) Set the center of the search to the minimum location:

$$(c_x, c_y) = (m_x, m_y).$$

   (c) `GOTO` step 5.

A.6 The Diamond Search (DS)

The diamond search (DS) algorithm was proposed by Zhu and Ma in 1997 [150,151]. An identical version of the algorithm has also been proposed by Tham *et al.* in 1998 [149]. The algorithm uses two fixed-search patterns. It starts with a pattern of nine locations forming a diamond with a step size of 2. At each step, this search pattern is centered at the minimum location from the previous step. This process continues until the minimum is in the center of the pattern (i.e., the minimum is the same as that of the previous step). In this case, the algorithm switches to the second pattern. This consists of five locations forming a diamond with a step size of 1. This pattern is used only once and the search is then terminated. This is explained in the following procedure.

1. Initialize the center of search to the origin of the search window:
$$(c_x, c_y) = (0, 0).$$
2. Evaluate the BDM at nine locations forming a diamond with a step size of 2 centered at the current center location (c_x, c_y). Out of this set of nine locations, find the one that achieves the minimum BDM:
$$(m_x, m_y) = \arg \min_{(i,j) \in \mathcal{P}_{d_1}} \text{BDM}(i, j),$$
where
$$\mathcal{P}_{d_1} = \{(c_x, c_y), (c_x + 2, c_y), (c_x - 2, c_y), (c_x, c_y + 2), (c_x, c_y - 2)$$
$$(c_x - 1, c_y - 1), (c_x + 1, c_y - 1), (c_x - 1, c_y + 1), (c_x + 1, c_y + 1)\}.$$
3. `IF` the minimum is at the center of the search pattern, i.e., $(m_x, m_y) = (c_x, c_y)$, `THEN`
   (a) Evaluate the BDM at five locations forming a diamond with a step size of 1 centered at the current center location (c_x, c_y). Out of this set of five locations, set the motion vector to the one that achieves the minimum BDM:
   $$(d_x, d_y) = \arg \min_{(i,j) \in \mathcal{P}_{d_2}} \text{BDM}(i, j),$$
   where
   $$\mathcal{P}_{d_2} = \{(c_x, c_y), (c_x + 1, c_y), (c_x - 1, c_y), (c_x, c_y + 1), (c_x, c_y - 1)\}.$$
   (b) `STOP`.
4. `ELSE` (when the minimum is not in the center, i.e., $(m_x, m_y) \neq (c_x, c_y)$),
   (a) Set the center of the search to the new minimum location:
   $$(c_x, c_y) = (m_x, m_y).$$
   (b) `GOTO` step 2.

Bibliography

[1] UMTS Forum. The future mobile market: Global trends and developments with a focus on Western Europe. Report 8, UMTS Forum, March 1999. Download from http://www.umts-forum.org/reports.html.

[2] The GSM Association. http://www.gsmworld.com.

[3] Mobile GPRS. http://www.mobileGPRS.com.

[4] The Electronics & Communications Division of the IEE. Special issue on the universal mobile telecommunications system. *IEE Electronics & Communication Engineering Journal*, 12(3):89–152, June 2000.

[5] M. Budagavi, W. R. Heinzelman, J. Webb, and R. Talluri. Wireless MPEG-4 video communication on DSP chips. *IEEE Signal Processing Magazine*, 17(1):36–53, January 2000.

[6] A. Launiainen, A. Jore, E. Ryytty, T. Hämäläinen, and J. Saarinen. Evaluation of TMS320C62 performance in low-bit-rate video encoding. In *Proceedings of the IEEE Third Annual Multimedia and Applications Conference (MTAC)*, pages 364–368, Anaheim, CA, 15–17 September 1998.

[7] D. R. Bull, C. N. Canagarajah, and A. R. Nix, editors. *Insights into Mobile Multimedia Communications*. Signal Processing and Its Applications. Academic Press, London, 1999.

[8] D. T. Hoang, P. M. Long, and J. S. Vitter. Efficient cost measures for motion estimation at low bit rates. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(4):488–500, August 1998.

[9] C. A. Poynton. Frequently asked questions about color. http://home.inforamp.net/~poynton/ColorFAQ.html.

[10] A. Murat Tekalp. *Digital Video Processing*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs, NJ, 1995.

[11] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall. *MPEG Video Compression Standard*. Digital Multimedia Standards Series. Chapman & Hall, New York, 1996.

[12] C. A. Poynton. Frequently asked questions about gamma. http://home.inforamp.net/~poynton/GammaFAQ.html.

[13] A. N. Netravali and B. G. Haskell. *Digital Pictures: Representation, Compression and Standards*, 2nd edition. Applications of Communications Theory Series. Plenium Press, New York, 1995.

[14] CCIR. Recommendation 601-2: Encoding parameters of digital television for studios. In *Digital Methods of Transmitting Television Information*, pages 95–104. CCIR (currently ITU-R), 1990.

[15] M. Ghanbari. *Video Coding—an Introduction to Standard Codecs*. Volume 42 of IEE Telecommunications Series. The Institution of Electrical Engineers IEE, London, 1999.

[16] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27(3):379–423, 1948.

[17] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. Part 4, IRE National Convention Record, 1959.

[18] T. Berger. *Rate Distortion Theory*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[19] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6:7–12, March 1960.

[20] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–137, March 1982.

[21] R. C. Wood. On optimum quantization. *IEEE Transactions on Information Theory*, 15(2):248–252, 1969.

[22] D. A. Huffman. A method for the construction of minimum redundancy codes. *Proceedings of the IRE*, 40:1098–1101, 1952.

[23] M. Hankamer. A modified huffman procedure with reduced memory requirements. *IEEE Transactions on Communication*, 27(6):930–932, 1979.

[24] G. G. Langdon. An introduction to arithmetic coding. *IBM Journal, Research and Development*, 28(2):135–149, 1984.

[25] ITU-R. Methodology for the subjective assessment of the quality of television pictures. Recommendation BT.500-6, ITU-R, 1994.

[26] K. T. Tan, M. Ghanbari, and D. E. Pearson. An objective measurement tool for MPEG video quality. *Signal Processing*, 7:279–294, 1998.

[27] T. Alpert, V. Baroncini, D. Choi, L. Contin, R. Koenen, F. Pereira, and H. Peterson. Subjective evaluation of MPEG-4 video codec proposals: Methodological approach and test procedures. *Signal Processing: Image Communication*, 9(4):305–325, May 1997.

[28] C. C. Cutler. Differential quantization of communication signals. U.S. Patent No. 2605361, 29 July 1952.

[29] M. Rabbani and P. W. Jones. *Digital Image Compression Techniques*. Volume TT7 of SPIE Tutorial Texts in Optical Engineering. SPIE–The International Society for Optical Engineering, Washington, DC, 1991.

[30] R. J. Clarke. *Transform Coding of Images*. Microelectronics and Signal Processing. Academic Press, London, 1985.

[31] E. Feig and S. Winograd. Fast algorithms for the discrete cosine transform. *IEEE Transactions on Signal Processing*, 40(9):2174–2193, September 1992.

[32] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23:90–93, January 1974.

[33] K. R. Rao and P. Yip. *Discrete Cosine Transform — Algorithms, Advantages, Applications*. Academic Press, San Diego, CA, 1990.

[34] D. E. Pearson and M. W. Whybray. Transform coding of images using interleaved blocks. *IEE Proceedings, Part F*, 131:466–472, August 1984.

[35] H. S. Malvar and D. H. Staelin. The LOT: Transform coding without blocking effects. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(4):553–559, April 1989.

[36] S. Minami and A. Zakhor. An optimization approach for removing blocking effects in transform coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(2):74–82, April 1995.

[37] S. Nanda and W. A. Pearlman. Tree coding of image subbands. *IEEE Transactions on Image Processing*, 1(2):133–147, 1992.

[38] R. E. Crochiere, S. A. Webber, and F. L. Flanagan. Digital coding of speech in subbands. *Bell Systems Technical Journal*, 55(8):1069–1085, 1976.

[39] J. Woods and S. O'Neil. Subband coding of images. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-34:1278–1288, October 1986.

[40] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41:3445–3462, 1993.

[41] A. Said and W. A. Pearlman. Image compression using the spatial orientation tree. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 279–282, Chicago, May 1993.

[42] A. Said and W. A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243–250, June 1996.

[43] O. Egger, W. Li, and M. Kunt. High compression image coding using an adaptive morphological subband decomposition. *Proceedings of the IEEE*, 83:272–287, February 1995.

[44] D. W. Redmill and D. R. Bull. Non-linear perfect reconstruction filter banks for image coding. In *Proceedings of the IEEE International*

Conference on Image Processing (ICIP), pages 593–596, Lausanne, Switzerland, 16–19 September 1996.

[45] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28(1):84–95, 1980.

[46] P. A. Chou, T. Lookabaugh, and R. M. Gray. Entropy-constrained vector quantization. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(1):31–42, January 1989.

[47] Chang-Hsing Lee and Ling-Hwei Chen. A fast search algorithm for vector quantization using mean pyramids of codewords. *IEEE Transactions on Communications*, 43:1697–1702, 1995.

[48] T. D. Lookabaugh and R. M. Gray. High-resolution quantization theory and the vector quantizer advantage. *IEEE Transactions on Information Theory*, IT-35:1020–1033, 1989.

[49] M. Kunt, A. Ikonomopoulos, and M. Kocher. Second-generation image-coding techniques. *Proceedings of the IEEE*, 73(4):549–574, April 1985.

[50] M. Kunt, M. Bernard, and R. Leonardi. Recent results in high-compression image coding. *IEEE Transactions on Circuits and Systems*, CAS-34(11):1306–1336, November 1987.

[51] M. E. Al-Mualla. Second-generation image coding techniques. Master's thesis, University of Bristol, Faculty of Engineering, Department of Electrical and Electronics Engineering, October 1996.

[52] R. J. Clarke. *Digital Compression of Still Images and Video*. Signal Processing and Its Applications. Academic Press, London, 1995.

[53] B. G. Haskell, F. W. Mounts, and J. C. Candy. Interframe coding of videotelephone pictures. *Proceedings of the IEEE*, 60:792–800, July 1972.

[54] J. R. Jain and A. K. Jain. Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, COM-29(12):1799–1808, December 1981.

[55] F. Dufaux and F. Moscheni. Motion estimation techniques for digital TV: A review and a new contribution. *Proceedings of the IEEE*, 83(6):858–875, June 1995.

[56] M. F. Chowdhury, A. F. Clark, A. C. Downton, and D. E. Pearson. A switched model-based coder for video signals. *IEEE Transactions on Circuits and Systems for Video Technology*, 4:216–217, June 1994.

[57] D. E. Pearson. Developments in model-based video coding. *Proceedings of the IEEE*, 83(6):892–906, June 1995.

[58] CCITT/SG XV. Codecs for videoconferencing using primary digital group transmission. Recommendation H.120, CCITT (currently ITU-T), Geneva, 1989.

[59] CCITT/SG-XV.-Video-codec-for-audiovisual-services-at-$p \times 64$-kbit/s.-Recommendation-H.261,-CCITT-(currently-ITU- T),-Geneva,-1993.-

[60] CCIR.- Transmission- of- component-coded- digital- television- signals-for- contribution-quality- applications- at- bit- rates- near- 140-Mbit/s.-Recommendation-721,-CCIR-(currently-ITU-R),-Geneva,-1990.-

[61] CCIR.-Digital-coding-of-component-television-signals-for-contribution-quality-applications-in-the-range-34–45-Mbit/s.-Recommendation-723,-CCIR-(currently-ITU-R),-Geneva,-1992.-

[62] ISO/IEC- JTC1/SC29/WG11.- Information- technology—Coding- of-moving-pictures-and-associated-audio-for-digital-storage-media-at-up-to-about-1.5-Mbits/s.-Part-2:-Video.-Draft-ISO/IEC-11172-2-(MPEG-1),-ISO/IEC,-Geneva,-1991.-

[63] ISO/IEC- JTC1/SC29/WG11- and- ITU-T/SG15.- Information- technology—Generic-coding-of-moving-pictures-and-associated-audio.-Part-2:-Video.-Draft-ISO/IEC-13818-2-(MPEG-2)-and-ITU-T-Recommendation-H.262,-ISO/IEC-and-ITU-T,-Geneva,-1994.-

[64] ITU-T/SG15.- Video- coding- for- low- bitrate- communication.- ITU-T-Recommendation-H.263,-Version-1,-ITU-T,-Geneva,-1996.-

[65] B.-Girod,-E.-Steinbach,-and-N.-Färber.-Performance-of-the-H.263-video-compression- standard.- *Journal of VLSI Signal Processing: Systems for Signal, Image, and Video Technology*,-17:101–111,-1997.-

[66] ITU-T/SG16/Q15.-Video-coding-for-low-bitrate-communication.-ITU-T-Recommendation-H.263,-Version-2-(H.263+),-ITU-T,-Geneva,-1998.-

[67] ISO/IEC-JTC1/SC29/WG11.-Information-technology—Generic-coding-of- audio-visual- objects.- Part- 2:- Visual.- Draft- ISO/IEC- 14496-2-(MPEG-4),-Version-1,-ISO/IEC,-Geneva,-1998.-

[68] ITU-T/SG16/Q15.- Draft- for- "H.263++"- annexes- U,- V,- and- W- to-recommendation-H.263.-Draft,-ITU-T,-Geneva,-2000.-

[69] R.-Schäfer-and-T.-Sikora.-Digital-video-coding-standards-and-their-role-in- video- communications.- *Proceedings of the IEEE*,- 83(6):907–924,-June-1995.-

[70] T.- Ebrahimi- and- M.- Kunt.- Visual- data- compression- for- multimedia-applications.-*Proceedings of the IEEE*,-86(6):1109–1125,-June-1998.-

[71] G.-Côté,-B.-Erol,-M.-Gallant,-and-F.-Kossentini.-H.263 | :-Video-coding-at-low-bit-rates.-*IEEE Transactions on Circuits and Systems for Video Technology*,-8(7):849–866,-November-1998.-

[72] S.-Wenger,-G.-Knorr,-J.-Ott,-and-F.-Kossentini.-Error-resilience-support-in- H.263+.- *IEEE Transactions on Circuits and Systems for Video Technology*,-8(7):867–877,-November-1998.-

[73] T.- Sikora.- MPEG- digital- video-coding- standards.- *IEEE Signal Processing Magazine*,-pages-82–100,-September-1997.-

[74] F. Pereira, K. O'Connell, R. Koenen, and M. Etoh. Special issue on MPEG-4, part 1: Invited papers. *Signal Processing: Image Communication*, 9(4):291–477, May 1997.

[75] F. Pereira. Tutorial issue on the MPEG-4 standard. *Signal Processing: Image Communication*, 15(4–5):269–478, January 2000.

[76] Telenor Research. Video codec test model. TMN5, ITU-T/SG15/WP 15/1 Expert's Group on Very Low Bitrate Visual Telephony, Geneva, 1995.

[77] Y. T. Tse and R. L. Baker. Global zoom/pan estimation and compensation for video compression. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 2725–2728, Toronto, May 1991.

[78] C. R. Moloney and E. Dubois. Estimation of motion fields from image sequences with illumination variation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 2425–2428, Toronto, May 1991.

[79] M. Bertero, T. A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, August 1988.

[80] C. Stiller and J. Konrad. Estimating motion in image sequences. *IEEE Signal Processing Magazine*, 16(4):70–91, July 1999.

[81] J. O. Limb and J. A. Murphy. Measuring the speed of moving objects from television signals. *IEEE Transactions on Communication*, COM-23(4):474–478, April 1975.

[82] C. Cafforio and F. Rocca. Methods for measuring small displacements of television images. *IEEE Transactions on Information Theory*, IT-22(5):573–579, September 1976.

[83] H. Yamaguchi. Iterative method of movement estimation for television signals. *IEEE Transactions on Communications*, 37(12):1350–1358, December 1989.

[84] H. M. Ming, Y. M. Chou, and S. C. Cheng. Motion estimation for video coding standards. *Journal of VLSI Signal Processing: Systems for Signal, Image, and Video Technology*, 17:113–136, 1997.

[85] A. N. Netravali and J. D. Robbins. Motion compensated television coding: Part I. *Bell System Technical Journal*, 58:631–670, March 1979.

[86] D. R. Walker and K. R. Rao. Improved pel recursive motion compensation. *IEEE Transactions on Communications*, COM-32(10):1128–1134, October 1984.

[87] H. G. Musmann, P. Pirsch, and H. J. Grallert. Advances in picture coding. *Proceedings of the IEEE*, 73(4):523–548, April 1985.

[88] B. G. Haskell. Frame to frame coding of television pictures using two dimensional Fourier transforms. *IEEE Transactions on Information Theory*, 20:119–120, January 1974.

[89] P.-A.-Lynn-and-W.-Fuerst.-*Introductory Digital Signal Processing with Computer Applications*,-revised-edition,-Wiley,-London,-1994.-

[90] C.-D.-Kuglin-and-D.-C.-Hines.-The-phase-correlation-image-alignment-method.- In- *Proceedings of the IEEE International Conference on Cybernetics and Society*,-pages-163–165,-San-Francisco,-1975.-

[91] G.- A.- Thomas.- Television- motion- measurement- for- DATV- and-other- applications.- Technical- Report- 1987/11,- British- Broadcasting-Corporation-(BBC)-Research-Department,-1987.-

[92] B.-Girod.-Motion-compensating-prediction-with-fractional-pel-accuracy.-*IEEE Transactions on Communications*,-41(4):604–612,-April-1993.-

[93] Snell- &- Wilcox- Ltd.- Alchemist- Ph.C- D:- a- phase- correlation- 10-bit-motion-compensated-standards-converter-for-digital-I/O.-`http://www.snellwilcox.com/productguide/linker1/alc2index.html`.-

[94] Y.-M.-Chou-and-H.-M.-Hang.-A-new-motion-estimation-method-using-frequency-components.-*Journal of Visual Communication and Image Representation*,-8(1):83–96,-March-1997.-

[95] R.- W.- Young- and- N.- G.- Kingsbury.- Frequency-domain- motion-estimation-using-a-complex-lapped-transform.-*IEEE Transactions on Image Processing*,-2:2–17,-January-1993.-

[96] U.- V.- Koc and- K.- J.- R.- Liu.- DCT-based- motion- estimation.- *IEEE Transactions on Image Processing*,-7(7):948–965,-July-1998.-

[97] U.- V.- Koc and- K.- J.- R.- Liu.- Interpolation-free- subpixel- motion-estimation-techniques-in-DCT-domain.-*IEEE Transactions on Circuits and Systems for Video Technology*,-8(4):460–487,-August-1998.-

[98] M.-H.-Chan,-Y.-B.-Yu,-and-A.-G.-Constantinides.-Variable-size-block-matching-motion-compensation-with-applications-to-video-coding.-*IEE Proceedings, Part 1*,-137(4):205–212,-August-1990.-

[99] G.- J.- Sullivan- and- R.- L.- Baker.- Efficient- quadtree- coding- of- images-and- video.- In- *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*,- pages-2661–2664,-Toronto,-May-1991.-

[100] S.- Ericsson.- Fixed- and- adaptive- predictors- for- hybrid- predictive/ transform- coding.- *IEEE Transactions on Communications*,- COM-33(12):1291–1302,-December-1985.-

[101] H.- Watanabe- and- S.- Singhal.- Windowed- motion- compensation.- In-*Proceedings of the SPIE Conference on Visual Communications and Image Processing (VCIP)*,-volume-1605,-pages-582–589,-November-1991.-

[102] C.- Auyeung,- J.- Kosmach,- M.- Orchard,- and- T.- Kalafatis.- Overlapped-block-motion-compensation.-In-*Proceedings of the SPIE Conference on Visual Communications and Image Processing (VCIP)*,-volume-1818,-pages-561–572,-November-1992.-

[103] M. T. Orchard and G. J. Sullivan. Overlapped block motion compensation: An estimation-theoretic approach. *IEEE Transactions on Image Processing*, 3(5):693–699, September 1994.

[104] T. Y. Kuo and C. C. J. Kuo. Fast overlapped block motion compensation with checkerboard block partitioning. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(6):705–712, October 1998.

[105] G. J. Sullivan. Multi-hypothesis motion compensation for low-bit-rate video coding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 5, pages 437–440, Minneapolis, April 1993.

[106] B. Girod. Efficiency analysis of multihypothesis motion-compensated prediction for video coding. *IEEE Transactions on Image Processing*, 9(2):173–183, February 2000.

[107] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, 1990.

[108] G. J. Sullivan and R. L. Baker. Motion compensation for video compression using control grid interpolation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 2713–2716, Toronto, May 1991.

[109] C. L. Huang and C. Y. Hsu. A new motion compensation method for image sequence coding using hierarchical grid interpolation. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(1):42–51, February 1994.

[110] J. Nieweglowski and P. Haavisto. Temporal image sequence prediction using motion field interpolation. *Signal Processing: Image Communication*, 7:333–353, 1995.

[111] J. Nieweglowski, T. G. Campbell, and P. Haavisto. A novel video coding scheme based on temporal prediction using digital image warping. *IEEE Transactions on Consumer Electronics*, 39(3):141–150, August 1993.

[112] A. Neri and S. Colonnese. On the computation of warping based motion compensation in video sequence coding. *Signal Processing: Image Communication*, 13:155–160, 1998.

[113] A. Nosratinia and M. T. Orchard. Optimal unified approach to warping and overlapped block motion estimation in video coding. In *Proceedings of the SPIE Conference on Visual Communications and Image Processing (VCIP)*, volume 2727, pages 634–644, Orlando, FL, March 1996.

[114] Y. Nakaya and H. Harashima. Motion compensation based on spatial transformations. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3):339–356, June 1994.

[115] M.- Ghanbari,- S.- de- Faria,- I.- N.- Goh,- and- K.- T.- Tan.- Motion-compensation- for- very- low-bit-rate- video.- *Signal Processing: Image Communication*,-7:567–580,-1995.-

[116] A.- Sharaf- and- F.- Marvasti.- Motion- compensation- using- spatial-transformations- with- forward- mapping.- *Signal Processing: Image Communication*,-14:209–227,-1999.-

[117] F.- J.- P.- Lopes- and- M.- Ghanbari.- Analysis- of- spatial- transform-motion- estimation- with- overlapped- compensation- and- fractional-pixel-accuracy.-*IEE Proceedings on Vision, Image and Signal Processing*,-146(6):339–344,-December-1999.-

[118] C.-A.-Papadopoulos-and-T.-G.-Clarkson.-Motion-compensation-using-second-order- geometric- transformations.- *IEEE Transactions on Circuits and Systems for Video Technology*,-5(4):319–331,-August-1995.-

[119] V.- Seferidis- and- M.- Ghanbari.- General- approach- to- block-matching-motion-estimation.-*Optical Engineering*,-32(7):1464–1474,-July-1993.-

[120] V.- Seferidis- and- M.- Ghanbari.- Generalised- block-matching- motion-estimation- using- quad-tree- structured- spatial- decomposition.- *IEE Proceedings on Vision, Image and Signal Processing*,- 141(6):446—452,-December-1994.-

[121] Y.- Wang- and- O.- Lee.- Active- mesh—a- feature- seeking- and- tracking-image-sequence-representation-scheme.-*IEEE Transactions on Image Processing*,-3(5):610–624,-September-1994.-

[122] Y.- Wang- and- O.- Lee.- Use- of- two-dimensional- deformable- mesh-structures-for-video-coding.-Part-I—the-synthesis-problem:-Mesh-based-function-approximation-and-mapping.-*IEEE Transactions on Circuits and Systems for Video Technology*,-6(6):637–646,-December-1996.-

[123] Y.-Wang,-O.-Lee,-and-A.-Vetro.-Use-of-two-dimensional-deformable-mesh- structures- for- video- coding.- Part- II—the- analysis- problem-and- a- region-based- coder- employing- an- active- mesh- representation.-*IEEE Transactions on Circuits and Systems for Video Technology*,-6(6):647–659,-December-1996.-

[124] M.-Dudon,-O.-Avaro,-and-C.-Roux.-Triangular-active-mesh-for-motion-estimation.-*Signal Processing: Image Communication*,-10:21–41,-1997.-

[125] Y.-Altunbasak-and-A.-M.-Tekalp.-Closed-form-connectivity-preserving-solutions- for- motion- compensation- using- 2-D- meshes.- *IEEE Transactions on Image Processing*,-6(9):1255–1269,-September-1997.-

[126] Y.-Wang-and-J.-Osterman.-Evaluation-of-mesh-based-motion-estimation-in-H.263-like-coders.-*IEEE Transactions on Circuits and Systems for Video Technology*,-8(3):243–252,-June-1998.-

[127]- A.-M.-Tekalp,-P.-V.-Beek,-C.-Toklu,-and-B.-Günsel.-Two-dimensional-mesh-based-visual-object-representation-for-interactive-synthetic/natural-digital-video.-*Proceedings of the IEEE*,-86(6):1029–1051,-June-1998.-

[128] H. Brusewitz. Motion compensation with triangles. In *Proceedings of the 3rd International Workshop on 64 kbits/s Coding of Moving Video*, Free session, Rotterdam, September 1990.

[129] K. T. Tan, I. N. Goh, and M. Ghanbari. Fast motion estimation with spatial transformation. *IEE Electronics Letters*, 30:847–849, 1994.

[130] D. B. Bradshaw and N. G. Kingsbury. A fast, two-stage translational and warping motion compensation scheme. In *Proceedings of the IX European Signal Processing Conference (EUSIPCO)*, volume II, pages 905–908, Island of Rhodes, Greece, 1998.

[131] ISO/IEC JTC1/SC29/WG11. Core experiment on global motion compensation (P1). In *Description of Core Experiments on Coding Efficiency in MPEG-4 Video*, No. N1385, September 1996.

[132] ISO/IEC JTC1/SC29/WG11. Core experiment N3: Dynamic sprites and global motion compensation. In *Description of Core Experiments on Coding Efficiency in MPEG-4* Video, No. N1875, October 1997.

[133] ISO/IEC JTC1/SC29/WG11. Core experiments on STFM/LTFM for motion prediction (P3). In *Description of Core Experiments on Coding Efficiency in MPEG-4 Video*, No. N1385, September 1996.

[134] N. Mukawa and H. Kuroda. Uncovered background prediction in interframe coding. *IEEE Transactions on Communications*, COM-33(11):1227–1231, November 1985.

[135] T. Wiegand, X. Zhang, and B. Girod. Motion-compensating long-term memory prediction. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 53–56, Santa Barbara, CA, 26–29 October 1997.

[136] T. Wiegand, X. Zhang, and B. Girod. Long-term memory motion-compensated prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(1):70–84, February 1999.

[137] T. Wiegand, E. Steinbach, A. Stensrud, and B. Girod. Multiple-reference picture video coding using polynomial motion models. In *Proceedings of the SPIE Conference on Visual Communications and Image Processing (VCIP)*, volume 3309, pages 134–145, San Jose, CA, February 1998.

[138] E. Steinbach, T. Wiegand, and B. Girod. Using multiple global models for improved block-based video coding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 56–60, Kobe, Japan, October 1999.

[139] T. Wiegand, E. Steinbach, and B. Girod. Long-term memory prediction using affine motion compensation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 51–55, Kobe, Japan, October 1999.

[140] T.-Wiegand,-N.-Färber,-B.-Girod,-and-B.-Andrews.-Proposed-draft-for-annex-on-enhanced-reference-picture-selection.-Document-Q15-F-32r1,-ITU-T/SG16/Q.15,-Seoul,-Korea,-November-1998.-Downlond-from-`http://www-nt.etechnik.uni-erlangen.de/~wiegand/publications.html`.-

[141] T.-Wiegand,-B.-Lincoln,-and-B.-Girod.-Fast-search-for-long-term-memory-motion-compensated-prediction.-In-*Proceedings of the IEEE International Conference on Image Processing (ICIP)*,-volume-3,-pages-619–622,-Chicago,-4–7-October-1998.-

[142] T.-R.-Gardos,-editor.-Video-codec-test-model,-near-term,-version-10-(TMN10)-draft-1.-Document-Q15-D-65d1,-ITU-T/SG16/Q.15,-Tampere,-Finland,-April-1998.-Download-from-`ftp://standard.pictel.com/video-site/h263plus/tmn10.doc`.

[143] G.-J.-Sullivan-and-T.-Wiegand.-Rate-distortion-optimization-for-video-compression.-*IEEE Signal Processing Magazine*,-15(6):74–90,-November-1998.-

[144] Telenor-Research-and-Development.-H.263-TMN-software-codec,-version-2.0.-`ftp://bonde.nta.no/pub/tmn/software`.-

[145] T.-Koga,-K.-Iinuma,-A.-Hirano,-Y.-Iijima,-and-T.-Ishiguro.-Motion-compensated-interframe-coding-for-video-conferencing.-In-*Proceedings of the National Telecommunications Conference (NTC)*,-pages-G5.3.1–G5.3.5,-New-Orleans,-November-29–December-3-1981.-

[146] R.-Srinivasan-and-K.-R.-Rao.-Predictive-coding-based-on-efficient-motion-estimation.-*IEEE Transactions on Communications*,-COM-33(8):888–896,-August-1985.-

[147] M.-Ghanbari.-The-cross-search-algorithm.-*IEEE Transactions on Communications*,-38(7):950–953,-July-1990.-

[148] K.-H.-K.-Chow-and-M.-L.-Liou.-Genetic motion-search-algorithm-for-video-compression.-*IEEE Transactions on Circuits and Systems for Video Technology*,-3(6):440–445,-December-1993.-

[149] J.-Y.-Tham,-S.-Ranganath,-M.-Ranganath,-and-A.-A.-Kassim.-A-novel-unrestricted-center-biased-diamond-search-algorithm-for-block-motion-estimation.-*IEEE Transactions on Circuits and Systems for Video Technology*,-8(4):369–377,-August-1998.-

[150] S.-Zhu-and-K.-K.-Ma.-A-new-diamond-search-algorithm-for-fast-block-matching-motion-estimation.-In-*Proceedings of the International Conference on Information, Communication and Signal Processing (ICICS)*,-pages-292–296,-9–12-September-1997.-

[151] S.-Zhu-and-K.-K.-Ma.-A-new-diamond-search-algorithm-for-fast-block-matching-motion-estimation.-*IEEE Transactions on Image Processing*,-9(2):287–290,-February-2000.-

[152] H. Ghavari and M. Mills. Blockmatching motion estimation—New results. *IEEE Transactions on Circuits and Systems*, 37(5):649–651, May 1990.

[153] M. J. Chen, L. G. Chen, T. D. Chiueh, and Y. P. Lee. A new block-matching criterion for motion estimation and its implementation. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(3):231–236, June 1995.

[154] Y. Baek, H. S. Oh, and H. K. Lee. An efficient block matching criterion for motion estimation and its VLSI implementation. *IEEE Transactions on Consumer Electronics*, 42(4):885–892, November 1996.

[155] K. Sauer and B. Schwartz. Efficient block motion estimation using integral projections. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(5):513–518, October 1996.

[156] B. Natarajan, V. Bhaskaran, and K. Konstantinides. Low-complexity block-based motion estimation via one-bit transforms. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(4):702–706, August 1997.

[157] B. Liu and A. Zaccarin. New fast algorithms for the estimation of block motion vectors. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(2):148–157, April 1993.

[158] Y. L. Chan and W. C. Siu. New adaptive pixel decimation for block motion vector estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(1):113–118, February 1996.

[159] M. Bierling. Displacement estimation by hierarchical blockmatching. In *Proceedings of the SPIE Visual Communications and Image Processing (VCIP)*, volume 1001, pages 942–951, 1988.

[160] K. M. Nam, J. S. Kim, R. H. Park, and Y. S. Shim. A fast hierarchical motion vector estimation algorithm using mean pyramid. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(4):344–351, August 1995.

[161] C. D. Bei and R. M. Gray. An improvement of the minimum distortion encoding algorithm for vector quantization. *IEEE Transactions on Communications*, COM-33(10):1132–1133, October 1985.

[162] W. Li and E. Salari. Successive elimination algorithm for motion estimation. *IEEE Transactions on Image Processing*, 4(1):105–107, January 1995.

[163] S. H. Huang and S. H. Chen. Fast encoding algorithm for VQ-based image coding. *IEE Electronics Letters*, 26(19):1618–1619, 13 September 1990.

[164] Y. C. Lin and S. C. Tai. Fast full-search block-matching algorithm for motion-compensated video compression. *IEEE Transactions on Communications*, 45(5):527–531, May 1997.

[165] M.- E.- Al-Mualla,- C.- N.- Canagarajah,- and- D.- R.- Bull.- A- fast- block-matching-motion-estimation-algorithm-based-on-simplex-minimisation.- In- *Proceedings of the IX European Signal Processing Conference (EUSIPCO)*,-volume-III,-pages-1565–1568,-Island-of-Rhodes,-Greece,-8–11-September-1998.-

[166] M.- E.- Al-Mualla,- C.- N.- Canagarajah,- and- D.- R.- Bull.- Simplex-minimisation- for- fast- block- matching- motion- estimation.- *IEE Electronics Letters*,-34(4):351–352,-19-February-1998.-

[167] M.- E.- Al-Mualla,- C.- N.- Canagarajah,- and- D.- R.- Bull.- Simplex-minimisation-for-multiple-reference-motion-estimation.-In-*Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*,-volume-IV,-pages-733–736,-Geneva,-28–31-May-2000.-

[168] M.- E.- Al-Mualla,- C.- N.- Canagarajah,- and- D.- R.- Bull.- Simplex-minimization- for- fast- long-term- memory- motion- estimation.- *IEE Electronics Letters*,-37(5):290–292,-1-March-2001.-

[169] M.- E.- Al-Mualla,- C.- N.- Canagarajah,- and- D.- R.- Bull.- Simplex-minimization- for- single- and- multiple-reference- motion- estimation.- *IEEE Transactions on Circuits and Systems for Video Technology*,-11(12):1209–1220,-December-2001.-

[170] D.- E.- Knuth.- Searching- and- sorting.- In- *The Art of Computer Programming*,-volume-3.-Addison-Wesley,-Reading,-MA,-1973.-

[171] M.- R.- Hestenes.- *Conjugate Direction Methods in Optimization*.-Springer-Verlag,-New-York,-1980.-

[172] D.- E.- Goldberg.- *Genetic Algorithms in Search, Optimization and Machine Learning*.-Addison-Wesley,-Reading,-MA,-1989.-

[173] J.-A.-Nelder-and-R.-Mead.-A-simplex-method-for-function-minimization.-*The Computer Journal*,-7:308–313,-1965.-

[174] W.-H.-Press,-S.-A.-Teukolsky,-W.-T.-Vetterling,-and-B.-P.-Flannery.-*Numerical Recipes in C: The Art of Scientific Computing*,-2nd-ed.-Cambridge-University-Press,-New-York,-1992.-

[175] O.-Sohm.-Fast-block-motion-estimation-algorithm-for-MPEG-4.-Progress-Report-for-the-Project:-MPEG-4-Video-Coding-Using-the-TMS320C62x,-Image-Communications-Group,-Center-for-Communications-Research,-University-of-Bristol,-U.K.,-March-2000.-

[176] ISO/IEC- JTC1/SC29/WG11.- MPEG-4- video- verification- model,-version-14.0.-Document-N2932,-ISO/IEC,-October-1999.-

[177] D.-W.-Redmill.-*Image and Video Coding for Noisy Channels*.-PhD-thesis,- University- of- Cambridge,- Department- of- Engineering,- Signal-Processing-and-Communications-Laboratory,-November-1994.-

[178] T.-J.-Ferguson-and-J.-H.-Rabinowitz.-Self-synchronizing-Huffman-codes.-*IEEE Transactions on Information Theory*,-30:687–693,-1984.-

[179] Y. Wang and Q. F. Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86(5):974–997, May 1998.

[180] Y. Wang, S. Wenger, J. Wen, and A. K. Kastaggelos. Error resilient video coding techniques. *IEEE Signal Processing Magazine*, 17(4):61–82, July 2000.

[181] P. Haskell and D. Messerschmitt. Resynchronization of motion compensated video affected by ATM cell loss. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume III, pages 545–548, San Francisco, March 1992.

[182] G. Côté and F. Kossentini. Optimal intracoding of blocks for robust video communication over the internet. *Signal Processing: Image Communication*, 15(1–2):25–34, September 1999.

[183] J. Y. Liao and J. D. Villasenor. Adaptive intra update for video coding over noisy channels. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume III, pages 763–766, Lausanne, Switzerland, 16–19 September 1996.

[184] G. Côté, S. Shirani, and F. Kossentini. Optimal mode selection and synchronization for robust video communications over error prone networks. *IEEE Journal on Selected Areas in Communications*, 18(6):952–965, June 2000.

[185] D. W. Redmill and N. G. Kingsbury. The EREC: An error resilient technique for coding variable length blocks of data. *IEEE Transactions on Image Processing*, 5(4):565–574, April 1996.

[186] M. Ghanbari. Two layer coding of video signals for VBR networks. *IEEE Journal on Selected Areas in Communications*, 7(5):771–781, June 1989.

[187] M. Khansari and M. Vetterli. Layered transmission of signals over power constrained wireless channels. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume III, pages 380–383, Washington, DC, October 1995.

[188] V. A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Transactions on Information Theory*, 39(3):821–834, May 1993.

[189] Q. F. Zhu, Y. Wang, and L. Shaw. Coding and cell loss recovery in DCT based packet video. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(3):248–258, June 1993.

[190] M. Ghanbari and V. Seferidis. Cell loss concealment in ATM video codecs. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(3):238–247, June 1993.

[191] P. Salama, N. B. Shroff, E. J. Coyle, and E. J. Delp. Error concealment techniques for encoded video streams. In *Proceedings of the IEEE*

International Conference on Image Processing (ICIP),- volume- I,- pages-9–12,-Washington,-DC,-23–26-October-1995.-

[192] A.- Narula- and- J.- Lim.- Error- concealment- techniques- for- an- all-digital- high-definition- television- system.- In- *Proceedings of the SPIE Conference on Visual Communications and Image Processing (VCIP)*,-volume-2094,-pages-304–315,-1993.-

[193] Y.- W.- Wang,- Q.- F.- Zhu,- and- L.- Shaw.- Maximally- smooth- image-recovery-in-transform-coding.-*IEEE Transactions on Communications*,-41(10):1544–1551,-October-1993.-

[194] W.-Kwok-and-H.-Sun.-Multi-directional-interpolation-for-spatial-error-concealment.- *IEEE Transactions on Consumer Electronics*,- 39(3):-455–460,-August-1993.-

[195] P.-Salama,-N.-B.-Shroff,-and-E.-J.-Delp.-A-Bayesian-approach-to-error-concealment- in- encoded- video- streams.- In- *Proceedings of the IEEE International Conference on Image Processing (ICIP)*,- volume- II,-pages-49–51,-Lausanne,-Switzerland,-16–19-September-1996.-

[196] W.- M.- Lam,- A.- R.- Reibman,- and- B.- Liu.- Recovery- of- lost- or-erroneously- received- motion- vectors.- In- *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing ICASSP*,-volume-V,-pages-417–420,-Minnesota,-U.S.A.,-April-1993.-

[197] K.- W.- Kang,- S.- H.- Lee,- and- T.- Kim.- Recovery- of- coded- video-sequences-from-channel-errors.-In-*Proceedings of the SPIE Conference on Visual Communications and Image Processing (VCIP)*,- volume-2501,-pages-19–27,-1995.-

[198] M.- J.- Chen,- L.- G.- Chen,- and- R.- M.- Weng.- Error- concealment-of- lost- motion- vectors- with- overlapped- motion- compensation.- *IEEE Transactions on Circuits and Systems for Video Technology*,-7(3):560-–563,-June-1997.-

[199] S.- Shirani,- F.- Kossentini,- and- R.- Ward.- A- concealment- method-for- video- communications- in- an- error-prone- environment.- *IEEE Transactions on Selected Areas in Communications*,-18(6):1122–1128,-June-2000.-

[200] H.- Sun,- K.- Challapali,- and- J.- Zdepski.- Error- concealment- in- digital-simulcast- AD-HDTV- decoder.- *IEEE Transactions on Consumer Electronics*,-38(3):108–117,-August-1992.-

[201] B.-Girod-and-N.-Färber.-Feedback-based-error-control-for-mobile-video-transmission.- *Proceedings of the IEEE*,- 87(10):1707–1723,- October-1999.-

[202] E.-Steinbach,-N.-Färber,-and-N.-Girod.-Standard-compatible-extension-of- H.263- for- robust- video- transmission- in- mobile- environments.- *IEEE Transactions on Circuits and Systems for Video Technology*,-7(6):872–881,-December-1997.-

[203] M. Wada. Selective recovery of video packet loss using error concealment. *IEEE Journal on Selected Areas in Communications*, 7(5):807–814, June 1989.

[204] M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull. Error concealment using motion field interpolation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume II, pages 512–516, Chicago, 4–7 October 1998.

[205] M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull. Temporal error concealment using motion field interpolation. *IEE Electronics Letters*, 35(3):215–217, 4 February 1999.

[206] M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull. On the performance of temporal error concealment for long-term motion-compensated prediction. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume III, pages 376–379, Vancouver, 10–13 September 2000.

[207] M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull. Motion field interpolation for temporal error concealment. *IEE-Proceedings, Vision, Image and Signal Processing*, 147(5):445–453, October 2000.

[208] M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull. Multiple-reference temporal error concealment. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, volume V, pages 149–152, Sydney, 6–9 May 2001.

[209] A. Nosratinia. New kernels for fast mesh-based motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(1):40–51, January 2001.

[210] F. Pereira and T. Alpert. MPEG-4 video subjective test procedures and results. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):32–51, February 1997.

Index

A

Accuracy effect, 110
Accuracy problem, 98
AC prediction, 83
Additional supplemental enhancement information, H.263++ standard, 71
Additive color system, 11
Advanced INTRA coding mode, H.263+ standard, 59–61
Advanced prediction mode, H.263 standard, 55–56
Alpha planes, 76
Alternative INTER VLC mode, H.263+ standard, 68–69
Ambiguity problem, 98
Analog video, 9–12
Analog video signal, 9–11
Analysis stage, 34, 41
Analysis-synthesis coding, 41
Anisotropic nonstationary predictive coding, 37
Aperture problem, 96
Apparent motion, 94
Arbitrary slice ordering (ASO) mode, 62
Arithmetic coding, 26
ARQ. *See* Automatic repeat request
Aspect ratio, 10
Asynchronous transfer mode (ATM) networks, 46
Audiovisual objects (AVOs), 72
Automatic repeat request (ARQ), 225
AVOs. *See* Audiovisual objects

B

BABs. *See* Binary alpha blocks
Background mosaic, 85
Background sprite, 85
Backward motion estimation, 95
Backward node tracking, 130–131
Backward prediction, 39
Bandpass image, 33
Base layer, 64
Base-layer encoder, 86
Baseline profile, 71
Basic sprite coding, 85
BCH code, 211
BDM. *See* Block distortion measure
Best-match block, 40, 105
Bidirectionally-predicted-VOP (B-VOP), 80
Bidirectional prediction, 39–40
Binary alpha blocks (BABs), 77
Binary alpha plane, 76
Binary mask, 77
Binary shape coding, 76–79
Bit allocation, 32
Bitmap, 77
Bit-preserving methods, 19
Block distortion measure (BDM), 105, 161
Block matching, as optimization problem, 176–177
Block-matching algorithm (BMA), 105, 114–115, 118, 134
Block-matching methods, 105–117
  critique of, 139
  efficiency at very low bit rates, 122–124
Block-matching motion estimation (BMME), 40, 41, 105
  block size, 108–109
  matching function, 106–108
  overlapped motion compensation, 112–115
  reduced complexity, 175, 202
  search accuracy, 110–111
  search range, 109–110
  simplex minimization optimization for, 178–180

as two-dimensional constrained optimization problem, 176–177
unrestricted motion vectors, 111–112
Block-motion fields
long-term, 145–147
properties, 115–117
subsampled, 164–166
Blocks, 49
image warping, 126
reconstruction, 54
Block size, 108–109
Block-truncation coding, 38
BMA. *See* Block-matching algorithm
BMA-H algorithm, 122
BMA-HO algorithm, 135–139
BMME. *See* Block-matching motion estimation
Bose-Chaudhuri-Hocquenghem code. *See* BCH code
Boundary effect, 104
Brightness, 11
Brute-force search, 160
Burst errors, 207
B-VOP. *See* Bidirectionally-predicted-VOP

C

CAE. *See* Context-based arithmetic encoding
CCIR-601, 15–16, 18, 28
CCIR-721, 45
CCIR-723, 45
CCITT, 44
CD optimization. *See* Conjugate directions (CD) optimization method
CDS. *See* Conjugate-directions search
CGI. *See* Control grid interpolation
Channel encoder, 205, 206
Chroma components, 12
Chroma-keying, 63
Chroma subsampling, 14–15
Chrominance, 11, 12
CIF, 3, 16–17, 18
CMY system, 11
Codebook, 36
Codeword assigner, 19
Coding control block, 53
Coding efficiency, 2–3, 91
motion estimation, 91, 93–125
multiple-reference motion estimation, 141–155
warping-based motion estimation, 125–140
Coding-mode recovery, 224
Coding modes, H.263 standard, 50
Coding redundancy, 18
Color, chroma subsampling, 14–15
Color-difference components, 12
Color representation, analog video, 11–12
Common Intermediate Format. *See* CIF
Compensation, MPEG-4, 79–80
Compression, 17, 19, 48
performance measures of, 26
in vector quantization, 36
Computational complexity, 3, 157
reduced-complexity MFI, 235–236
reduced-complexity motion estimation, 157, 159–173, 175
simplex minimization search (SMS), 157, 175–202
Concealment displacement estimation, 221
Conditional replenishment (CR), 39
Conjugate directions (CD) optimization method, 177
Conjugate-directions search (CDS), 163
Content-based interactivity, 48
Context-based arithmetic encoding (CAE), 7
Continuous methods, warping-based motion estimation, 129
Contour/texture-oriented techniques, 37
Control grid interpolation (CGI), 133
Correspondence field, 95
CR. *See* Conditional replenishment
Critical decimation, 34
Cross-search algorithm (CSA), 163, 263–264
CSA. *See* Cross-search algorithm
Custom source formats, 58

D

Data partitioned slice mode, H.263++ standard, 70–71
Data partitioning, MPEG-4, 88, 218
DC prediction, 82–83
DCT. *See* Discrete cosine transform
Deblocking filter mode, H.263+, 61, 71
Decoder, 206, 219–220
Delta modulation (DM), 29
Dense motion field, 101, 102
Descriptions, 218
DF. *See* Displaced frame
DFA algorithm, 118

DFD. *See* Displaced-frame difference
Diamond search (DS), 163, 265
Differential methods, motion estimation (ME), 98–100
Differential pulse code modulation (DPCM), 29
Digital signal processors (DSPs), 3
Digital video, 13–17
  advantage over analog video, 13
  chroma subsampling, 14–15
  digitization, 13–14
  formats, 15–17
  quantization, 20–23
Digital video formats, 15–17
Digitization, 13–14
Directional decomposition coding, 37
Discontinuous methods, warping-based motion estimation, 129
Discrete cosine transform (DCT), 30–31, 40, 81
Discrete memoryless source (DMS), 19
Discrete wavelet transform (DWT), 35, 84
Displaced frame (DF), 4, 39
Displaced-frame difference (DFD), 5, 39, 40, 94
Displacement compensation, 221
Displacement field, 95
Displacement wrapping, 104
DM. *See* Delta modulation
DMS. *See* Discrete memoryless source
DPCM. *See* Differential pulse code modulation
DS. *See* Diamond search
DSPs. *See* Digital signal processors
DWT. *See* Discrete wavelet transform
Dynamic sprites, 142

E

ECVQ. *See* Entropy-constrained vector quantization
EDGE system. *See* Enhanced data rates for GSM evolution
Embedded zero-tree wavelet (EZW), 35
Encoder, 205
Energy compaction, 30
Enhanced data rates for GSM evolution (EDGE), 3
Enhanced reference picture selection (ERPS) mode, H.263++ standard, 70
Enhancement-layer encoder, 86
Enhancement layers, 64
Entropy, 19
Entropy-constrained quantizers, 21
Entropy-constrained vector quantization (ECVQ), 36
Entropy encoder, 24–26, 206, 213
Erasure errors, 207
EREC. *See* Error-resilience entropy code
ERPS mode. *See* Enhanced reference picture selection (ERPS) mode
Error concealment, 219–220
  coding-mode recovery, 224
  hybrid error concealment, 224
  motion-compensated concealment, 221
  with motion field interpolation (MFI), 231–257
  spatial error concealment, 220–221
  temporal error concealment, 221–223, 231–232
Error detection, 209–210
Error resilience, 2, 5, 203, 205
  forward techniques, 210–219, 229, 231
  interactive techniques, 224–228, 229
  MPEG-4, 86–89
  postprocessing (concealment) techniques, 219–224, 229
Error-resilience entropy code (EREC), 215–216
Errors. *See also* Error concealment; Error resilience
  detection of, 209–210
  effects of, 207–209
  spatial error propagation, 208
  temperal error propagation, 208–209
  types of, 206–207
Error surface, 17, 117
Exhaustive search, 160
Extended padding, 80
EZW. *See* Embedded zero-tree wavelet

F

Fast block-matching algorithms, 259–265
Fast Fourier transforms (FFTs), 104
Fast full-search techniques, reduced-complexity motion estimation, 168–170
FD. *See* Frame difference
FDIFF algorithm, 122
FEC. *See* Forward error correction
FFTs. *See* Fast Fourier transforms
Filtering, 13
Fixed-interval synchronization, 88, 213–214
Fixed-length coding (FLC), 24

Forward DCT transform, H.263 standard, 52–53
Forward error correction (FEC), 210, 211
Forward error-resilience techniques, 210–219, 229, 231
Forward motion estimation, 96
Forward node tracking, 130–131
Forward prediction, 39
Fourier transform (FT), 102
Fractal coding, 38
Frame difference (FD), 39
Frame differencing, 39
Frame rate, 10
Frame skip, 107
Frequency component method, 105
Frequency-domain methods, motion estimation (ME), 102–105
FS algorithm. *See* Full-search (FS) algorithm
FT. *See* Fourier transform
Full-search (FS) algorithm, 106, 160, 170–172, 187
Function surface, 177

G

Gamma-corrected components, 12
GA optimization. *See* Genetic algorithm (GA) optimization method
Generalized scalability, 85
General Packet Radio Service (GPRS), 3
Genetic algorithm (GA) optimization method, 17
Genetic motion search (GMS), 163
Global motion compensation (GMC), 142
Global System for Mobile (GSM), 2
GMC. *See* Global motion compensation
GMS. *See* Genetic motion search
GOBs. *See* Groups of blocks
GOV. *See* Group of video object planes
GPRS. *See* General Packet Radio Service
Gradient-based optimization, 100
Gradient methods, 99
Gray-scale alpha plane, 76, 79
Gray-scale shape coding, 79
Grid, 127
Group of video object planes (GOV), 72
Groups of blocks (GOBs), 49
GSM. *See* Global System for Mobile

H

H.26L standard, 48
H.120 standard, 44
H.261 standard, 44–45
H.263 decoder, temporal error concealment using MFI, 243–246
H.263 standard, 46–47, 49–57
  Annex D, 55
  Annex E, 55
  Annex F, 55–56
  Annex G, 56–57
  Annex H, 211
H.263+ standard, 47, 57–69
  Annex I, 59–61, 71
  Annex J, 61, 71
  Annex K, 61–62, 71
  Annex L, 62–63
  Annex M, 63–64
  Annex N, 64, 142
  Annex O, 64–67
  Annex P, 67
  Annex Q, 67–68
  Annex R, 212
  Annex S, 68–69
  Annex T, 69, 71
  modified Annex D, 58–59
  scalability, 218
H.263++ standard, 48, 70–72, 218
  Annex U, 70
  Annex V, 70–71
  Annex W, 71
  Annex X, 71–72
H.263-like codec, simplex minimization search (SMS) simulation results in, 188–190
HDTV systems, 17, 18, 160
Header extension code (HEC), 87, 211
HEC. *See* Header extension code
Hexagonal matching algorithm (HMA), 131–132
Hierarchical search techniques, reduced-complexity motion estimation, 166–168, 170–172
Hierarchical trees algorithm, 35
HMA algorithm. *See* Hexagonal matching algorithm
HME algorithm, 170–172
Horizontal retrace, 10
Hue, 11
Huffman coding, 24
Hybrid error concealment, 224
Hybrid MC-DPCM/DCT coding, 41

I

IDCT. *See* Inverse discrete cosine transform
IDCT mismatch error, 50
Image sequence, 9
Image warping, 126
Improved PB-frames mode, H.263+ standard, 63–64
Independent segment decoding (ISD) mode. 61
Independent segment decoding mode, H.263+ standard, 68, 212
Information theory, 19–20
Integral projections, 163
Interactive error-resilience techniques, 224–228, 229
Interframe coding, 38–42
Interlaced scanning, 10
Interleaved coding, 219
INTER macroblocks, 68
INTER mode, 50, 56
International Telecommunications Union. *See* ITU
Interpolation kernel, 233
INTRA algorithm, 122
INTRA block, 82
INTRA coding mode, 71
Intraframe coding, 28–37
  predictive coding, 28–29
  second-generation coding, 37
  subband coding, 33–35
  transform coding, 29–33
  vector quantization (VQ), 21, 35–37
INTRA macroblocks, 59, 212
INTRA mode, 50, 56
INTRA refresh, 212, 225–227
Intra-VOP (I-VOP), 80
Inverse discrete cosine transform (IDCT), 50, 54
Inverse quantization, H.263 standard, 54
Irreversible methods, 19
ISD mode. *See* Independent segment decoding (ISD) mode
Isolated test environment, 183–188
ITU-R, 15, 44
ITU-T, 44
I-VOP. *See* Intra-VOP

K

Karhunen-Loève transform (KLT), 30
Knowledge-based coding, 42

L

Lapped orthogonal transform (LOT), 33
Layered coding with prioritization, 217–218
LBG algorithm. *See* Linde-Buzo-Gray (LBG) algorithm
Least-mean-square (LMS) algorithm, 105
Levels
  H.263++, 71
  MPEG-4, 89
Linde-Buzo-Gray (LBG) algorithm, 36
Line-scanning technique, 235
Lloyd-Max quantizers, 21, 36
LMS algorithm. *See* Least-mean-square (LMS) algorithm
Local-operator-based techniques, 37
Long-term block-motion fields, 145–147
Long-term memory motion-compensated prediction (LTM-MCP), 142, 144–145, 202
Lossless methods, 19, 20
Lossy methods, 19
LOT. *See* Lapped orthogonal transform
Low-latency sprite coding, 85
Low-pass extrapolation (LPE), 81
LPE. *See* Low-pass extrapolation
LTM-MCP. *See* Long-term memory motion-compensated prediction
Luminance, 11, 12

M

Macroblocks (MBs), 49, 75
Main object type, 89
Main profile, 89
Mapper, 18, 206
MAP problem. *See* Maximum a posteriori probability (MAP) estimation problem
Markov-*K* random processes, 19
Markov random field (MRF), 97
Matching function, 106–108
Maximum a posteriori probability (MAP) estimation, 97, 221
Maximum likelihood (ML) estimator, 97
MBs. *See* Macroblocks
MC. *See* Motion compensation
MCP. *See* Motion-compensated prediction
ME. *See* Motion estimation
Mean pyramid, 166
Mean squared error (MSE), 27

Mesh, 127, 128–129
MFI. *See* Motion field interpolation
MHMC. *See* Multihypothesis motion compensation
Midprocessor, 86
Minimized maximum (MiniMax) error, 163
Mobile video communication, 9–21
  coding efficiency. *See* Coding efficiency
  computational complexity. *See* Computational complexity
  error resilience. *See* Error resilience
Model-based coding, 41–42
Modified Huffman code, 26
Modified quantization mode, H.263+ standard, 69, 71
Modified unrestricted vector mode, H.263+ standard, 58–59
Motion-based approach, 5
Motion-compensated coding, 39–41
Motion-compensated concealment, 221
Motion-compensated prediction (MCP), 4, 39
Motion compensation (MC), 4, 39, 93–94, 125
  H.263 standard, 54
  multihypothesis motion compensation, 114, 232, 252–255
  multiple-reference motion compensation, 142, 143
  overlapped motion compensation (OMC), 112–115
  warping-based motion estimation, 132–133
Motion encoding, 75
Motion estimation (ME), 5, 39. *See also* Block-matching motion estimation (BMME)
  backward motion estimation, 95
  Bayesian model, 97
  coding efficiency, 91, 93–125
  comparative study of algorithms, 117–121
  defined, 125
  deterministic model, 97
  differential methods, 98–100
  forward motion estimation, 96
  frequency-domain methods, 102–105
  H.263 standard, 51–52
  ill-posed problem, 96–97
  maximum a posteriori probability (MAP) estimation problem, 97
  motion compensation (MC), 4, 39, 54, 93–94, 112–114, 125
  motion models, 4, 97–98
  MPEG-4, 79–80
  multiple-reference motion estimation, 141–155, 160
  nonparametric model, 98
  numerical methods, 100–101
  overlapped motion compensation, 112–115
  parametric model, 97–98
  pel-recursive methods, 100–102
  probabilistic model, 97
  reduced-complexity motion estimation, 159–173
  region of support, 98
  steepest-descent methods, 100–101
  two-dimensional motion estimation, 94
  warping-based motion estimation, 125–140
Motion field, 95
Motion field interpolation (MFI), 233
  MFI-MH, 253
  reduced-complexity MFI, 235–236
  temporal error concealment using combined BM-MFI, 236–237
Motion information recovery, 222
Motion models, 4, 97–98
Motion overhead, warping-based motion estimation, 133–134
Motion parameters, 97
Motion vector, 40, 78
Motion vector coding/decoding, H.263 standard, 52, 54
Motion vector difference (MVD), 78
MPEG-1, 45–46
MPEG-2, 46, 211
MPEG-4, 47–48, 72–89
  compensation, 79–80
  data partitioning, 88, 218
  error resilience, 86–89
  fixed-interval synchronization, 213–214
  levels, 89
  motion estimation (ME), 79–80
  MR-MCP techniques, 142
  object-based representation, 72–76
  padding, 80, 81, 194
  profiles, 89
  quantization, 81–82
  scalability, 85–86, 218
  scanning, 83
  shape coding, 76–79
  sprite coding, 84–85
  still-texture coding, 84
  texture coding, 75, 81–84

variable-length coding, 83–84
MPEG-4 codec, simplex minimization search (SMS) simulation results in, 190–196
MR-3DSM algorithm, 197–198
MR-FS/SMS algorithm, 197
MR-MCP. *See* Multiple-reference motion-compensated prediction
MR-SMS algorithm, 197
MSE. *See* Mean squared error
Multicopy retransmission, 225
Multiframe memory, 141, 142
Multihypothesis motion compensation (MHMC), 114, 232, 252–255
Multilayer scalability, 67
Multiple description coding, 218–219
Multiple-reference encoder, 151
Multiple-reference motion-compensated prediction (MR-MCP), 64, 141, 142, 175
Multiple-reference motion estimation, 141–155, 160
efficiency at very low bit rates, 149–154
long-term memory motion-compensated prediction (LTM-MCP), 142, 144–145, 202
prediction gain, 147–149
properties of long-term block-motion fields, 145–147
simplex minimization search (SMS) for, 196–201
Multiple-reference rate-constrained encoder, 151, 153
Multiplexer, 86
Multiresolution coding, 38
MVD. *See* Motion vector difference

N

NACK message. *See* Negative acknowledgment (NACK) message
National Television System Committee. *See* NTSC
Natural video objects, 72
NCCF. *See* Normalized cross-correlation function
Negative acknowledgment (NACK) message, 227
Neural-network-based coding, 38
Node points, 127
Node tracking, 130–131
Node-tracking algorithm, 131–132
Nondegenerate simplex, 178
Noninterlaced scanning, 10
Nonparametric model, motion estimation (ME), 98
Normalized cross-correlation function (NCCF), 106
N quantization intervals, 21
N-steps search (NSS), 261–262
NTSC, 11, 12
Numerical methods, motion estimation (ME), 100–101
Nyquist rate, 14

O

Object-based coding, 42
Object-based representation, MPEG-4 standard, 72–76
Occlusion problem, 96
OMC. *See* Overlapped motion compensation
One-at-a-time search (OTS), 163, 262–263
One-bit/pixel, 163
One-dimensional binary logarithmic search, 17
Opaque BAB, 77
Optical flow field, 95
Optimization methods, BMME algorithms based on, 17
OTS. *See* One-at-a-time search
Overlapped motion compensation (OMC), 55–56, 112–115

P

Padding, MPEG-4, 80, 81, 194
PAL, 11, 12
Parametric model, motion estimation (ME), 97–98
Partial distortion elimination (PDE) algorithm, 168, 170–172
Patches, 128
PB-frames, 58
PB-frames mode, H.263 standard, 56–57
PCA algorithm, 118
PDC. *See* Pel difference classification
PDE algorithm. *See* Partial distortion elimination (PDE) algorithm
Peak signal-to-noise ratio (PSNR), 27
Pel, 14

Pel difference classification (PDC), 163
Pel-recursive methods, motion estimation (ME), 100–102, 118–121
Phase Alternation Line. *See* PAL
Picture element, 14
Picture freeze mode, 62
Picture freeze with resizing mode, 62–63
Picture snapshot mode, 63
Pixel, 14
PLUSPTYPE, 58–59
Polygon matching, 80, 194
Postprocessing error-resilience techniques, 219–224, 229
Postprocessor, 86
PRA algorithm, 118–121
PRA-C algorithm, 119
Predicted-VOP (P-VOP), 80
Prediction gain, LTM-MCP, 147–149
Predictive coding, 28–29, 37
Profiles
  H.263++, 71
  MPEG-4, 89
Progressive refinement segment mode, 63
Progressive scanning, 10
Projected motion, 94
PSNR. *See* Peak signal-to-noise ratio
Psychovisual redundancy, 18
P-VOP. *See* Predicted-VOP
Pyramidal coding, 37

Q

QCIF, 3, 18
QMF. *See* Quadrature mirror filter
QSIF, 16
Quadrature mirror filter (QMF), 34
Quadrilateral patches, 128
Quad-tree coding, 38
Quantization, 20–23
  entropy-constrained vector quantization (ECVQ), 36
  H.263 standard, 53
  MPEG-4, 81–82
  scalar quantization, 20–21
  uniform quantization, 21–23
  vector quantization (VQ), 21, 35–37
Quantizer, 19, 206
Quantizer step size, 21

R

Random bit errors, 207
Raster scanning, 10
Ratd-distortion function, 20
Rate-distortion theory, 20
RBMAD. *See* Reduced-bits mean absolute difference
Reconstruction level, 21
Rectangular (RS) submode, 62
Recursive coding, 38
Reduced-bits mean absolute difference (RBMAD), 163
Reduced complexity block-matching motion estimation (BMME), 175
Reduced-complexity MFI, 235–236
Reduced-complexity motion estimation, 159–173, 175, 202
  fast full-search techniques, 168–170
  hierarchical search techniques, 166–168, 170–172
  need for, 159–161
  subsampled block-motion field, 163, 164–166
  two-dimensional logarithmic (TDL) search, 161–163
Reduced-resolution update mode, H.263+ standard, 67–68
Redundancy, 18, 24
Reference picture resampling mode, H.263+ standard, 67
Reference picture selection (RPS), 64, 142, 227–228
Refresh rate, 10–11
Region-growing, 37
Region of support, motion estimation (ME), 98
Repetitive padding, 80, 194
Restricted motion vectors, 111
Resynchronization, MPEG-4, 86–87
Resynchronization codewords, 87, 213–214, 291
Retrace, 10
Retransmission without waiting, 225
Reversible methods, 19
Reversible variable-length coding (RVLC), 59, 216–217
RGB system, 11
Robust entropy coding, 213–217
Robust waveform coding, 211–213
RPS. *See* Reference picture selection
Run-length encoding, 24
RVLC. *See* Reversible variable-length coding

S

Saturation, 11
Scalability, 64–67, 85–86, 218
Scalability pictures, 58
Scalability pre-/postprocessor, 86
Scalable shape coding, 79
Scalable sprite coding, 85
Scalar quantization, 20–21
Scanning, 10, 83
SDM algorithm, 170–172
SEA. *See* Successive elimination algorithm
Search range, 109–110
Search window, 40
SECAM, 11, 12
Second-generation coding, 37
Segmented coding, 37
Segmented decoding mode, 212
Selective recovery technique, 227
Semantic-based coding, 42
Sequential Coleur Avec Memoire. *See* SECAM
Shape coding, MPEG-4, 76–79
Shape encoding, 75
Short-term frame memory/long-term frame memory (STFM/LTFM), 142
Side-match distortion (SMD), 223
SIF. *See* Source Input Format
Simple profile, 89
Simplex, 177–178, 181
Simplex minimization (SM) algorithm, 177–178
Simplex minimization (SM) optimization, 177–180
Simplex minimization search (SMS), 157, 175–202
  constraints on independent variable, 182
  in H.263-like codec, 188–190
  initialization procedure, 181–182
  in isolated test environment, 183–188
  in MPEG-4 codec, 190–196
  motion vector refinement, 183
  for multiple-reference motion estimation, 196–201
  simplex minimization optimization, 177–180
  termination criterion, 182–183
Simplex minimization search (SMS) algorithm, 181
Single-reference encoder, 149, 150
Single-reference rate-constrained encoder, 150–151
Slice, 62
Sliced structured mode, H.263+ standard, 61–62, 71
Sliding-window control method, 144
SM algorithm. *See* Simplex minimization algorithm
SMD. *See* Side match distortion
SMF algorithm, 170–172
SMS. *See* Simplex minimization search
SMS algorithm. *See* Simplex minimization search (SMS) algorithm
SNR scalability, 65–67, 218
Source coding theorem, 20
Source encoder, 205
Source Input Format (SIF), 16
Spatial displacement, long-term memory, 145
Spatial error concealment, 220–221
Spatial error propagation, 208
Spatial redundancy reduction, 28
Spatial scalability, 66, 67, 218
Spatial transformation, 129
Spatial transformation functions, 126
Spectral leakage, 104
SPIHT algorithm. *See* Hierarchical trees algorithm
Split-and-merge, 37
Sprite, 84–85
Sprite coding, MPEG-4, 84–85, 142
SQCIF. *See* Sub-QCIF
SSD. *See* Sum of squared differences
Statistical redundancy, 18
Steepest-descent methods, 100–101
STFM/LTFM. *See* Short-term frame memory/long-term frame memory
Still image, 9
Still-image coding, 28
Still-texture coding, 84
Subband coding, 33–35
Subpictures, 70
Sub-QCIF (SQCIF), 3
Subsampling, 14, 164–166
Subtractive color system, 11
Successive elimination algorithm (SEA), 168–169
Sum of squared differences (SSD), 107
Supplemental enhancement information mode, H.263+ standard, 62–63
Symbol encoder, 19, 206
Symbol encoding, 23–26

Syntax-based arithmetic coding mode, H.263 standard, 55
Synthesis stage, 34, 41, 125–126

T

TDL search. *See* Two-dimensional logarithmic (TDL) search
Telenor H.263, 209
Television, 10, 11, 15
Temporal displacement, 142, 145–146
Temporal error concealment, 5, 221–223, 231–232
  with combined BM-MFI, 236–237
  within multiple-reference video codec, 247–255
Temporal redundancy reduction, 38
Temporal replacement (TR), 223, 247
Temporal scalability, 65, 66, 217
Texture coding, MPEG-4, 75, 81–84
Texture mapping, 126
Three-dimensional coding, 38
Three-step search (TSS), 163, 261
Threshold coding, 32–33
TR. *See* Temporal replacement
Transform coding, 29–33
Transformer, 18
Translational model, 40
Transparent BAB, 77
Triangular patches, 128
Trichromatic theory of color, 11
TSS. *See* Three-step search
Two-dimensional constrained minimization problem, 175
Two-dimensional logarithmic (TDL) search, 161–163, 170–172, 269–270
Two-dimensional motion estimation, 94

U

UMTS. *See* Universal Mobile Telecommunication System
Unequal error protection, 217
Uniform quantization, 21–23
Unimodal error surface assumption, 161, 175, 202
Unitary transform, 30
Universal access, 48
Universal Mobile Telecommunication System (UMTS), 3
Unrestricted motion vector mode, H.263 standard, 55
Unrestricted motion vectors, 111–112

V

Variable-length coding (VLC), 24–26
  MPEG-4, 83–84, 88–89
Vector quantization (VQ), 21, 35–37
Velocity field, 95
Version 2 Interactive and Streaming Wireless Profile, 71
Vertical resolution, 10
Vertical retrace, 10
Video. *See also* Video coding; Video coding standards
  color representation, 11–12
  defined, 9
Video coding, 9–21. *See also* Video coding standards
  analog video, 9–12
  analysis-synthesis coding, 41
  anisotropic nonstationary predictive coding, 37
  arithmetic coding, 26
  basics of, 17–21
  block-truncation coding, 38
  digital video, 13–17
  directional decomposition coding, 37
  entropy encoding, 24–26
  fractal coding, 38
  Huffman coding, 24
  hybrid MC-DPCM/DCT coding, 41
  interframe coding, 38–42
  interleaved coding, 219
  intraframe coding, 28–37
  knowledge-based coding, 42
  model-based coding, 41–42
  motion-compensated coding, 39–41
  multiple description coding, 218–219
  multiresolution coding, 38
  neural-network-based coding, 38
  object-based coding, 42
  performance measures, 26–28
  predictive coding, 28–29
  pyramidal coding, 37
  quad-tree coding, 38
  quantization, 20–23

recursive coding, 38
redundancy, 18
reversible variable-length coding (RVLC), 59
run-length encoding, 24
second-generation coding, 37
segmented coding, 37
semantic-based coding, 42
subband coding, 33–35
symbol encoding, 23–26
three-dimensional coding, 38
threshold coding, 32–33
transform coding, 29–33
variable-length coding (VLC), 24–26
vector quantization (VQ), 21, 35–37
waveform coding, 37
zonal coding, 32
Video coding standards, 43–89
CCIR-721, 45
CCIR-723, 45
H.26L, 48
H.120, 44
H.261, 44–45
H.263, 46–47, 49–57
H.263+, 47, 57–69
H.263++, 48, 70–72
MPEG-1, 45–46
MPEG-2, 46
MPEG-4, 47–48, 72–89
Video communication system, components, 205–206
Video object layers (VOLs), 72
Video object planes (VOPs), 72–75, 86
Video objects (VOs), 72
Video time segment mode, 63
VLC. *See* Variable-length coding
VOLs. *See* Video object layers
VOPs. *See* Video object planes
VOs. *See* Video objects
VQ. *See* Vector quantization

W

Warp, 127
Warping-based algorithm, 135
Warping-based motion estimation, 125–140
backward vs. forward node tracking, 130
continuous vs. discontinuous methods, 129
efficiency at very low bit rates, 134–139
motion compensation method, 132–133
node-tracking algorithm, 131–132
spatial transformation, 126, 129
transmitted motion overhead, 133–134
Waveform coding, 37
Waveform encoder, 206, 211
WBA algorithm, 135–137

Z

Zonal coding, 32