MOTION-ADAPTIVE INTRAFRAME TRANSFORM CODING OF VIDEO SIGNALS

by P.H.N. DE WITH

Philips Research Laboratories, P.O. Box 80000, 5600 JA Eindhoven, The Netherlands

Abstract
Spatial transform coding has been widely applied for image compression because of its high coding efficiency. However, in many intraframe systems, in which every TV frame is independently processed, coding of moving objects in the case of interlaced input signals is not addressed. In this paper, we extend intraframe transform coding techniques for interlaced video signals with limited additional complexity. After discussing key aspects of an interlaced video signal, we present a simple motion-detection scheme which is suitable for an a priori block-coding decision, thereby saving hardware complexity. The transformer is modified to obtain either intraframe or intrafield transformed blocks by performing—partially—different fast cosine computation algorithms. As a result, the subjective image quality of the motion-adaptive coding technique is considerably improved as compared with the nonadaptive system.

Keywords: discrete cosine transform, encoding, frame sequence, Hadamard transform, motion, video signals

1. Introduction
Video data compression, by which the high initial bit rate of the digitized video signal can be reduced substantially, is a key aspect for realizing a digital consumer video recorder with acceptable playing time and recording mechanics of modest complexity. However, the application of compression techniques in digital recording systems is constrained by limitations imposed by the recording system itself. Firstly, the magnetic recording channel has a poor performance compared with, for example, an optical fibre transmission channel, so that the error sensitivity of a coding algorithm should be considered carefully. Secondly, playing back in multispeed mode—without loss of picture quality—requires that information blocks are decoded independently of each other. This constraint hampers the application of
interframe coding for recording systems, since decoding in interframe techniques depends on the previously coded frame(s). Furthermore, in the portable applications of video recording, the system designer has to reconcile conflicting constraints of a small cassette and a sufficiently long playing time. At the present stage of recording technology, this only is feasible if the bit rate to be recorded is as low as possible, which calls for highly efficient video coding algorithms.

Predictive coding and transform coding can be considered as two main classes in bit rate reduction techniques, although the diversity of algorithms has increased substantially in recent years. From these main classes, block transform coding techniques provide very effective image compression. Numerous transforms have been proposed for bit-rate reduction, such as the Hadamard, Haar, slant, sine and cosine transforms, among which the first and the last have been considered most extensively. The Hadamard transform is particularly interesting because of its simplicity for realization, whereas the discrete cosine transform (DCT) is very efficient in signal energy compaction at the expense, however, of increased complexity. Advanced DCT-based algorithms for speech and image coding applications have been widely investigated. Interest in employing such advanced coding techniques in more practical situations has increased with the continuous growth in VLSI technology. Recently, DCT coding experiments have been reported for still picture transmission, video conferencing and digital video recording. For recording, alternative block coding techniques with less complexity, such as Hadamard transform coding and adaptive dynamic range coding, have also been employed.

In an earlier publication, we have reported results on design aspects for the transformer itself and we have dealt with a bit assignment technique which is suitable for efficient error protection. In this paper we discuss intraframe coding of interlaced video signals. A video frame may be a combination of two interlaced fields which have been sampled at different time instants. For this reason, an interlaced video frame contains locally specific data structures around and/or in moving objects, which are usually difficult to code. We present a locally motion-adaptive system which has a better subjective image quality than the nonadaptive system.

Section 2 gives a general outline of the total coding algorithm by briefly identifying each processing block and discussing its function. Parts of the scheme that are especially relevant for motion adaptivity, the transformer and the motion detector, are dealt with in more detail in the subsequent sections. Key aspects for intraframe coding of interlaced video are discussed in Sec. 3. A simple motion detection technique is presented in Sec. 4. The application
Motion-adaptive coding of video signals

Fig. 1. Motion-adaptive DCT encoder block diagram.

of a partial transform of the input sample block provides a hardware-efficient method for an a priori coding decision. Section 5 treats the modifications required for merging motion adaptivity with the transform calculation. Section 6 addresses the bit assignment technique applied in the algorithm, which is based on variable-length coding. A performance comparison between the adaptive and the non-adaptive system is made in Sec. 7, where the results of computer simulations of both algorithms are discussed.

2. System description

The coding technique which is investigated may best be identified as an intraframe DCT threshold coding algorithm. Basically, video data compression is obtained in two steps. At the start, an orthogonal transform (DCT) is carried out on small two-dimensional blocks within a frame. The result after transformation is a block of spectral components (in the following, termed coefficients) with the signal energy concentrated in an—on average—limited number of components. In the second step, quantization and variable-length coding of coefficients are employed for obtaining sufficient data reduction and high efficiency coding.

Figure 1 portrays a block scheme of the motion-adaptive encoder. The frame memories for luminance (Y) and chrominance (U and V) gather two fields to construct one video frame, which is subsequently segmented into small blocks of 8 x 8 samples. The luminance and chrominance blocks are multiplexed into one data stream, thereby saving hardware, and improving overall coding efficiency because chrominance blocks usually require fewer
bits for coding. A motion detector analyses every incoming sample block, and governs the transform computation and—partially—the subsequent processing blocks, so that motion-adaptive processing is achieved. A block-adaptive intraframe—intrafield DCT is carried out, which is explained in Sec. 5. After reordering (scanning), the coefficients are weighted individually to obtain frequency-dependent quantization. In addition, the quantization adapts to local picture statistics. Variable-length coding of the significant coefficients is performed in three steps. First, the coefficient amplitudes are ranked by their magnitude and the difference between successive amplitudes is constructed. Secondly, the address information, required to indicate the place of a coefficient inside a block, is minimized. Finally, variable-length codes are assigned to both coefficient differences and addresses. A buffer in combination with a control unit smooths the variable bit rate output and provides the required interface to the subsequent fixed-rate processing blocks in a recorder, such as error correction coding and/or channel modulation.

3. Motion adaptivity

In the following, we concentrate on the characteristics of an interlaced signal, and we discuss motion-adaptive system architectures for coding.

To understand the problem of motion we have first to specify the signal which has to be coded. For this reason, we distinguish a video frame and a general image. In the literature, an *image* is often assumed to be a sequentially scanned rectangular array of samples, e.g. $512 \times 512$, whereas video *frames* may be a combination of two interlaced fields which have been sampled at different time instants. The time interval in between is long enough to permit objects in fields to be moved over a significant number of samples in an arbitrary spatial direction. In an intraframe transform coding system for video signals, two fields are combined into one frame prior to coding. As a result, the time definition of vertically adjacent samples differs by one field period. An example of the resulting data structures is shown in figs 2 and 3. Figure 2 is a photograph of a video frame extracted from a scene in which the gate in the left-hand part of the picture is moving from left to right. Inside this area we have indicated—framed in black—an $8 \times 8$ block of samples on a regular grid basis. The contents of the sample block are shown in an enlarged view in fig. 3a). It is clearly visible that the adjacent video lines are stemming from different time instants. Furthermore, one may notice the fairly odd data structure for coding.
It is emphasized here that motion reduces the local coding efficiency of an intraframe compression technique. Different system architectures may be used to tackle this efficiency problem. Since, eventually, the number of bits which is required for coding—at a certain quality—a block containing motion is the best decision criterion, the optimum architecture would be to have two encoders in parallel and to make the decision for minimum bit cost after bit assignment (an a posteriori coding decision). This solution has been proposed for simpler techniques, such as the Hadamard transform\textsuperscript{13}). However, an a posteriori decision is an expensive solution for implementation in the case of advanced coding algorithms, such as DCT with variable-length coding. In this paper we propose an a priori decision for motion-adaptive coding. Since the decision is made prior to coding, it yields a single-branch coding system, thereby substantially saving hardware. A comparable decision scheme has been presented for dynamic range coding\textsuperscript{12}). Finally, the simplest solution for coding of interlaced video signals is to apply intrafield coding\textsuperscript{14,15}), so that coding of motion-generated data structures is circumvented.
Fig. 3. Enlarged view of the block from fig. 2 on a) a frame basis and b) a separate field basis.
Unfortunately, the overall performance of intrafield coding is considerably worse than with intraframe coding, and it is therefore not considered here.

Now we concentrate on the type of data structure in the spectral domain generated by motion within a sample block. Consider a part of an object, with luminance value \( A \), in front of a background, with luminance value \( B \). Two examples in the sample domain are shown in figs 4a) and 4b). In figs 4c) and 4d) the object started to move downwards and to the left respectively. In figs 4e) and 4f) the corresponding coefficients resulting from the cosine transformation are shown. It can be noticed that motion in the vertical direction does not result in a substantial increase in vertical frequency components: without motion, nonzero components occur in the same column. However, motion in the horizontal direction results in additional components in the vertical and diagonal frequencies. This can be understood by recalling the basis functions of the DCT \(^{16}\), namely \( \cos[(2i+1)u\pi/2N] \), where \( i \) and \( u \) are sample and frequency index \((0 \leq u, i \leq N - 1)\) respectively. The sample block structure of fig. 4d) contains columns of the form ABAB..., which contribute to the energy of the highest vertical frequency \( u = N - 1 \). However, the ac energy of the input waveform is not completely compacted into the frequency \( u = N - 1 \), since the basis function of the DCT has a reduced amplitude at the borders of the block because the envelope of the basis function has a cosine shape. The missing part of the signal energy is transformed into coefficients of lower frequencies, i.e. \( u = N - k, 2 \leq k \leq N - 1 \). For this reason, a large variety of nonzero coefficients could be expected in the case of transformation of the data in fig. 4d). From these observations, it can be concluded that measuring high frequency energy in the vertical direction makes sense for motion detection. In addition, we conclude that the DCT coefficient pattern which results from motion-generated data does not lend itself to motion detection because of the insufficient compaction into a limited number of coefficients.

For the required coding adaptivity, once motion has been detected, it is useful to split the frame-based block again into two field-based blocks \(^{17}\). By doing so, high frequency energy is transferred to low frequency energy, as illustrated in the following example.

A block containing data due to motion has sample columns \( f(i) \) which are modelled to the square wave ABAB..., the latter being rewritten to

\[
  f(i) = \frac{A + B}{2} + (-1)^i \frac{A - B}{2}, \quad 0 \leq i \leq N - 1. \tag{1}
\]

For intrafield transformation, a split between even \( l(i) \) and odd samples \( k(i) \)

---

**Philips Journal of Research Vol. 44 Nos 2/3 1989**

351
in the column is made; hence

\[ l(i) = f(2i), \]

\[ k(i) = f(2i + 1), \quad 0 \leq i \leq N/2 - 1. \]
The values for the sample columns in the intrafield blocks are found by substituting eq. (1) in eq. (2):

\[ l(i) = (A + B)/2 + (-1)^{2i}[(A - B)/2], \]
\[ k(i) = (A + B)/2 + (-1)^{2i+1}[(A - B)/2], \quad 0 \leq i \leq N/2 - 1. \] (3)

Because evidently the dependence on the index \( i \) in the right-hand terms in eq. (3) vanishes, the varying terms become constant and are added to the left-hand term, so \( l(i) = A \) and \( k(i) = B \), as could be expected. We have presented this example because in practice the principle still holds, although the signal is more complex so that generally signal energy is also present at frequencies in between.

From the previous example we notice that, by taking intrafield blocks, the data are vertically subsampled so that the highest vertical frequencies, which contain a substantial amount of motion information, are folded back to zero frequency. This phenomenon improves the coding efficiency substantially. In fact the data to be transformed resemble the case in which there is no motion. A photograph of intrafield blocks is shown in fig. 3b). It can be noticed that the complex structure of fig. 3a) is replaced by two relatively simple structures only containing sample transitions in the horizontal direction. It can easily be verified that intrafield transformation of the pattern in fig. 4d) would result in two rows of nonzero coefficients in the transform domain. This is significantly less information for coding than the amount of nonzero components to be coded of the data block shown in fig. 4f).

4. Motion detection

In this section we provide simple means for motion detection. The purpose of the motion detector is to determine—prior to coding—whether the actual block which has to be coded contains data structures formed by a local motion component.

The heuristic arguments of Sec. 3 provide evidence that motion detection can be based on vertical high frequency energy measurement. Let us concentrate on the meaning of eq. (1) from the previous section. A square-wave function is split into two parts: the left-hand term in eq. (1) represents a dc component of the signal, while the right-hand term is the ac part. In fact, we performed a transform on the signal from which the ac part can be split into the coefficient (expression between brackets) and the basis function \((-1)^i\). This basis function equals the basis function of the Hadamard transform,
which has the highest number of sign transitions. In other words, we are able to detect structures due to motion by using a partial transform (a single Hadamard coefficient) in the vertical direction. A simplified Hadamard transform for motion analysis has been applied in other systems, e.g. for analysing interframe differences in a conditional-replenishment coding algorithm\(^{18}\).

The final motion decision is based on a two-step calculation, which has been visualized in fig. 5. First, for each column the aforesaid Hadamard coefficient is calculated and, secondly, the magnitudes of the coefficients obtained are summed and compared with a predetermined threshold. If the result of the summation exceeds the threshold, motion-dedicated coding is chosen. We now summarize the detection scheme in a more formal description.

For every column \(j\) of the sample block the Hadamard coefficient, which is associated with the highest number of sign transitions in the basis function, denoted by \(H_N^{(j)}\), is calculated with

\[
H_N^{(j)} = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^j f(i, j). \tag{4}
\]

In eq. (4), \(f(i, j)\) represents the samples of the input block and \(N\) denotes the block size (typically \(N = 8\)). We decide for motion if

\[
\sum_{j=0}^{N-1} |H_N^{(j)}| > T, \tag{5}
\]

in which \(T\) denotes the motion threshold.
The performance of the detector may be illustrated by the picture in fig. 6. All the blocks which have a coefficient sum above the threshold are indicated by a black luminance value (fig. 2 can be used for comparison). Threshold optimization has been performed experimentally. We have found the best performance by turning the threshold in such a way that the motion decision was consistent in areas with high motion (e.g. the moving gate on the left-hand side). A high threshold causes blocks in high motion areas to be coded as if there were no motion, which results in well-visible blocks owing to increased coding noise inside areas with good signal-to-noise ratio. It can be seen that misdetections also occur, for example in the background or on vertical edges. Misdetections can be reduced by adding more complexity, but this appears not to be really required since the number of misdetections is small and, moreover, we have found that the effect of the misdetections is scarcely noticeable at bit rates giving a good picture quality. This observation is explained by the fact that a few less efficiently coded blocks do not harm the

Fig. 6. Motion blocks (dark luminance) in a moving scene.
total coder performance.

5. Transform calculation

For proper motion-adaptive processing two subjects have to be dealt with: motion detection and the subsequent coding adaptivity. This section focuses on the coding adaptivity in the transformer, assuming that the motion decision has been made. In Sec. 3 we noticed that intrafield transformation in the case of motion minimizes the number of nonzero components for coding. In the following, we show that adaptive intrafield–intraframe transformation can be accomplished by controlling the computation in the vertical direction.

The DCT is a separable transform, which means that first the rows of a block of picture elements are transformed by a one-dimensional DCT and, subsequently, the columns of the result obtained. It is well known that the complexity of the computation of the DCT can be substantially reduced by applying a fast algorithm, similar to a fast Fourier transform technique. We have studied several algorithms for implementation and have considered calculation accuracy aspects as well. As a result of this study we have designed a new fast algorithm which is not optimum with respect to the amount of operations but which has a satisfactory calculation accuracy. This fast algorithm can be obtained by subsequently adding pairs of terms that will be multiplied by the same cosine terms. Thereafter, all multiplications are performed and some results have to be accumulated. Figure 7 shows a flowgraph of this fast algorithm. The input samples are denoted by \( f(i) \), \( 0 \leq i \leq 7 \), and the corresponding coefficients at the output by \( F(u) \), \( 0 \leq u \leq 7 \). The circles in which the terms \( c_m^u \) are indicated represent multiplications by a factor of \( \cos(n\pi/m) \).

As an intrafield transform requires separate processing of odd and even lines, it suffices—because of the separability of the DCT—to modify the computation in the vertical direction only. The block diagram for a motion-adaptive two-dimensional transformation is exhibited in fig. 8. Two processors (HDCT and VDCT) are required, for the horizontal and vertical transforms respectively, with a memory (TM) for matrix transposition in between. The first processor performs a calculation according to the computation diagram as depicted in fig. 7. The second processor carries out either the same algorithm (H: normal mode; equal to HDCT), or a modified calculation in the case of motion (M: motion mode). The output of the motion detector (MD) governs the choice between the algorithms.

We now define the modified algorithm for the second processor in the case of motion. The two-dimensional definition of the DCT is given in ref. 16.
Fig. 7. Eight-point DCT fast algorithm.

Fig. 8. Motion-adaptive two-dimensional transformer.
For a separable computation, we rewrite the definition to:

\[
F(u, v) = \frac{2}{N} C(u) \sum_{i=0}^{N-1} \left\{ \frac{2}{N} C(v) \sum_{j=0}^{N-1} f(i, j) \cos \left[ \frac{(2j+1)\nu \pi}{2N} \right] \right\} \cos \left[ \frac{(2i+1)\nu \pi}{2N} \right],
\]

which is valid for \(0 \leq u, v \leq N - 1\). The input samples are denoted by \(f(i, j)\), with \(0 \leq i, j \leq N - 1\), and the normalization variable \(C(w)\) equals unity for \(w > 0\) and \(1/\sqrt{2}\) for \(w = 0\). The part of eq. (6) in braces is a one-dimensional transform \(F_h(i, v)\), which is carried out for the \(i\)th row of the matrix \(f(i, j)\), and from which the computation can be reduced to the diagram depicted in fig. 7. In the case of motion, the calculation outside the braces is modified to an intrafield algorithm, in which we perform separate transforms on even and odd samples, labelled a and b respectively. The (double) intrafield transformation results from

\[
F_a(u, v) = \frac{4}{N} C(u) \sum_{i=0}^{N/2-1} F_h(2i, v) \cos \left[ \frac{(2i+1)\nu \pi}{N} \right], \quad 0 \leq u \leq \frac{N}{2} - 1,
\]

\[
F_b(u, v) = \frac{4}{N} C(u) \sum_{i=0}^{N/2-1} F_h(2i+1, v) \cos \left[ \frac{(2i+1)\nu \pi}{N} \right], \quad 0 \leq u \leq \frac{N}{2} - 1.
\]

The coefficients \(F_a(u, v)\) and \(F_b(u, v)\) represent the coefficients from the block in field a and field b respectively. Note that, by applying eqs (7), an intrafield transform is performed twice with block size \(N/2\). The modified vertical transform can also be calculated by means of a fast algorithm by applying similar techniques to those in the horizontal computation. The calculation diagram obtained for the vertical computation in the case of motion is shown in fig. 9. Since an intrafield computation is carried out the operations for the odd and even samples are completely separated as can be noticed. The output coefficients \(F_a(u, v)\) and \(F_b(u, v)\), as shown in fig. 9, correspond to definition (7).

The inverse discrete cosine transform can be obtained by simply reversing the flowgraph of the algorithm because the DCT is an orthogonal transform.

6. Bit assignment

In this section we discuss briefly the bit assignment technique used in our experiments. The reader is referred to ref. 19 for more details. We opted for a threshold coding algorithm because it adapts intrinsically to varying data structures. As a result, threshold coding leads to good image qualities, even
Motion-adaptive coding of video signals

at low bit rates, although at the cost of increased complexity compared with other techniques such as zonal coding 5).

In the first step coefficients are ranked by their magnitude (sorting). Instead of separately coding the ranked amplitudes, an amplitude difference signal is constructed, which is the difference between the previous amplitude selected and the actual amplitude. This difference signal is subsequently variable-length encoded. As the coefficients are ranked by their magnitude, the spatial coordinates are mixed up so that information about the position of the coefficients (in the following, termed the address) has to be stored also. Two cases can be distinguished for address coding:

- amplitude difference signal non-zero: in this case absolute addressing is applied, in which address counting starts from the first coefficient of the actual block;
- amplitude difference signal zero: now relative addressing is carried out starting from the address of the previously coded coefficient.

Fig. 9. Vertical DCT computation diagram in the case of motion.
The efficiency of the address coding procedure is further improved by taking into account the number of already coded coefficients (see step 3 in the algorithm below). A formal description of the algorithm now follows.

The two-dimensional block of coefficients \(F(u, v)\), with \(0 \leq u, v \leq N - 1\), is scanned prior to coding, which results in a serial block of coefficients. Each coefficient after scanning, denoted by \(F(A)\), is uniquely described by its amplitude \(F\) and its address \(A\) (ref. 19). The absolute address \(A\) \((0 \leq A \leq N^2 - 1)\) of a coefficient is defined as the ranking number after scanning. We assume \(l\) ac coefficients (of course with \(A > 0\)) to be of nonzero amplitude.

**Step 1 (full sort):**
Select the coefficients with \(A > 0\) and order them in an amplitude stack. Let \(k\) be the index of the coefficients after sorting (if allowed, the notation of the \(k\)th coefficient \(F_k(A_k)\) is abbreviated to \(F_k(A)\)). For the \(k\)th selected coefficient \(F_k(A_k)\) the following holds:

\[
|F_k(A)| \geq |F_{k+1}(A)|, \ldots, |F_k(A)| > 0, \quad 1 \leq k \leq l - 1. \tag{8}
\]

The corresponding addresses \(A_k\) are stored in an address stack. If \(|F_j(A_j)| = |F_k(A_k)|\) and \(A_j < A_k\), then \(j < k\).

**Step 2 (construct difference signal):**
Calculate the amplitude differences \(D_k(A)\) defined by:

\[
D_k(A) = |F_{k-1}(A)| - |F_k(A)|, \quad 2 \leq k \leq l, \\
D_1(A) = |F_1(A)|, \quad k = 1. \tag{9}
\]

This difference signal \(D_k(A)\) is a nonnegative sequence for each block. The original amplitude signs of \(F_k(A)\) are encoded separately. There are no changes in the address stack during this step.

**Step 3 (address calculation):**
Recalculate from each address \(A_k\) an address \(T_k\) according to:

\[
T_k = A_k - N_k(0, A_k), \quad D_k > 0, \\
T_k = A_k - A_{k-1} - N_k(A_{k-1}, A_k), \quad D_k = 0, \tag{10}
\]

where \(N_k(A_1, A_2)\) refers to the number of (previously encoded) coefficients \(F_j(A_j)\), \(1 \leq j \leq k - 1\), which have addresses satisfying \(A_1 < A_j < A_2\). The stack with amplitude differences is not changed during this step.
Motion-adaptive coding of video signals

Step 4 (bit mapping):
Coding of the sequence $D_k$ and the corresponding addresses $T_k$ for $1 \leq k \leq l$ is performed here with two variable-length code tables, i.e. $VLC_D(D_k)$ and $VLC_T(T_k)$. The dc coefficient $F(0)$ is always transmitted as well as the signs of each selected coefficient $F_k$.

7. Results

In order to assess the performance of motion-adaptive coding, intraframe coding experiments have been carried out for the adaptive and the non-adaptive cases. We have calculated the mean-squared error and the signal-to-noise ratio of the reconstructed pictures.

The applied colour test sequences have 576 active video lines per frame and 720 active samples per line, according to CCIR recommendation 601-1. The video samples are quantized with 8-bit accuracy for luminance and chrominance. The video frames are constructed from interlaced fields (288 lines) with a 50 Hz field rate. Although several test scenes are available for experiments, only a very few are suitable for judging the picture quality in areas with motion, since in some cases the human eye is not capable of tracking the objects because they move too fast, whereas in other situations the motion is too slow. In the latter case, intraframe coding is more efficient than intrafield, whereas in the former case the gain of motion adaptivity is perceptually less relevant, as mode coding noise is permitted. The results in the paper will be based on the sequence 'CAR', from which a frame is shown in fig. 2.

The mean-squared error $\sigma^2_e$, defined by

$$\sigma^2_e = \frac{1}{KL} \sum_{i=1}^{K} \sum_{j=1}^{L} [f(i,j) - \hat{f}(i,j)]^2,$$

(11)

where $\hat{f}(i,j)$ ($0 \leq \hat{f}, f \leq 255$) is the reconstructed sample and $K \times L$ the picture size, has been measured for the motion-adaptive algorithm and the nonadaptive technique. The results are shown in table I for various bit rates, between 1.0 and 2.4 bits/sample. It can be seen that the motion adaptivity significantly improves the reconstructed picture. Moreover, the gain turns out to be fairly constant, roughly 10%, for a wide range of bit rates.

We have also measured the signal-to-noise ratio (SNR):

$$\text{SNR} = 10 \log_{10} \left( \frac{255^2}{\sigma^2_e} \right),$$

(12)
TABLE I
Mean-squared error for motion-adaptive and non-adaptive coding at different bit rates (bit rate expressed in bits/sample).

<table>
<thead>
<tr>
<th>Bit rate</th>
<th>Non-adaptive</th>
<th>Motion adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>86.02</td>
<td>76.75</td>
</tr>
<tr>
<td>1.2</td>
<td>67.53</td>
<td>60.86</td>
</tr>
<tr>
<td>1.4</td>
<td>56.67</td>
<td>51.55</td>
</tr>
<tr>
<td>1.6</td>
<td>48.66</td>
<td>44.82</td>
</tr>
<tr>
<td>2.0</td>
<td>38.10</td>
<td>35.71</td>
</tr>
<tr>
<td>2.4</td>
<td>32.12</td>
<td>30.57</td>
</tr>
</tbody>
</table>

Fig. 10. SNR improvement with motion adaptivity.

where $\sigma_e^2$ is the mean-squared error defined in eq. (11). The results for the sequence CAR are shown in fig. 10. It can be concluded that the improvement in the SNR is noticeable for a large range of bit rates. The difference seems moderate but it is emphasized that the improved sample reconstruction obtained in moving areas of the picture is compensated by a larger number of samples without improvement in the stationary areas. The number of motion blocks is generally a fraction of the total number of blocks in a frame.

We conclude this section by discussing the subjective image quality, as the figures from eqs (11) and (12) are only a rough indication of the quality
Motion-adaptive coding of video signals

obtained. The subjective picture quality improves in all areas where motion appears at sufficiently high speed so that motion-dedicated coding is chosen, but is not so fast that the human eye cannot follow the moving object. The major improvement is a reduction in coding noise in the areas of motion. The subjective picture quality increases more than the expected improvement from the calculated SNR. At low bit rates, e.g. 1.0 bits/sample, the improvement is considerable since then coding noise is more visible than in systems based on higher bit rates. The gain in subjective image quality decreases at higher bit rates, i.e. at 2.0 bits/sample or more.

8. Conclusions

Motion-adaptive intraframe transform coding has been described. In intraframe coding of interlaced video signals, vertically adjacent samples in a block differ by one field period in time definition. We have highlighted the increase in spectral components in the vertical and diagonal frequencies in the case of motion, which results in a decreased coding efficiency in areas with moving objects. For this reason, motion-adaptive processing is required in order to obtain a high performance in intraframe coding of moving scenes.

A simple motion detection scheme has been provided on the basis of a Hadamard coefficient calculation. The performance of such a detector can be easily controlled by a threshold, and may be further improved by additional complexity. A major advantage of the proposed algorithm is its a priori character, so that the application of hardware-expensive a posteriori systems—in which several techniques run in parallel—is avoided. We have shown that with limited additional complexity the transformer can be modified for block-adaptive intraframe–intrafield computations. This is obtained by applying two different (fast) algorithms in the vertical DCT processor, while the choice for one of the two algorithms is governed by the motion detector.

Motion-adaptive intraframe–intrafield coding improves the subjective image quality for a large range of bit rates. The gain in the measured mean-squared error, a reduction of roughly 10%, appears to be fairly constant. Reduction in coding noise is mostly visible in areas with sufficient motion where the human eye is capable of following the movement (at 1–2 bits/sample).

Acknowledgement

Words of appreciation are addressed to my colleague S. Borgers for his valuable suggestions and opinions in our instructive system discussions.
REFERENCES

17) W. Keesen, Deutsche Thomson-Brandt, Hanover, F.R.G., personal communication.

Author

P. H. N. de With: Ir. degree (Electrical Engineering), University of Technology, Eindhoven, The Netherlands, 1984; Philips Research Laboratories, Eindhoven, 1984-. In the Magnetic Recording Department he has been working on data reduction techniques and their implementation for digital video recording. From 1985 to 1988 he participated in a European research project on video recording. At present he is involved in the European RACE project in which recording of HDTV signals is studied.