NEAR-LOSSLESS COMPLEXITY-SCALABLE EMBEDDED COMPRESSION ALGORITHM FOR COST REDUCTION IN DTV RECEIVERS

Mihaela van der Schaar[1] and Peter H. N. de With[2]

[1]Philips Research Briarcliff, 345 Scarborough Rd., Briarcliff Manor NY 10510, USA

[2]University Mannheim, Fac. Computer Engineering, B 6-26, Mannheim, Germany

Abstract— **This paper presents a novel complexity-scalable technique to reduce the memory costs of an MPEG-2 decoder by a factor of four to six, by recompressing the I- and P-reference pictures prior to motion-compensated reconstruction. The proposed scheme features some unique characteristics, like robustness for multiple encodings and high memory and memory-bandwidth reduction factors. At the same time, the near-lossless quality obtained is preserved, independently of the ratio between the input and display resolution and transmission/storage bit-rates, because no interference between the memory reduction algorithm and the MPEG-2 decoder occurs. Due to the inherent scalability of the presented algorithm, trade-offs between memory, silicon costs or over-all system costs, and image quality can be easily made. The proposed embedded compression algorithm can also be used for memory and memory-bandwidth reduction in alternative systems, like video coding ICs (e.g. MPEG-4) and memory intensive image processing ICs (e.g. de-interlacing and 100-Hz conversion).**

Keywords— **Video compression, MPEG coding, memory reduction, memory-bandwidth reduction, DTV receivers, embedded memory compression, scalable complexity compression techniques.**

I. Introduction

In the coming years, digital television broadcasting will gradually replace analog television. It is expected that the increasing number of digitally broadcasted services and the high video and audio quality offered will lead to a growing demand for DTV receivers. However, the consumer acceptance of this new technology will rely considerably on the possibility to lower the implementation costs of the DTV receivers. This paper presents a new, near-lossless and complexity-scalable technique to reduce the memory costs of a key component in all DTV receivers: the MPEG-2 decoder [1]. By recompressing the I- and P-reference pictures prior to memory storage for motion-compensation, the memory costs of an MPEG-2 decoder can be significantly reduced. The implementation of such an embedded compression technique is non-trivial, because an inappropriate algorithm could lead to an undesired increase in the over-all system costs or to visible quantization error propagation. The error accumulation can occur, because the embedded compression is inserted in the recursive MPEG-coding process (i.e. it builds further on previous reconstructions). This can be seen from the temporal loop in Fig. 1.

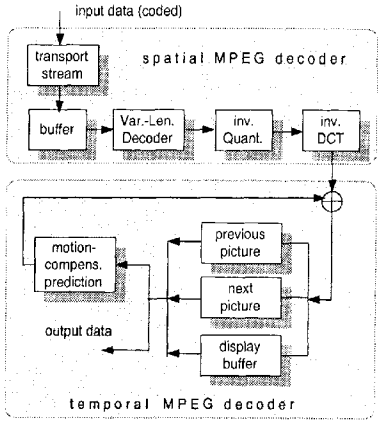


Fig. 1. Block-diagram of an MPEG-2 decoder.

In [2][5], it was already shown that the embedded memory compression (from now on referred to as "embedded compression") of the MPEG-decoder reference pictures is possible. This paper presents a new complexity-scalable memory reduction system, whose complexity can be varied gradually between the low-cost implementation presented in [2] and a relatively complex algorithm partially reusing the MPEG decoding system itself, to obtain a near-lossless performance. Due to its scalability in complexity and quality, the proposed memory-reduction algorithm can be successfully applied for reducing the memory of alternative systems, like image processing and other video coding ICs. Besides memory cost reduction, the embedded compression within DTV receivers is an attractive aid to reduce the bandwidth for communication with the shared (background) memory. This communication channel is often congested, because many digital signal processing functions and a control processor demand access to the memory simultaneously. With the scalable memory-reduction algorithm presented in this paper, trade-offs between memory size and memory-bandwidth reduction can easily be made.

The paper is divided as follows. Section II presents a thorough survey of the system and architectural constraints which are important when defining a memory and memory-bandwidth reduction algorithm. Subse-

Original manuscript received April 20, 2000

quently, the algorithmic and architectural constraints for embedding a compression system in the motion-compensated prediction loop of the MPEG codecs are identified. Section III is devoted to the embedded memory-compression system being used. We start by presenting the framework employed for the complexity-scalable architecture, which is based on a feedforward coding system. Subsequently, the various building blocks of the feedforward system are described in detail, and the conditions for near-lossless quality operation are discussed. The section concludes with an analysis of the memory-bandwidth reduction obtained with the proposed embedded compression system. Section IV presents the simulation results of an MPEG-2 decoder with embedded compression. Then, examples of complexity-scalable schemes based on the proposed architecture are given and their performances are evaluated. Finally, the conclusions are drawn.

II. System requirements

In this section, a classification of the systems which can benefit from memory and memory-bandwidth reduction algorithms is presented. The purpose is to identify the specific set of requirements imposed by each system and to justify the need for a solution which is scalable in both complexity and quality. Section II-A discusses in detail the system issues which are of major importance in the design of embedded compression schemes, and classifies our solution with respect to alternative proposals from literature. Subsequently, a decision diagram portrays an expert system which can determine which algorithmic settings should be chosen for the memory-reduction schemes. Section II-B describes the specific architectural and algorithmic requirements for the memory reduction of a DTV system, having the same input and output resolution.

A. System considerations and classification

Currently, several memory-reduction schemes have been proposed in the literature, but most studies fail to analyze the impact of the reduction schemes on the overall system in which they are embedded and viceversa. Also, the focus is mostly on the algorithmic constraints, neglecting the specific system and architectural requirements imposed by the embedding system on the memory-reduction scheme. However, it is our believe that for a successful design of memory-reduction schemes, it is of utmost importance to first assess the system parameters and architectural constraints of the enclosing system. For example, the embedded compression algorithm should consider the resulting increase/decrease in memory bandwidth and the way it affects the entire architecture and communication within the total system. Furthermore, the image quality of the memory-reduction algorithm should be thoroughly assesed, since it can influence considerably the performance of the overall system (e.g. an image enhancement module positioned after a low-quality memory-reduction scheme can result in poor output results). Therefore, in this section we try to characterize and classify the requirements imposed by the system on the memory-reduction scheme.

Display resolution vs. input resolution.
Within the ATSC standard, 18 input and 5 display formats are specified for the digital transmission of video. Therefore, a DTV receiver should be able to decode all the different input formats and display them on the chosen display resolution. Two types of systems are currently developed within the consumer market: (1) systems preserving the received resolution of the input signal, and (2) systems where the displayed resolution is clearly lower than the resolution of the coded video data.

If the display resolution is lower than the input image size of the original MPEG-2 coded sequence, a simple downscaling mechanism as proposed in [3][4] can be successfully deployed for reducing the memory and memory-bandwidth requirements of an MPEG-2 decoder. This is possible since the introduced artifacts are masked by the lower resolution display. In this case, more complex techniques for memory reduction (e.g. embedded compression) would unnecessarily raise system costs for providing an unnoticable increase in image quality.

However, if the display resolution is larger or equal to the input image size, downscaling techniques would lead to visible and unacceptable artifacts. Example systems are an HDTV-receiver with HDTV display or the European DVB-standard, having equal input and output resolutions. In such a system, embedded compression techniques, as introduced in [2][5], provide a better solution.

The principal difference between downscaling and embedded compression is that with downscaling, the reference I- and P-frames are stored with reduced resolution, whereas embedded compression preserves the resolution and reduces the required memory for MPEG-2 decoding by recompressing the I- and P-frames (see Fig. 2).

High-quality vs. consumer applications.
In high-quality applications, such as studio-to-studio communication, the quality and bit rate of the video sequences is much higher than for ordinary broadcasting to consumers. The image quality of the decoded MPEG-2 streams (or alternatively, the transmission bit-rate) is an essential parameter in the design of an embedded compression system, since its image qual-

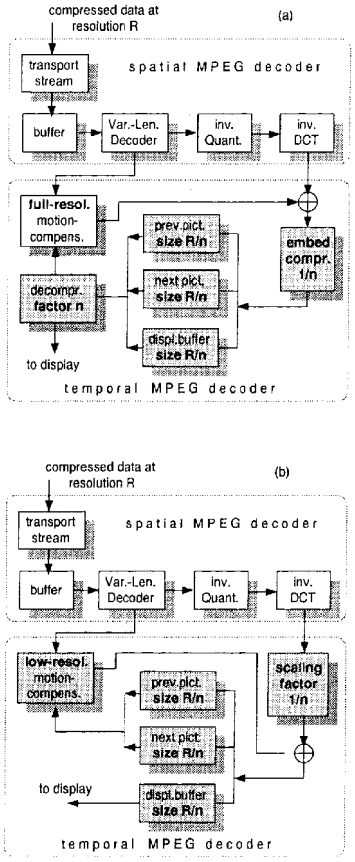


Fig. 2. Block diagrams of MPEG-2 decoders with memory reduction, (a) via embedded compression and (b) via scaling.

ity needs to be considerably higher than that of the MPEG-2 system in which it is embedded, in order to avoid visible interferences between the two coding systems (see [2]). In Section III, a clear distinction between low and high transmission bit-rates is made, leading to different system implementations. For example, if an MPEG-2 encoder operates at 4 Mbit/s, the image quality and the corresponding complexity of the embedded-compression algorithm can be lower than for an encoder working at 15 Mbit/s.

MPEG-2 encoding vs. MPEG-2 decoding.
A compliant MPEG-2 decoder should be able to decompress the image, independently of the received/stored bit-rate. The compliant MPEG-2 decoder can therefore make no *a priori* assumptions about the statistics of the decoded images to be stored in the reference frames. This implies that an adequate memory-reduction algorithm should be able to cope with both high and low quality images, without introducing visible artifacts in the decoding process. In the case of the MPEG-2 encoder designed for consumer use, the reference frame statistics are known in advance: the operation bit-rate is known at encoding time and moreover, most rate-controllers estimate in real-time the complexity of the video images. Additionally, the embedded compression system can be tuned and integrated more easily in the quality control of the encoders.

Dedicated vs. media-processor based solutions.
Current digital signal processing ICs provide a large amount of flexibility and modularity and as a result, programmable media-processors are becoming increasingly popular [8]. Media-processors typically contain a large external memory based on SDRAM, which is shared by a number of image processing functions (see Fig. 3). In a media-processor performing several video processing functions, the data is compressed and decompressed repeatedly prior to memory storage, such that a very high image quality should be preserved by the memory-reduction algorithm. Moreover, a low-quality memory-reduction algorithm would interfere with the proper functioning of the image processing functions (e.g. sharpness improvement, noise reduction etc.). Therefore, for media-processors which perform a succession of image (post-) processing functions while preserving the input image resolution, the only option for memory reduction is near-lossless embedded compression, since downscaling would result in a considerable image degradation after multiple down/up-scaling cycles.

Memory vs. memory-bandwidth reduction (external vs. on-chip memory architecture).
In many MPEG architectures, the coding and image processing functions are integrated and share the same external memory (see Fig. 3). In such a system, the communication channel to the memory threatens to be congested, so that reducing the used bandwidth to the memory becomes of vital importance. However, for a dedicated system with on-chip (embedded) memory, the reduction of the memory bandwidth is less relevant.

The memory-bandwidth reduction that can be theoretically achieved with embedded compression is computed in Section III-D. The memory-bandwidth reduction achieved by downscaling is roughly equal to the downscaling factor. For embedded compression, such a relationship does not always hold, as illustrated by the detailed analysis in Section III-D.

Expert system for memory reduction schemes.
Based on the previously mentioned system and architectural consideration, we can conclude that finding an optimal memory/memory-bandwidth reduction algorithm for a particular system involves solving a multi-dimensional optimization problem, in which various cost-functions play a role. For example, the im-

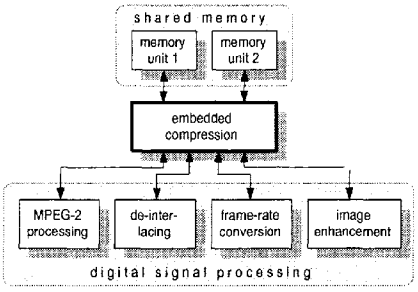


Fig. 3. Digital signal processing system based on a shared memory architecture and memory-reduction algorithm (embedded compression).

plementation cost of a memory-reduction algorithm is traded-off with the costs of the memory saved (this is especially true in the case of embedded memories, where the entire system including the memory is implemented in the same IC-process). Moreover, the outcome of this trade-off depends on the IC technology available to the manufacturer. A memory-bandwidth reduction cost-function is more difficult to estimate. The bandwidth can be so scarce that increased system costs are accepted to reduce this parameter.

Recently, several papers have been devoted to downscaling by sample-rate conversion and embedded memory compression, however, without any comparisons. In this paper we attempt to classify these techniques and also provide a framework for comparison. Fig. 4 depicts a system comparison and trade-offs for selection, based on the aforementioned system classification parameters. A key decision at the top in the diagram is high-quality versus consumer applications. For high-quality applications, costs are not relevant so that an advanced embedded compression system based on MPEG building blocks is possible. Secondly, for dedicated solutions the trade-off between downscaling and compression is important and depends on the input and display resolution. Thirdly, if the memory bandwidth is a critical parameter, the designer may further optimize on costs or quality and use adapted compression systems. A detailed example of a trade-off between high-quality and consumer applications and the resulting configuration of the memory-reduction scheme can be found in Table II at the end of this paper. The Table II provides important parameters for the embedded compression system and the corresponding costs and quality.

B. Embedded compression requirements

Let us now list the most relevant requirements that have been used for designing our proposal. These requirements can be divided into two categories: system and algorithmic requirements.

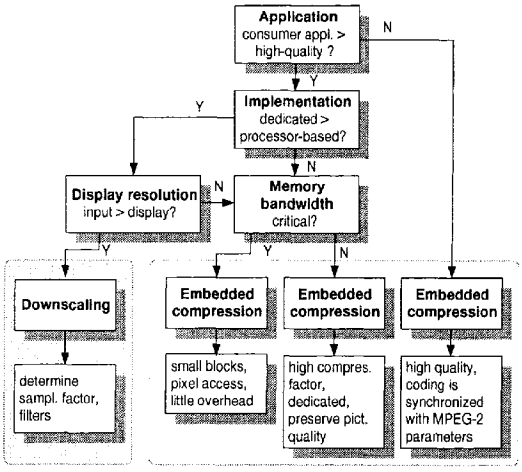


Fig. 4. Decision diagram for embedded compression, where "$\geq$" means "more important in the system than".

System and architectural requirements.

- Our aim is to reduce the overall system costs of a DTV receiver, by reducing the memory size of its MPEG-2 decoder by a factor of 4–6, independent of the transmission bit-rate or picture format. This is a relevant feature, since a DTV receiver should be able to operate flawlessly for various bit rates and image formats.
- The target DTV receivers have a display resolution that is equal (e.g. HDTV display for ATSC formats or DVB) to the input image resolution. This is the most challenging system, because any interference in quantization between MPEG-2 decoding and embedded (memory) compression becomes clearly visible and is not masked by a lower resolution display, as assumed in alternative systems [3][4]. In the remainder of the paper, it is assumed that the input and output images have standard-definition resolution (i.e. 720 pixels $\times$ 576 lines at 25 frames/s). However, the algorithm presented is suited for all 18 different ATSC formats, including the 1920 $\times$ 1080 format at 30 frames/s.
- The memory bandwidth should also be reduced with a factor of at least two, to allow for a programmable implementation of the DTV functions using a shared memory architecture. Various post-processing functions, for example deblocking and deringing, could be activated simultaneously with the MPEG-2 decoding.
- The memory-reduction algorithm should have a low complexity, such that the costs of the overall system with embedded compression are clearly lower than the original costs of the DTV system.
- The memory-reduction algorithm should preferably be scalable in complexity, such that the same algo-

rithmic structure can be reused for memory reduction in alternative systems with different architectural requirements. Moreover, in a media-processor based solution, where the memory reduction can be performed in software on a CPU, a scalable algorithm is attractive, because it allows a real-time trade-off between the complexity, memory-reduction factor and image quality. However, for such an architecture, high-speed media-processors are required to allow the reuse of hardware and software building blocks for embedded compression.

Algorithmic requirements.

For a successful application of embedded compression to reduce the memory costs of MPEG, a number of algorithmic requirements need to be satisfied.

- Fixed compression factor for each individual picture.
- Near-lossless image quality compression, such that artifacts from the embedded compression are not visible.
- Limited or no prediction drift due to repetitive coding/decoding in the recursive MPEG coding loop.
- Fast access to pixel data in the compressed domain, to enable real-time operation of the motion-compensated decoding.

The last three requirements reveal that special constraints are imposed upon an embedded compression algorithm, which are very different from the requirements of traditional coding schemes.

III. Embedded compression algorithm

In this section, we present in detail the framework adopted for the design of the memory-reduction schemes. All embedded compression algorithms employed are based on a generic feedforward coding scheme, which is introduced in Section III-A. This scheme defines only the high-level architecture, and various algorithms can be employed for its implementation, depending on the imposed system and algorithmic requirements, thereby being inherently complexity scalable. In Section III-B, a specific implementation of the feedforward scheme is presented and then in Section III-C, various quantization methodologies are discussed to ensure near-lossless quality for the embedded coding system. Finally, in Section III-D, an analysis of the memory-bandwidth reduction obtained with the proposed embedded compression system is given.

A. Feedforward coding scheme

To fulfill the previously mentioned system and algorithmic requirements, while preserving a high image quality independent of the MPEG-2 transmission bitrate, a feedforward-buffered coding scheme, as shown in Fig. 5, has been adopted. (A detailed description of the functioning of a feedforward coder can be found in [9]. In the sequel, it is assumed that the feedforward coding mechanism is known.) The feedforward coding scheme has been used, because it is able to code small subimages (e.g. a few MPEG-2 macroblocks) with a fixed compression factor. This is a major advantage, since fast access to the data in the compressed domain is possible: the start address of each compressed group of pixels can be directly computed from the fixed segment length. A feedback coding system with a stable buffer regulation is generally too slow to realize fixed-length coding of segments, yielding variable coded length segments.

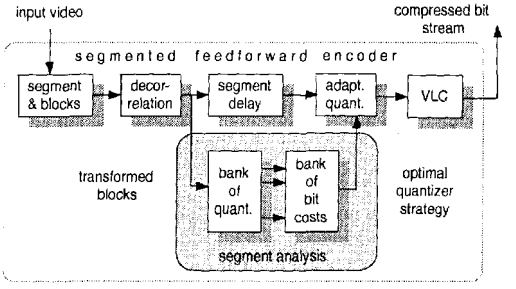


Fig. 5. Architecture of the feedforward coding system.

The feedforward coding system has already been successfully applied for the embedded memory compression in MPEG-2 decoding [2]. In this paper, we attempt to define a complexity-scalable system based on the same feedforward architecture. The processing steps of the employed feedforward coding system are briefly listed below.

i. Segment construction. This unit groups a set of MPEG-2 macroblocks together into a *segment*, to be coded with a fixed compression factor. The segment size depends on the memory that can be spent for embedded compression (based on system costs) and can influence considerably the image quality of the compressed data (see [2]). For example, in a scalable system, depending on the predefined or real-time system requirements, a variable segment size can be applied to influence the trade-off between memory size and memory-bandwidth reduction. With a large segment size, a high-compression factor can be obtained by the embedded compression system, thereby enabling considerable memory savings. On the other hand, a small segment size, e.g. one MPEG-2 macroblock, allows easy access to the compressed data for motion-compensation with limited overhead, and therefore the memory bandwidth can be limited (see [2]).

ii. Decorrelation. Subsequently, the video data within the 8×8 MPEG-2 blocks of the segment are decorrelated employing either a low-cost DCT technique which is described in Section III-B or the standard DCT-transform used in MPEG-2 encoders. The embedded codec also employs blocks of 8×8 pix-

els to match MPEG-2, thereby avoiding any reformatting of the data by the embedded coder. Since the performance of the embedded coder is influenced by the decorrelation capability of the applied DCT-IDCT, trade-offs should be made between the desired memory-reduction factor, corresponding image quality and the allowed DCT/IDCT complexity.

iii. Local adaptivity measurement. An adaptivity measure for each block is applied to choose a suitable quantizer, which matches the local image characteristics. According to this local adaptivity measure, the blocks can be grouped in different classes, each adopting a specific quantization. In our system, the maximum absolute value of the AC coefficients has been used to quantify the activity within a DCT block. The number of classes can be varied.

iv. Segment analysis. This unit involves the calculation of the bit cost for one segment as a function of various quantizer tables, *prior* to the final quantization and coding. The bit cost of the quantized transform coefficients depends on two parameters: the local adaptivity per block (step **iii**) and the "average" quantizer choice for the complete segment, called *strategy*. In the segment analysis, the bit costs obtained with different quantizer strategies are computed and compared with the desired memory reduction factor. The strategy that yields the desired bit cost with the best quality will be selected for the final quantization. As in the local adaptivity measurement, the number of evaluated "quantization strategies" can be varied to trade-off system costs versus image quality.

v. Adaptive quantizer. In this unit, all blocks of the actual segment are quantized with the same "average" strategy, which was selected by the segment analyzer. The applied quantizer in a block depends on both this "average" strategy, and the local adaptivity measure mentioned in step **iii**. Therefore, the quantizer can be seen as a 2-D table, of which the access is determined by the two aforementioned parameters (see [2]).

Finally, to encode one frame, steps **i** to **v** are repeated for each segment. The subsequent sections are devoted to an in-depth study of the components required for implementing the feedforward system and to the specific scalable complexity trade-offs that can be made.

B. DCT coding algorithm

Multiplication-free DCT. The key issues for an efficient parallel hardware architecture of a fast DCT (FDCT) algorithm are: the number of multipliers and adders, the regularity and modularity in the computational structure, the number of registers (needed for the storage of the partial results) and the transformation accuracy. In [6], several algorithms have been compared for the low-cost implementation of the DCT. This study revealed that the 1-D FDCT scheme proposed by Chen [10] is very regular, with a small number of registers for the computational flow. Moreover, the accuracy of the transformation proved to be very high. However, the scheme requires a relatively large number of multipliers. To eliminate this drawback, we have examined the flow graph of the 1-D FDCT scheme of Chen and observed that all its multiplications are of the type $y = c_i \times x$, where c_i are fixed real numbers ($c_i = \cos((2i+1) \times \pi/16)$). To simplify the hardware involved in the FDCT computation, these multiplications can be replaced by add (subtract) and shift operations: each coefficient c_i is multiplied by 256 and then approximated by the closest integer that can be represented as a sum of three or five powers of 2. A similar result holds for the coefficients of the IDCT. The relevant cosine terms for an 8×8 DCT now become as follows.

$$c_0 = 256 \times \cos(\tfrac{\pi}{16}) \approx 251 = 256 - 4 - 1, (1)$$

$$c_1 = 256 \times \cos(\tfrac{3\times\pi}{16}) \approx 213 = 256 - 32 - 8 - 4 + 1, (2)$$

$$c_2 = 256 \times \cos(\tfrac{5\times\pi}{16}) \approx 142 = 256 - 64 - 32 - 16 - 2, (3)$$

$$c_3 = 256 \times \cos(\tfrac{7\times\pi}{16}) \approx 50 = 64 - 12 - 2. (4)$$

The drawback of this proposal is a reduction of the decorrelation properties achieved by this modified DCT transform. The resolution of the DC coefficients is limited to 9 bits and the AC coefficients are represented with 8 bits plus a sign bit.

The proposed multiplication-free DCT-IDCT was compared with various other fast algorithms and found to be superior with respect to cost and computational transparency, at the sacrifice of a small decrease in the decorrelation capability (see [6]). Moreover, since only limited compression factors need to be achieved by the embedded compression mechanism, the limited efficiency of the low-cost DCT does not significantly influence the algorithm performance.

Robust Quantization. For the frequency-dependent quantization of the transform coefficients, two options exist depending on the desired/allowed complexity: either adopt a dedicated low-complexity quantizer based on simple shift operations (i.e. quantization with powers of 2), as in (5), or employ the MPEG-2 intra-quantizer itself, as in (6). The first option gives

$$F_q(u,v) = \frac{F(u,v)}{2^{n(u,v,s)}}, \quad (5)$$

where $F(u,v)$ and $F_q(u,v)$ are the DCT coefficients prior to and after the quantization, respectively. The shifting factor depends not only on the coefficient po-

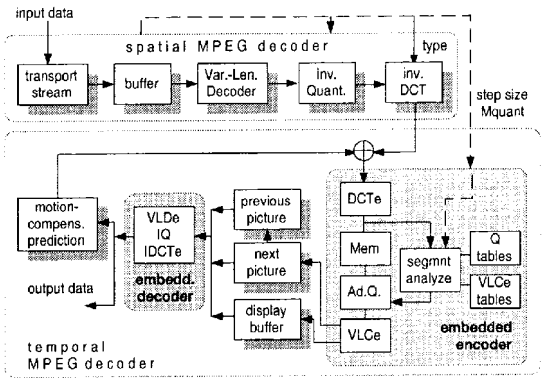


Fig. 6. MPEG-2 decoder with embedded compression using feedforward transform coding.

sition (u, v) in the block, but also on the global quantization strategy s (see step **D** in Section III-A). The MPEG-2 intra-quantization yields

$$F_q(u,v) = \frac{\frac{32\times F(u,v)}{W(u,v)} + \frac{3\times mquant}{4}}{2 \times mquant}, \qquad (6)$$

where $W(u,v)$ is the weighting matrix (which is coefficient position dependent) and *mquant* stands for the *global* quantization step for the entire MPEG-2 macroblock.

In (6), the weighting matrix $W(u,v)$ has values between 8 and 83 [1], which are not powers of 2. Therefore, the MPEG-2 intraframe quantization process requires two divisions for each transformed coefficient, using $W(u,v)$ and *mquant*. Hence, it is considerably more complex than the quantization described in (5), which is based on shift-operations only.

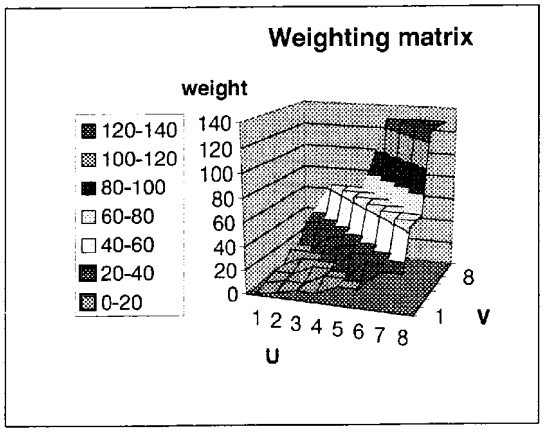


Fig. 7. 2-D Weighting matrix based on powers of 2, plotted for quantization strategy s_i, i.e. $2^{n(u,v,s=s_i)}$.

Fig. 7 and Fig. 8 refer to equations (5) and (6), respectively, and show that the MPEG weighting is

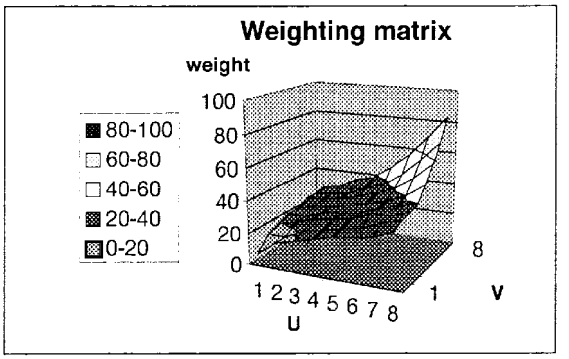


Fig. 8. 2-D Weighting matrix used in MPEG-2, i.e. $W(u,v)$.

smoother, and the shift-based weighting has a more *abrupt* characteristic. For equation (5), a quantization strategy s has been selected to illustrate the *coarseness* of the first quantization mechanism. The abrupt weighting from equation (5) (i.e. 2^n) does not match very well with the human visual system characteristics and therefore leads to a decrease in the perceived image quality compared with the MPEG-2 based weighting. Therefore, a trade-off needs to be made between image quality and complexity.

Since the choice between the two quantization mechanisms is key to our embedded compression algorithm, we will discuss it later in more detail (see Section III-C).

Entropy Coding. For the efficient transmission of the quantized transform coefficients, a low-cost but very efficient entropy coding method has been adopted. In this algorithm [7], the zonal area containing relevant coefficients is determined and transmitted to the decoder, by specifying the largest column V_{max} and largest row U_{max} values for which non-zero coefficients $F(u,v)$ occur (see Fig. 9).

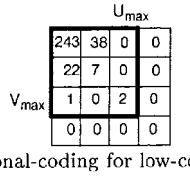


Fig. 9. Adaptive zonal-coding for low-cost entropy coding.

All the coefficients inside the determined zone $(0...U_{max}, 0...V_{max})$ are transmitted, including the zero-valued coefficients. For an improved compression, the coefficients inside the zone are Huffman coded. For illustration, in Table I, the Huffman codes for a specific *adaptivity* class (see "Local adaptivity measurement" in Section III-A) are presented. The AC coefficients with absolute values not indicated in the table are coded using an escape code (101) followed by the uncompressed representation of the absolute value of the AC coefficient.

AC-Value	Huffman-code
0	0
1	100
2	1100
3	1101
4	1110
5	11110
6	11111
>6	101+value

TABLE I
EXAMPLE OF HUFFMAN CODES USED FOR ADAPTIVE ZONAL-CODING.

The algorithm is basically adaptive zonal coding, leading to a bitcost B per block:

$$B = \sum_{u=0}^{U_{max}} \sum_{v=0}^{V_{max}} Huff_q(u,v) + 6, \qquad (7)$$

where the 6-bit addition is required to indicate the U_{max} and V_{max} values.

Despite its slightly lower compression capability, this entropy coder is clearly less complex than the MPEG-2 significance map compression, because it requires less processing steps and memory.

C. *Near-lossless embedded compression dependent on the transmission bit-rates*

The conventional video coding algorithms (i.e. MPEG-2, H.263) and the more recent MPEG-4 algorithm use different quantization techniques and can be divided in two classes, depending on their image quality and the corresponding transmission bit-rate.

Low to moderate transmission bit-rates. The first class includes video coding schemes giving low and moderate decoded image quality, like MPEG-2 codecs at low bit-rates (e.g. bit-rates lower than 4 Mbit/s for SD-resolution images), H.263, MPEG-4, etc. To obtain these target bit-rates, a relatively coarse quantization is applied for most macroblocks and only a small number of non-zero AC coefficients are transmitted on the average. Consequently, the employment of a relatively coarse quantizer in the embedded coder does not lead to a visible quantization interference with the surrounding video coding system. For this class, the 2^n quantizer mentioned in Section III-B forms an appropriate choice. It allows a very efficient hardware implementation and prevents the unnecessary VLC coding of the "artificial" high-frequencies typically represented by blocking artifacts, which can appear if the MPEG-2 and embedded compression block-grids or transforms do not match. Since the image quality obtained by the embedded coder considerably outperforms that of the surrounding system (i.e. MPEG-2 decoder), the two coding mechanisms do not need to be tightly coupled (i.e. can work independently of each other). It is interesting to mention that this low-complexity embedded coder is extremely suitable for reducing the cost of the conventional video encoders, since the transmission bit-rate is known *a-priori* and can be controlled by the encoder itself. For low and moderate bit-rates, a more accurate quantizer would result in an unnecessary complexity increase.

High transmission bit-rates. The second class includes the DTV receivers that should be able to decode all possible bit-rates and image qualities. For relatively large bit-rates (e.g. bit-rates higher than 9 Mbit/s for SD resolution images), the decoded image quality is very high (e.g. PSNR higher than 36 dB) and the transform coefficients are quantized only marginally. Therefore, the embedded compression quantizer should avoid the zeroing of higher-order AC coefficients, by employing a finer weighting function than $2^{n(u,v,s)}$ from equation (5). To minimize the potential interference between the embedded compression and the applied coding standard, the best alternative is to "track" the applied MPEG dequantization step (i.e. *mquant*) and reuse the MPEG quantization process. As already mentioned, the MPEG-2 weighting matrix $W(u,v)$ in equation (6) follows the human visual system more smoothly than the $2^{n(u,v,s)}$ function in equation (5) (see also Fig. 7 and 8). Moreover, the MPEG-2 quantization process does not remove high-frequencies in contrast to the low-cost $2^{n(u,v,s)}$ quantizer. To realize a near-lossless compression, the MPEG-2 coding parameters are therefore also employed by the embedded compression. Let us now discuss the embedded compression of the I and P-pictures separately. The quantization parameter *mquant* used by the MPEG coder for the I-pictures can be reused by the embedded coder. Whenever an image has been coded on a frame basis, in order to obtain the same quantization process as in MPEG-2, the same DCT-structure of the luminance blocks (e.g. frame- or field-based) should also be employed. Consequently, the MPEG-2 IDCT and the embedded-compression DCT can be discarded from the computational flow of the I-picture decoding, thereby reducing the overall computational complexity. Moreover, if the total bit-cost obtained by quantization with *mquant* fits within the desired memory size, then the computational flow can be even further simplified by eliminating the MPEG-2 de-quantization and the embedded compression quantization, as depicted in Fig. 10. In most cases, the compression factor achieved by MPEG-2 for each individual segment is already larger than 6 and the embedded compression of the I-image is therefore lossless. When the desired

amount of bits per segment is exceeded, the embedded feedforward codec increases the quantization strategy per segment to ensure that the desired compression factor is achieved (see Fig. 11, where the initialization is performed with the *mquant* value employed by MPEG-2 instead of *mquant* equal to unity).

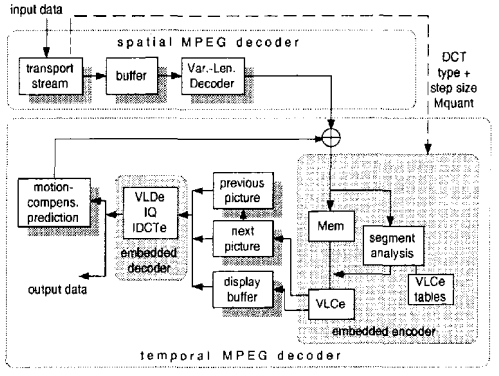


Fig. 10. Simplified block-diagram of an MPEG-2 decoder with embedded compression based on re-usage of MPEG-2 functions and parameters.

The embedded compression of the P-pictures is different, since the decoded MPEG-2 blocks represent only temporal prediction errors, while the entire reconstructed picture should be stored within the memory for the decoding process (i.e. for motion-compensation). After a group of MPEG-2 macroblocks has been decoded (i.e. prediction-error decompression and motion compensation), the data is again (embedded) compressed prior to its storage within the reference frame memory. The feedforward codec presented in Section III-A analyzes the data within a segment by employing different quantization strategies, varying from coarse to fine. For the MPEG-2 quantizer inside the embedded coder, these different quantization strategies are represented by various values of the *mquant* parameter (see Fig. 11). For each quantization strategy, the total bit-cost B_{mquant} after the entropy coding of the significant AC coefficients is compared with B/C, where B denotes the bit-cost of the uncompressed data in the segment and C is the desired compression factor, which is fixed per frame. If the desired compression factor is not reached, the quantization step *mquant* is increased and the process is repeated.

D. Memory-bandwidth reduction computation

In MPEG-2, the motion-compensation requires that macroblock data should be retrieved from memory in "arbitrary" positions, which mostly do not coincide with the block grid employed for embedded compression. Therefore, supplementary (embedded) decoding of border pixels is required, which consumes additional memory bandwidth, as illustrated in Fig. 12. Since

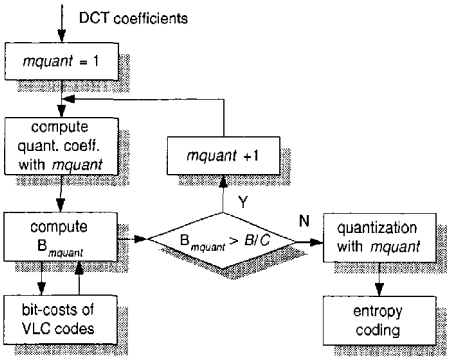


Fig. 11. Feedforward coding algorithm based on the MPEG-2 quantization mechanism.

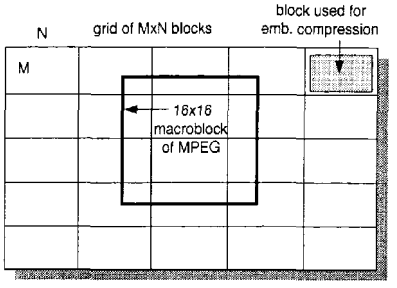


Fig. 12. Macroblock extraction from a compressed frame, where the employed block-size for embedded compression has been $N \times M$ pixels.

the image data is stored in the memory in compressed form, an image header is stored separately, containing both the start addresses of the coded segments and the (offset) positions of each compressed block within the segment, called P_B (Pointer Bits). The memory-bandwidth reduction (MBR) factor is determined in (8), assuming that the embedded compression system with compression C employs $N \times M$ blocks, so that

$$MBR = \frac{16}{\left\lfloor \frac{16+N-1}{N} \right\rfloor} \cdot \frac{16}{\left\lfloor \frac{16+M-1}{M} \right\rfloor} \cdot \frac{1}{\frac{M \cdot N}{C} + \frac{P_B}{8}} . \quad (8)$$

For the system discussed with $N = M = 8$, C=6 and P_B=9, the MBR equals 2.5. The MBR should be considered carefully because for a C=2, the memory bandwidth would have increased, due to the embedded compression. Similar calculations can be derived for various block sizes, compression factors and address information systems. This analysis becomes extremely important when an embedded compression system needs to be dimensioned such that it achieves the best trade-off between image quality (i.e. large block sizes) and memory-bandwidth reduction (i.e. small block sizes).

It is important to notice that the previously computed MBR factor does not take into account the overhead induced by burst reading (i.e. contiguous data

access) of memory blocks from an SDRAM. If large bursts of data need to be accessed simultaneously, the memory bandwidth for retrieving motion-compensated data can increase considerably depending on the burst-length, since a large amount of overhead is also retrieved. To improve the efficiency in memory communication, the data corresponding to subsequent blocks should be written in independent memory banks, which can be accessed individually during one clock-cycle, so that the average access time to a specific macroblock is reduced. These observations indicate that the actual MBR depends on both the memory structure, its addressing interface, and the data arrangement within the memory.

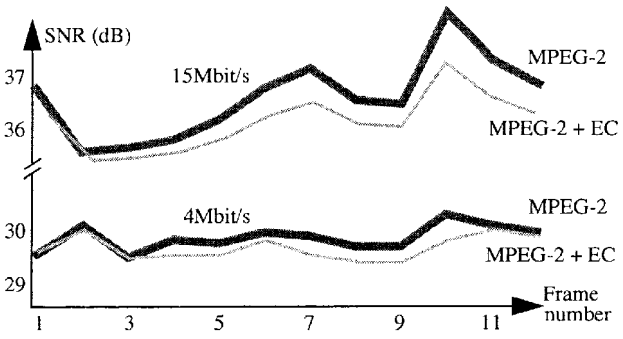


Fig. 13. Relative comparison between the embedded compression (EC) image quality and the image quality of the MPEG-2 codecs at various bit-rates for the MOBI sequence.

IV. Results and Conclusions

Experiments have been carried out employing the novel embedded compression technique inside MPEG-2 decoders. The results obtained show that the described feedforward embedded coder with the MPEG-2 quantizer achieves a memory reduction of 6 with almost no loss in image quality, independent of the transmission bit-rate (see Fig. 13). Simulations with the embedded compression have been performed for a range of image sequences at various bit rates (4-15 Mbit/s). Mostly, the quality decrease due to embedded compression is very small, e.g. 0.2 dB and 1 dB for the worst case, whereas the memory cost of the DTV receiver is considerably reduced.

A major advantage of the proposed algorithm is its robustness for repetitive coding/decoding in the recursive MPEG decoding loop, so that negligible prediction drift occurs. Another important feature of the technique proposed is that it is equally well applicable to other image resolutions, because it can "follow" the MPEG decoder, as it is partially based on the same techniques and coding parameters.

The hardware complexity of the proposed system can be downscaled, if required. The complexity of the embedded coder is dominated by the feedforward buffer and the MPEG-2 quantizer. Fortunately, both blocks can be reduced in complexity, e.g. by limiting the number of MPEG macroblocks used for feedforward buffering and/or employing a dedicated 2^n quantizer, which can be implemented with bit-shifts only. A trade-off between cost and performance is therefore possible and is illustrated by Fig. 14 a and b.

In Table II, two possible configurations of the presented feedforward coding architecture are given. The corresponding performance is also indicated. This illustrates that the architecture is scalable in complexity and quality and can be adapted to meet the specific requirements of different systems.

Finally, it was found that the novel algorithm can also be successfully applied to reduce the memory requirements of other digital signal processing functions, such as spatio-temporal sample-rate converters, or video encoding systems, like MPEG codecs. The proposed embedded coder outperforms the algorithms currently known in the literature for reducing the memory size of DTV receivers, because it can operate with near-lossless quality for a large range of bit-rates and image resolutions. Therefore, the proposed algorithm can be successfully employed for reducing the memory of a scalable (programmable) DSP system, of which the operating range and input image size is not known in advance.

REFERENCES

[1] *Coding of Moving Pictures and Associated Audio*, International Standard, ISO/IEC 13818, November 1994.

[2] M. van der Schaar and P.H.N. de With, *Novel embedded compression algorithm for memory reduction in MPEG codecs*, VCIP '99, San Jose CA, SPIE Proc. Vol. 3653, part 2, pp. 864 -873, Jan. 1999.

[3] A. Vetro et al., *Minimum Drift Architectures for 3-Layer scalable DTV decoding*, ICCE '98 Digest Techn. Papers, Los Angeles CA, pp. 527–536, June 1998.

[4] J.G.N. Henderson, *All-Format Decoders and Set-Top Boxes*, Proc. ICIP '98, Vol. 1, Chicago IL, p. 4, Oct. 1998.

[5] P.H.N. de With. P.H. Frencken and M. van der Schaar, *An MPEG decoder with embedded compression for memory reduction*, IEEE Trans. Consum. Electron., Vol. 44, No. 3, pp. 545–555, August 1998.

[6] M. van der Schaar and P.H.N. de With, *On the Application of Fast DCT Transforms for Combined SW/HW Implementation*, Proc. 18th Int. Sympos. Inform. Theory in the Benelux, Veldhoven (NL), pp. 33–40, May 1997.

[7] C. Yamamitsu, et al., *An Experimental Study for Home-use Digital VTR*, IEEE Trans. Cons. Electr., Vol. CE-35, No. 3, pp. 450–457, August 1989.

[8] E.G.T. Jaspers, P.H.N. de With and J.G.W.M. Janssen, *A flexible heterogeneous video processor system for television applications*, IEEE Trans. Cons. Electr., Vol. CE-45, No. 1, pp. 1–11, February 1999.

[9] P.H.N. de With and A.M.A. Rijckaert, *Design considerations of the video compression system of the new DV camrecorder standard*, IEEE Trans. Cons. Electr., Vol. 43, No. 4, pp. 1160–1179, November 1997.

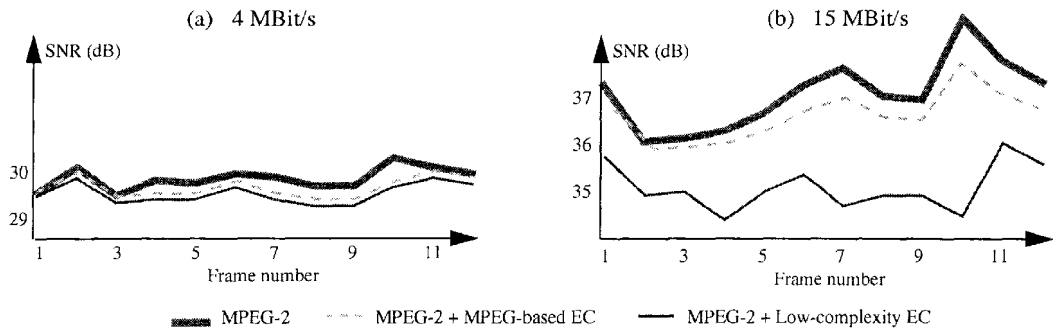


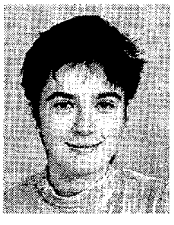
Fig. 14. Image quality of the MPEG-2 decoded SD-resolution MOBI sequence (very critical) at 4 Mbit/s (a) and 15 Mbit/s (b), without and with embedded compression (EC) with fixed compression factor 6.

System targeted	Low-end	High-end
DCT	Multiplication-free	Standard accuracy
Quantization	Shift-based (eq. 5)	MPEG-2 based (eq. 6)
Entropy coding	Adaptive zonal coding	(Run, Amplitude) VLC of MPEG-2
Segment-size	5 MPEG-2 Macroblocks	9 MPEG-2 Macroblocks
SNR_Y	approx. 40 dB	approx. 45 dB

TABLE II

TWO EMBEDDED COMPRESSION SYSTEMS AND THEIR PSNR WITH A COMPRESSION FACTOR OF 4, BASED ON THE PRESENTED COMPLEXITY SCALABLE ARCHITECTURE.

[10] W. H. Chen, C. H. Smith and C. Fralick, *A Fast Computational Algorithm for the Discrete Cosine Transform*, IEEE Trans. Communic., Vol. COM-25, No. 9, pp. 1004–1009, 1977.



Mihaela van der Schaar graduated in electrical engineering from Eindhoven University of Technology, the Netherlands, in April 1996. In the same year she joined Philips Research Laboratories Eindhoven, where she became a member of the TV Systems Department. From 1996 to 1998, she was involved in several projects which investigate low-cost very high quality video compression techniques and their implementation for TV and computer systems. Since 1998, she is an expatriate at Philips Research Briarcliff, USA, where she is a member of the Video Communications Department and is involved in the research of video coding techniques for Internet video streaming. Since 1999, she is an active participant to the MPEG-4 video standard, contributing to the Fine Granularity Scalability tool. Her research interests include video and graphics coding and image analysis and she co-authored more than 15 papers in this field and holds several patents.

Peter H.N. de With graduated in electrical engineering from the University of Technology in Eindhoven. In 1992, he received his Ph.D. degree from the University of Technology Delft, The Netherlands, for his work on video bit-rate reduction for recording applications. He joined Philips Research Laboratories Eindhoven in 1984, where he became a member of the Magnetic Recording Systems Department. From 1985 to 1993 he was involved in several European projects on SDTV and HDTV recording. In the early nineties he contributed as a video coding expert to the DV standardization committee. In 1994 he became a member of the TV Systems group, where he was working on advanced programmable video processing architectures. In 1996 he became senior TV systems architect and in October 1997, he was appointed as full professor at the University of Mannheim, Germany. At the faculty Computer Engineering, he is currently heading a group working on video coding, processing and its realization. Regularly, he is a teacher of the Philips Technical Training Centre and for other post-academic courses. In 1995, he co-authored the paper that received the IEEE CES Transactions Paper Award. In 1996, he obtained a company Invention Award. In 1997, Philips received the ITVA Award for its contributions to the DV standard. Mr. de With is a senior member of the IEEE, program committee member of the IEEE CES and board member of the Benelux working group for Information and Communication Theory.