System requirements and considerations for visual table of contents in PVR

System Requirements and Considerations for Visual Table of Contents in PVR

Onno Eerenberg and Peter H. N. de With

Abstract *— With the introduction of non-tape-based digital video recorders also known as Personal Video Recorders (PVR), consumers can expect alternative navigation methods to the well-known trick-play modes found on analog and digital tape-based systems. In this paper, we explore the system requirements and aspects of a Visual Table of Contents (VTOC). A primary advantage of implementing a VTOC search mode is that it provides a much higher visual performance at high search speeds (50 times or higher) than the conventional search techniques. We present a solution for generating a video signal for visual search that is based on reusing MPEG-2 compressed video data. The video search signal is composed of a set of MPEG-2 compressed sub-pictures, resulting in a mosaic screen. An efficient strategy is introduced, that allows either full or partial reuse of the compressed sub-pictures via motion compensation of earlier reference sub-pictures to allow the generation of a new mosaic screen*[1].

Index Terms — Mini-slice, Mosaic Screen, MPEG, Navigation, PVR, Trick-play, Visual Table of Content.

I. INTRODUCTION

Personal Video Recorders with optical disks [1] or Hard-Disk-Drive (HDD) storage media are equipped with trick-play modes such as fast-search and slow-motion, to navigate through the stored video information. Unlike tape-based video recorders [2] [3] [4] [5], non-tape-based storage media enable fast random access of the stored video information, resulting in fast-search trick-play facilities. This feature allows refresh-rates up to the frame rate of the video signal, at full spatial resolution [11]. For trick-play with low or medium speed-up factors, this results in an attractive spatial-temporal video quality when compared to the solutions of current analog and digital consumer Video Cassette Recorders (VCR). In this paper, we propose a (fast-) search technique based on a Visual Table Of Contents (VTOC), which is motivated by a plurality of reasons. First, for high speed-up factors such as 50 times or more, the perceived temporal quality decreases due to the low temporal correlation between the succeeding images creating the video trick-play sequence. This lower perceptual quality occurs due to the fact that the eye cannot track the rapid change of the video sequence. Second, the storage capacity and the video compression factors have increased rapidly in the past decade, so that many video sequences can be recorded on the same medium. This phenomenon makes traditional fast-search video navigation, a less powerful tool to search for a certain video extract. Third, the emergence of new standards addressing image analysis and its description [10], enable new possibilities for video summarization. In the past decades, mosaic screens have been successfully used to create an instant overview of program channels, offered by cable providers. We have adopted this concept for video navigation, to decouple the video refresh-rate during visualization from the trick-play search speed.

For home networks based on digital consumer devices, transported signals are preferably compliant with open standards to enable interoperability. In such networks, the digital receiver is unaware of the information source to which it is connected. This avoids complicated communication protocols and overhead, thereby eliminating communication latency.

This paper discusses a technique to derive and generate a VTOC in a networked home storage system, which allows transmission of this VTOC navigation signal over a digital interface [9]. A technical solution to provide video navigation functionality can be split into two situations: (1) signal processing is performed during recording for enabling trick-play, or (2) this processing is carried out during playback. Depending on the technical solution, the trick-play video quality varies for tape and disk-based systems, due to the nature of data storage.

It is beyond the scope of this paper to present a complete overview of trick-play techniques applied in various digital consumer recording systems. Instead, we will highlight a few typical examples. For tape-based digital storage systems such as DV [2] or D-VHS [5], during recording, signal processing is required onto the recorded video information to enable trick-play at playback. This is caused by the fact that the storage medium is only locally accessible since the tape is wrapped onto a cylinder. For example, let us consider the trick-play processing for DV-based systems. During fast search, the scan path for reading differs from the normal-play scan path. For some particular tape speeds, reading regions are predetermined, creating a virtual channel for search information. In a DV system, the compressed video data is stored in shuffled format on tape, such that during search the picture quality is optimized without sacrificing the normal-play picture quality [3]. For disk-based storage systems,

[1] Onno Eerenberg is with NXP Semiconductors Research, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands. (e-mail: onno.eerenberg@nxp.com).

Peter H. N. de With is with CycloMedia Technology and Eindhoven University of Technology, Den Dolech 2, 5600 MB, The Netherlands (e-mail: P.H.N.de.with@tue.nl).

Manuscript received June 4, 2008

another concept can be applied. A popular method is the use of the so-called Characteristic Point Information (CPI) [6]. CPI can be seen as a Look-Up-Table (LUT), used to store characteristics of the recorded program. The LUT provides e.g. the physical storage position on disk of the normal-play entry points. A normal-play entry point is created at the beginning of an MPEG-2 Group-Of-Pictures (GOP) by commencing with an intra-encoded picture. Conventional trick-play search is implemented by jumping to positions at the storage medium, based on CPI information, retrieving a fragment of the normal-play recording. This fragment contains an intra-encoded picture, which has a specified temporal distance to the previously-retrieved fragment that corresponds to the selected trick-play speed-up factor.

This paper presents an algorithm and architecture to implement a VTOC. The key features of this novel concept are decoupling of the frame rate and the video display-rate and the usage of an additional video data stream that is dedicated for constructing mosaic screens. The advantage of such a special information stream is that it allows fast response time and arbitrary search speeds to compose the mosaic screens. These mosaic screens allow an instant overview of a normal-play fragment. Our concept is based on deriving sub-pictures from the stored normal-play video sequence, compress these pictures using MPEG-2 video compression, and avoiding intermediate transcoding of the sub-pictures when composing different mosaic screens. The derived mosaic screens are MPEG-2 compliant [7] [8] and can be transmitted across a digital interface [9]. The attractiveness of the concept is that the involved MPEG-2 decoding and encoding is performed only once to construct new mosaic screens, either via reuse of the original MPEG-compressed sub-pictures or via motion-compensation techniques.

The paper is divided as follows. Section II presents an outline of a PVR system equipped with traditional and VTOC video navigation. Section III provides a brief overview of the relevant MPEG-2 coding syntax used in this paper. Section IV presents the concept of hierarchical mosaic-screen navigation. Section V explores the sub-picture generation and its reuse for MPEG-compressed hierarchical mosaic screens. Experimental results are described in Section VI. Finally, conclusions are presented in Section VII.

II. PVR System Overview

Our starting point is a PVR system, based on an HDD capable of storing an MPEG-2 Transport Stream (TS). This system is equipped with CPI generation [6] and the traditional trick-play functionality as described in [11], and mosaic-screen navigation. Figure 1 portrays a basic system overview. At the left side of the diagram, the compressed MPEG-2 TS enters the system (TSin). This signal can be viewed in real-time, via signal path (a), or in time-shift mode via signal path (b), or via path (c) for the situation that video navigation is applied for the stored program. Although an MPEG-2 decoder is depicted in Figure 1, the decoder could be outside the system and be connected to the PVR via a digital interface [9]. This is possible because an MPEG-compliant signal is available at the right side of the switch under all PVR modes of operation. The information extracted by the CPI block is stored in the Meta-database-block, which is permanently stored at the HDD. The Read-list block operates on the Meta-database, when the PVR is operated in navigation mode. The CPI information is enhanced with the information extracted by the trick-play and Mosaic-screen controller block. Examples of information added to CPI are e.g. the derived sub-pictures and scene-change information. Clustering descriptive information via CPI lowers the complexity for consistency between CPI and the corresponding main recording. The 'To-disk' block handles the incoming TS data traffic to the HDD, whereas the 'From-disk' block retrieves all normal-play TS data from the HDD. The scheduling mechanisms of the To- and From-disk blocks provide a streaming interface capable of handling streaming data preserving the real-time properties. For conventional navigation, e.g. fast- or slow-speed trick-

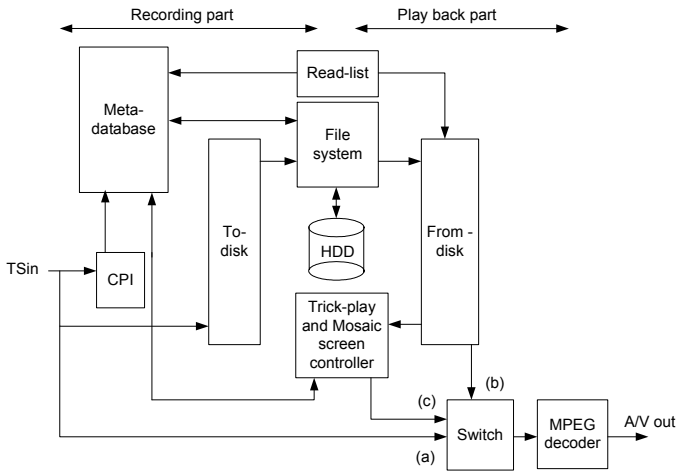

Fig. 1. PVR system block diagram with traditional and VTOC navigation support. Switch in position (a) indicates the real-time viewing mode; in case (b), the play-back of delay viewing; and in case (c), the navigation operation mode.

play, the 'Trick-play and Mosaic-screen controller' block operates on the normal-play data retrieved by the From-disk block under control of the Read-list block. For mosaic-screen navigation, the trick-play and Mosaic-screen controller block operates on the sub-picture data stored in the meta-data-base block. The trick-play and mosaic screen controller block is positioned in such a way, that for the generation of sub-pictures, no real-time constraints are imposed, allowing sub-pictures and more advanced scene-description information to be generated after finishing the actual recording. This allows the generation of mosaic screens based on the method as described in this paper, but also allows the generation of mosaic screens on the basis of e.g. scene-change information, which is also stored in the meta database and derived by Video Content Analysis (VCA). In the remaining sections of this paper, we elaborate on the involved VTOC signal processing, which is embedded in the trick-play and Mosaic-screen controller block.

III. INTRODUCTION TO MPEG-2 CODING SYNTAX

Block-based video compression schemes operate on a group of spatially adjacent pixels. A distinction can be made between intraframe and interframe compression. In intraframe compression, only spatial information is used for compression, whereas for interframe compression also information from the past and or the future pictures is used for compression. The advantage of intraframe compression is that the picture can be decoded without the need of information from previous or future pictures, whereas a drawback is the modest compression factor. To achieve efficient video compression, interframe compression needs to be part of the coding scheme. MPEG-2 video compression is achieved by applying a DCT transform on each group of 8×8 pixels, which are quantized and runlength coded. Four luminance DCT blocks are combined with two or maximally eight chrominance DCT blocks, depending on the applied color sub-sampling format, resulting in a so-called macroblock. Such a macroblock is equipped with horizontal locator information, also known as *macroblock_address_increment*. One or more macroblocks

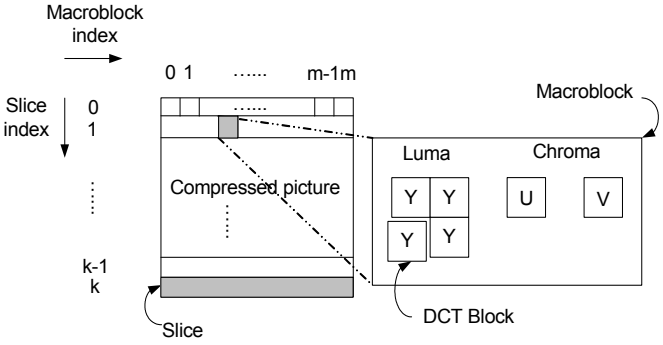

Fig. 2. A block-based compressed picture and its compression syntax elements for video with 4:2:0 color sub-sampling format.

can share the vertical locator information, resulting in a so-called slice. The locator information is part of the slice header and inserted into the *slice_start_code*, see Figure 2. Finally, for interframe compression, a macroblock can be predicted from previous and or future pictures, resulting in one or more motion vectors, depending on the picture coding type and the prediction type, which for MPEG-2 can be field- or frame-based. If no temporal correlation is found within the search area of motion estimation, a macroblock is intraframe coded, even if the picture type is predictive.

IV. HIERARCHICAL MOSAIC-SCREEN NAVIGATION

Navigation using a Visual Table of Content (VTOC) needs to be intuitive from the user perspective. To avoid navigation from becoming a tedious task, care must be paid to the VTOC implementation. A good starting point here is the temporal correlation of a regular video sequence. The temporal correlation heavily depends on the nature of the video material, resulting in typical scene durations between three up to five seconds. This observation forms the basis for the selection criteria of the normal-play pictures, from which the sub-pictures are derived to construct the base-layer-mosaic screens, i.e. the lowest hierarchical navigation layer. The number of sub-pictures constructing a mosaic screen depends on the appropriate viewing distance. From experiments, we have concluded that mosaic screens based on sixteen sub-pictures turn out to be a good trade-off between spatial complexity, that is the ability to concentrate on a specific sub-picture, versus sufficient resolution to interpret the sub-picture video content taking a normal viewing distance into account. Let us now proceed with a more formal approach.

A digital video signal is obtained via time-discrete sampling of the analog video signal. The result is a sequence of video frames denoted by $f_k(i,j)$, for $0 \le j \le W-1$, $0 \le i \le H-1$, k is the frame index, running from 0, 1, 2… Hereby, the parameter W represents the picture width, and H denotes the picture height.

The base-layer mosaic sub-pictures are derived from the sequence $f_k(i, j)$ by applying a sub-sampling of the frame index k, where the sub-sampling factor corresponds to the trick-play speed-up factor. This factor is explained below and differs for various video navigation layers.

Let us assume that k corresponds to the time kT_f, where T_f is the frame time which e.g. equals 40 ms for the European case. Our approach is to apply a selective sub-sampling within the sequence of frames. The speed-up selection factor for doing this is defined by parameter k_s. Using the observation that a video scene on the average changes every three seconds and a sub-picture of each scene is desired, the base-layer sub-sampling factor k_s becomes 75. Equation (1) indicates the frame index k_1 that selects the frames from the normal-play sequence $f_k(i, j)$, which form the base-layer sub-pictures, and

$$k_1 = kk_s \begin{cases} k = 0,1,2,..., \\ k_s = 75. \end{cases} \tag{1}$$

This results in 1,200 sub-pictures per hour of normal-play video, leading to 75 mosaic screens containing 16 sub-pictures at the base layer, each summarizing 48 seconds of normal-play video. A typical two-hour movie is summarized in this way by 150 mosaic screens. Interpretation of the individual screens, becomes a cumbersome task, especially when hours of video material is involved. This situation is avoided when a *hierarchical mosaic-screen* navigation is introduced. The hierarchical structure allows efficient browsing through a complete movie and enables a quick identification of the sequence of interest. The VTOC proposal as presented in this section is limited to three hierarchical layers. The relation between the hierarchical layers is chosen such, that when descending in the hierarchy, there is a well-defined relation between the sub-pictures of the higher layer and the mosaic screens of the layer directly below. Therefore, a higher layer mosaic screen is derived from the base-layer sub-pictures, using a speed-up factor that is an integer multiple of the base-layer speed-up factor, such that in the next hierarchical layer mosaic screen, only one sub-picture of the lower-layer mosaic

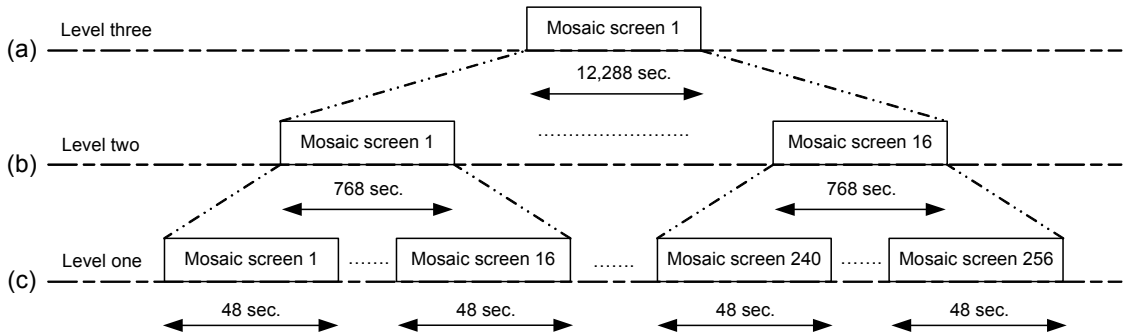

Fig. 3. Hierarchical navigation layers, their normal play abstract duration and absolute normal play picture index for mosaic screen constructed using 16 sub-pictures and a speed-up factor of 75 for the base layer. For level three (a), this result in an abstract interval of 12,288 seconds, for level two (b), 768 seconds, and for level one (c), 48 seconds per mosaic screen.

screen appears. Equation (2) indicates the frame index for the sub-pictures selection process of the second hierarchical navigation layer, which is specified by

$$k_2 = kk_s S \quad \begin{cases} k = 0,1,2,..., \\ k_s = 75, \\ S = 16. \end{cases} \tag{2}$$

In this expression, k is a running index, S the number of sub-pictures per mosaic screen, k_s the base-layer speed-up factor and k_2 the frame index of the second hierarchical navigation layer. Equation (3) indicates the frame index for the sub-pictures selection process of the third hierarchical navigation layer, resulting in

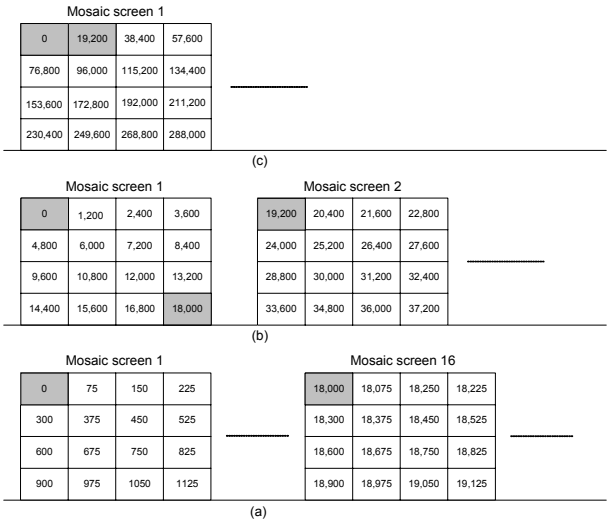

Fig. 4. Hierarchical mosaic screens composed of sub-pictures with their absolute frame index and the sub-picture position relation between two successive layers. (a) Mosaic screens of the first hierarchical navigation layer. (b) Mosaic screens of the second hierarchical navigation layer. (c) Mosaic screens of the third hierarchical navigation layer.

$$k_3 = kk_s S^2 \quad \begin{cases} k = 0,1,2,..., \\ k_s = 75, \\ S = 16. \end{cases} \tag{3}$$

Again k is a running index, k_s the base-layer speed-up factor, S the number of sub-pictures per mosaic screen and k_3 the resulting frame index for the third hierarchical navigation layer. Substituting the frame index k_2 and k_3 into index k for the sequence $f_k(i, j)$ delivers the sub-pictures for the hierarchical navigation layers two and three.

Figure 3 visualizes the normal-play summary per mosaic screen for each hierarchical layer. A selected sub-picture of a higher-layer mosaic screen always appears in the upper-left corner of the corresponding lower-layer mosaic screen. Figure 4 depicts the construction of hierarchical navigation layers using sub-pictures with indices derived by Equations (1), (2) and (3).

V. MPEG-2 Coded Hierarchical Mosaic-Screens

A. System aspects

Prior to starting this section, let us first consider a few system aspects to find a suitable approach for creating these sub-pictures. Construction of high-layer hierarchical mosaic screens, as presented in Section 3, is based on sub-pictures that are also used to construct mosaic screens at the base layer. Construction of high-layer mosaic screens based on reusing sub-pictures from the base layer, would conceptually involve a simple copying action, when the sub-pictures are available in the pixel domain. For the construction of a mosaic screen with low latency, corresponding to the third hierarchical layer and all its lower layers (see Figure 4), it would be required to store the complete base layer in RAM, involving a memory footprint of more than 200 MBytes. This holds for mosaic screens of 720 pixels by 576 lines and a 4:2:0 sampling format. This memory space is unacceptable for a search feature in a dedicated consumer product. Storage of the uncompressed sub-pictures on a hard disk medium may solve the storage footprint dilemma, but results into a communication-bandwidth penalty. Compression of the sub-pictures eliminates the footprint and bandwidth problem, but puts a heavy burden on the CPU cycle consumption when compression and decompression is performed in software. Besides this, a software implementation offers more flexibility. For this reason, we propose a novel compression concept that allows *reuse* of MPEG-compressed sub-pictures without the need for recompression.

B. Construction of mosaic screens

There are additional aspects that play a role in the construction of sub-pictures in the mosaic screen. It is preferred that each sub-picture is constructed from an integer number of macroblocks both horizontally and vertically. Compliant with MPEG-2 coding, a horizontal row of macroblocks forms a slice. Limiting the length of a slice to the width of a sub-picture creates a mini-slice with respect to the mosaic screen. For example, with four sub-pictures in the horizontal direction of a mosaic screen, we create four mini-slices horizontally in the mosaic screen (see Figure 6).

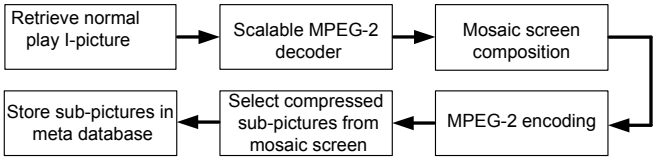

Fig. 5. Sub-picture processing chain for mosaic screens.

C. System aspects

Let us now discuss the coding aspects of our proposal. Prior to the detailed description, we briefly summarize our approach. The objective is to encode the sub-pictures facilitating the mosaic-screen application and yet avoiding an expensive implementation. This is achieved by selecting a proper picture coding type in combination with an elegant mosaic-screen composition. The usage of a predictive picture

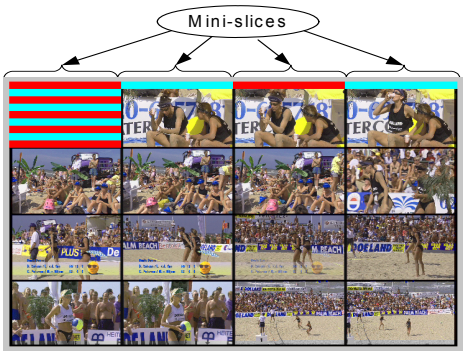

Fig. 6. Construction of a mosaic screen using sub-pictures composed of mini-slices.

type for compression allows motion-compensation techniques at a low computation cost, thereby reducing the computation cycles at the involved processor. Since each sub-picture is a frame-based selection from a video sequence, each sub-picture is independently coded, resulting in intraframe-coded macroblocks and thus intra-coded slices. The signal path for sub-picture generation which includes encoding, decoding and mosaic-screen composition is shown in Figure 5. For simplicity, we have employed MPEG-2 compression for the mini-slices, leading to MPEG-compressed sub-pictures. Although each mini-slice is composed of intraframe-compressed macroblocks, an MPEG-2 sub-picture, and thus a mosaic screen, can be of various picture types. MPEG-2 offers three picture types (I, P and B). A straightforward choice would be to select the I-type picture. However, this requires that each mosaic screen is fully recomposed and decoded, even if a significant similarity exists between two succeeding mosaic screens. Since this processing is implemented in software on a small co-processor, we prefer to avoid heavy computation at this processor, and thus reuse of already available information at the decoder side. As a result, instead of using the syntax associated to the I-type pictures, we use the syntax of a P-type picture, thereby enabling forward motion compensation. We propose to reuse the *already decoded* information that is waiting for reuse in the next mosaic screen, as the anchor information (reference) for the next MPEG-2 decoding cycle. The correct filling of a motion-compensated macroblock is obtained by defining a motion vector that refers to the desired information. Hence, the MPEG-2 decoder performs motion compensation for those areas that already exist with a zero difference signal (skip macroblock), and adds missing sub-pictures via intraframe decoding.

Now we will describe in more detail the physical construction of a mosaic screen. We target a high-quality mosaic screen that can be efficiently composed. On top of this, the generated signal should be decodable using a remote MPEG-2 decoder. Because a mosaic screen will be composed of sub-pictures from different time instants, special attention must be paid to the bit cost of each sub-picture to avoid a VBV-buffer overflow during mosaic-screen decoding. In fact, the system constraints are even more compelling than the previous statement. Sub-pictures may be constructed from mini-slices that originate from different time instants, with different compression settings. As a result, the decoding of such a sub-picture may already violate the VBV-buffer constraint. All these potential violations are solved when the mini-slices are coded with a fixed bit cost. As a bonus, the handling and memory management is reduced to straightforward pointer arithmetic. To compensate for the chosen simplicity and the possible loss of quality, individual sub-pictures are compressed at a relatively high bit rate. Fixed bit-cost compression of mini-slices in sub-pictures requires a considerable computing effort. However, it should be noted that the involved processing may be performed offline, which alleviates the computation effort.

D. Modification of the slice-start-code and the macroblock-address-increment locator information

Navigation through hierarchical mosaic screens requires the construction of new mosaic screens. Reuse of MPEG-compressed sub-pictures based on mini-slices becomes possible when the locator information is adapted. Two situations are distinguished:

- Modification of both mini-slice *slice-start-code* and the mini-slice first macroblock's *macroblock-address-increment*
- Modification of only the mini-slice *slice-start-code*

Construction of mosaic screens is performed in the compressed domain. Instead of a sub-picture-wise build-up of a mosaic screen at the MPEG-2 decoder side, we have chosen for a complete construction of a mosaic screen in the compressed domain, prior to transmission and/or decoding.

The construction of a new mosaic screen requires the correct positioning of the involved sub-picture mini-slices, which relates to the new mosaic screen. Because the mini-slices originate from different sub-pictures, the numbering of mini-slices and its first macroblocks (if necessary) needs to be modified into a regular scanning order, in order to obtain a fully MPEG-compliant picture. This requires a modification of the *slice_start_code* and/or *macroblock_address_increment* (see Section II for the introduction of these parameters). Fortunately, a modification of the Byte-aligned *slice_start_code* locator information field is easy and thus inexpensive, due to the fixed bit-cost compression approach avoiding slice parsing (pointer arithmetic). The modification of the variable-length coded *macroblock_address_increment* locator field requires bit shifting for byte alignment of the remaining mini-slice data. To avoid a change in mini-slice bit-cost, the original base-layer mini-slices are appended with zero-valued padding bytes. These bytes are used in the byte-alignment process to maintain the mini-slices at a fixed bit cost.

E. Alternative low-cost mosaic screen construction

A disadvantage of the mosaic-screen generation as described in the previous sub-section is that the modification of the *macroblock_address_increment* involves bit shifting, which is an expensive operation on an embedded general-purpose processor. This drawback can be avoided, by limiting the modification procedure to only the locator information contained in the *slice_start_code*. By constraining the modification process to only the *slice_start_code*, the positioning of the mini-slices is limited to only the vertical direction in the mosaic screen. As a consequence, this limitation results in a new sub-picture sub-sample grid applied to the base layer in order to derive higher-layer mosaic screens. The sub-pictures for the second hierarchical navigation layer are selected using sub-pictures that are derived from the base-layer sub-pictures, using the following relation

$$k_4 = kSk_s + (k \bmod S)\, l \quad \begin{cases} k=0,1,2,..., \\ k_s=75, \\ l=75, \\ S=16. \end{cases} \tag{4}$$

In this expression, k is a running index, S the number of sub-pictures per mosaic screen, k_s the base-layer speed-up factor. Parameter l represents the off-set resulting in a selection of MPEG-compressed sub-pictures, such that the horizontal positioning is preserved and only vertical repositioning of sub-pictures is allowed. The resulting frame index k_4 supports

(a)

(b)

Fig. 7. Composing a higher-layer mosaic screen via reuse of MPEG-2 compressed sub-pictures by changing only the slice-start-code of the involved mini-slices. (a) Indicates the reused (white border) sub-pictures. (b) Shows a mosaic screen based on slice-start-code adapted sub-pictures. The top row sub-pictures of (b) correspond to the sub-pictures of the diagonal of the (a) mosaic screen.

the construction of the second hierarchical navigation layer. The selected sub-pictures can be spatially repositioned via modification of the slice-start-code. The sub-pictures for the third hierarchical navigation layer are selected using sub-pictures from the base-layer sub-pictures using the following relation

$$k_5 = kSk_s^2 + (k \bmod S)\, m \quad \begin{cases} k=0,1,2..., \\ k_s=75, \\ m=1275, \\ S=16. \end{cases} \tag{5}$$

Again, k is the running index, S and k_s have the same meaning as in Equation (4), m is the off-set resulting in the proper selection of MPEG-compressed sub-pictures and k_5 the frame index for the third hierarchical navigation layer. Figure 7 visualizes the resulting construction of a mosaic screen of the second hierarchical navigation layer, reusing MPEG-compressed sub-pictures from the base layer, applying Equation (4). In this experimental example, the product $k{\cdot}S{\cdot}k_s$ was chosen to be 300 and l=375, to simplify the visualization of reusing MPEG-compressed sub-pictures by changing only the vertical locator information, as explained at the start of this sub-section.

VI. Experimental Results

Prior to presenting the experimental results, we commence with providing the details of key sub-blocks of our system.

First, we describe the core of the MPEG-2 scalable decoder sub-block as indicated in Figure 5. Second, we set the boundaries for the sub-picture compression. For this purpose, we determine the minimum bit cost and we verify the quality of a composed mosaic screen using the conditions that where derived from the sub-picture experiments.

A. MPEG-2 Scalable decoder

Let us first discuss the scalable MPEG-2 decoder. The sub-picture generation as presented in Section IV, indicates the usage of a scalable MPEG-2 decoder. Because the sub-pictures are derived from intraframe-compressed pictures, the scalable decoder involves conventional DCT processing steps, which are depicted in Figure 8. The reason for employing a scalable MPEG-2 video decoder are multifold and involve efficient usage of CPU cycles, memory footprint and bandwidth. Spatial sub-sampling in the MPEG-compressed domain is achieved by modification of the 2D-IDCT. Figure 9 indicates the maximal received MPEG-2 DCT coefficients of which a selection is made for applying the scalable MPEG-2 IDCT. The scalable 2D-IDCT uses only four coefficients, resulting in a 2×2 pixel block. The transform of limited coefficients converts a Standard Definition (SD) picture into a Quarter Common Intermediat Format (QCIF) picture. Equation (6) presents the regular MPEG-2 2D-IDCT formula which equals

$$f(x,y)=\frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cos\left(\frac{(2x+1)\pi u}{2N}\right)\cos\left(\frac{(2y+1)\pi v}{2N}\right), \quad (6)$$

$$C(u),C(v)=\begin{cases}\dfrac{1}{\sqrt{2}} & for\ u,v=0,\\ 1 & otherwise.\end{cases}$$

for $x,y,u,v=0,1,2,...,N-1$, where x,y are the spatial coordinates in the sample domain and u,v are coordinates in the transform domain.

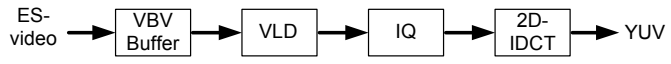

Fig. 8. Main building blocks of the MPEG-2 intraframe decoder.

TABLE I
PERFORMANCE OF A STANDARD 2×2 IDCT AND OPTIMIZED 2×2 IDCT RUNNING ON AN ARM926@160MHZ CLOCK, DECODING AN MPEG-2 ML INTRAFRAME-COMPRESSED PICTURE OF 1,620 MACROBLOCKS, AND 4:2:0 SAMPLING FORMAT

2×2 IDCT	Involved calculation time for one intra-frame (ms)	Mcycles
Standard	14.8	2.32
Optimized	8.5	1.36

For the standard MPEG-2 decoder *N=8,* but for the scalable MPEG-2 SD-to-QCIF decoder *N=2*. For a scalable MPEG-2 SD-to-QCIF decoder, the computation of Equation (6) is visualized in a diagram within Figure 10 for *N=2*. Figure 10

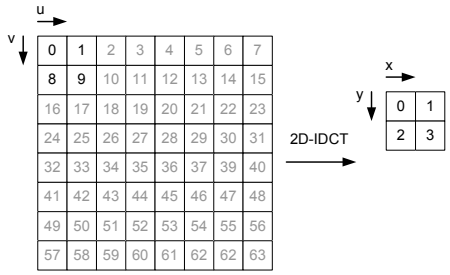

Fig. 9. Scalable 2×2 IDCT for an SD-to-QCIF MPEG-2 decoder. At the left, the received coefficients (0..63) are depicted. At the right, a 2×2 pixel block is depicted obtained by applying a 2×2 scalable IDCT.

indicates two different implementations for computing the scalable 2D-IDCT. The first implementation (a) is a direct translation of Equation (6), whereas the second implementation (b) is an optimized version, in which the multiplications have been removed. Optimization is obtained by normalizing the coefficients and adding a shift-right operation at the end of the computation. The two 2×2 IDCT implementations were benchmarked on an ARM926 processor running at 160 MHz. The influence of bus latency is eliminated, by continuously executing the test routine on the same data set. The test routine containing the IDCT is invoked

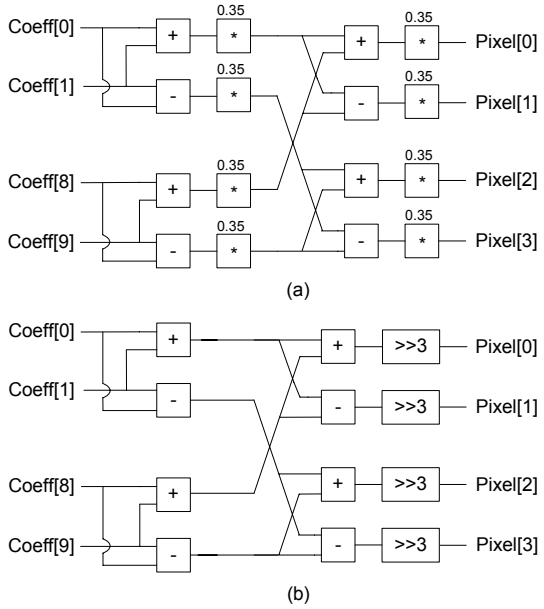

Fig. 10. Scalable 2×2 IDCT for an SD-to-QCIF MPEG-2 decoder. (a) Standard IDCT implementation. (b) Optimized IDCT implementation removing the coefficient multiplications.

9,720 times, which is equivalent to the maximum number of DCT blocks of an MPEG-2 ML picture with a 4:2:0 color sampling format. Table I contains the measurement results for the two IDCT implementations as depicted in Figure 10. The measurements show that the optimized implementation reduces the calculation time by almost a factor of two.

Let us now determine the quality level for individual sub-pictures of the mosaic screen. In Section IV, we discussed that the mini-slices within the sub-pictures are coded in a fixed bit cost. To determine this bit cost, we conducted a number of perceptual evaluations. The sub-pictures must be of good

quality because they are used for navigation purposes and may therefore be visually inspected by the viewer. This is emphasized for the situation that a sub-picture is tracked by the eye of the viewer during e.g. a scrolling operation. For the subjective experiments, a modified version of the MPEG-2 reference encoder software [12] was used. The primary modifications deal with the creation of mini-slices and the involved bit-rate control. In our experiments, we employed the MPEG-2 Main Level (ML) encoder settings, allowing a bit rate of 15 Mbit/s. Using the 25-Hz frame rate for European broadcasting, the video coder running at 15 Mbit/s, fills the VBV-buffer in almost three frame periods. The usage of a bit cost that is an integer multiple of the maximal bit cost per frame period, simplifies the generation of the final MPEG-2 TS. For our subjective experiments, we have encoded a mosaic screen at three different bit costs. The bit cost equals the product of the maximum MPEG-2 ML bit rate, the European frame period and a scale factor p, see Table II. The scale factor p expresses the mosaic-screen transmission time expressed in frame periods. Figure 11 indicates three sub-pictures, each originating from a mosaic screen that was differently encoded, using a bit cost according to Table II. The modified MPEG-2 reference encoder software [12], which is a single-pass encoder, delivers a bit cost that is lower than the bit cost indicated in Table II. The fixed bit cost per mini-slice is obtained by adding padding bytes to each mini-slice. From the simulation results, it is clearly visible that a mosaic-screen bit cost of 75,000 Bytes is insufficient and delivers sub-pictures with severe coding artifacts. A sub-picture from a mosaic screen based on 150,000-Bytes bit cost, still has some coding artifacts but also lacks sharpness, whereas a sub-picture from a mosaic screen with 225,000-Bytes bit cost results in a good picture quality. Because a mosaic screen consists of multiple sub-pictures, the total quality is determined by the quality of each individual sub-picture. Figure 12(b) depicts the Peak Signal-to-Noise Ratio (PSNR) of each individual sub-picture that is encoded using the single-pass encoder.

A good picture quality can be expected for a PSNR of 30 dB or more. This is the situation for mosaic screens with a bit cost of 225,000 Bytes, which is almost the complete VBV buffer-size. For MPEG-2 ML, this amounts to 229,376 Bytes. The proposed mosaic-screen bit cost does not result in severe requirements in terms of storage capacity and memory bandwidth. In Section III, we noticed that a two-hour movie resulted in 150 mosaic screens for the base layer. As a result, the sub-picture overhead corresponding to the 150 mosaic screens are stored in the meta-database, requiring almost 32 MBytes. This corresponds to a storage capacity of one minute of MPEG-compressed AV information at a bit rate of 4 Mbit/s. The required bandwidth for retrieval of the mosaic screen is 1.8 Mbit/s, if each second a complete new mosaic screen is to be presented on the display.

TABLE II
BIT COST FOR MOSAIC-SCREENS IN CASE OF VARIOUS TRANSMISSION TIMES AT A BIT RATE OF 15MBIT/S.

Mosaic screen bit cost (Bytes)	Transmission time p in number of frame periods	Mosaic screen transmission time (ms)
75,000	1	40
150,000	2	80
225,000	3	120

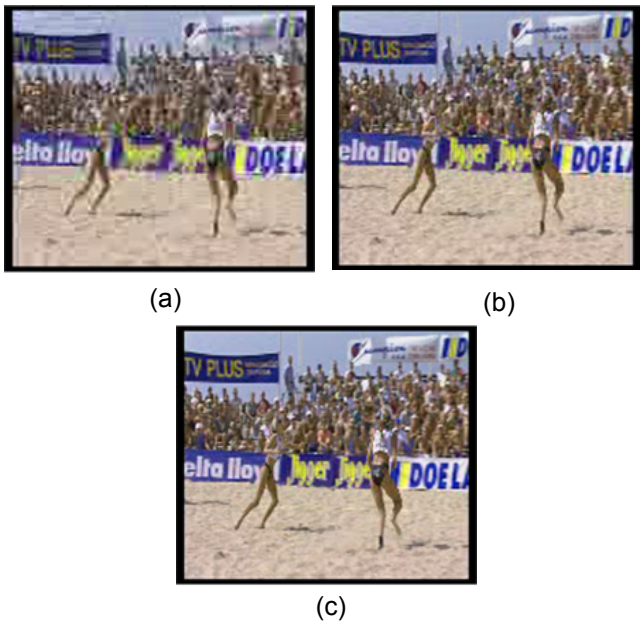

Fig. 11. MPEG-2 compressed sub-pictures using fixed bit cost mini-slices. (a) Sub-picture with a bit cost of 75,000 Bytes. (b) Sub-picture with a bit cost of 150,000 Bytes. (c) Sub-picture with a bit cost of 225,000 Bytes.

VII. RESULTS AND CONCLUSIONS

We have presented a new navigation method based on a VTOC that complements the existing conventional trick-play

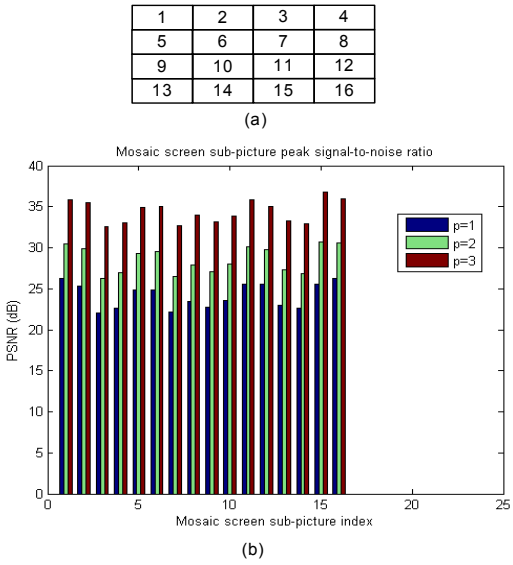

Fig. 12. PSNR of mosaic screen sub-pictures. (a) Sub-picture index. (b) Mosaic screen sub-picture PSNR, corresponding a transmission time p of one, two and three frame periods and a transmission rate of 15 Mbit/s.

search method. VTOC employs mosaic screens and is particularly attractive for high-search speed video navigation, due to decoupling of the video information and the video refresh-rate. The sub-pictures that build up the VTOC mosaic screens are derived from the normal-play video sequence, using either equidistant temporal sub-sampling, or they are based on e.g. scene-change detection. In the latter case, it provides an efficient instant overview of the normal play recording.

The VTOC implementation is elegantly supported by the use of so called *mini-slices,* which are compliant with the MPEG-2 standard. This approach allows the usage of a standard MPEG-2 decoder. The algorithm for creating the mosaic screens, has been optimized such that an extra processor for computation of the mosaic screens is not required. For this purpose, we allow that the sub-pictures of a mosaic screen can be constructed from previously constructed sub-pictures using the P picture-type syntax for coding of the *mini-slices.* We expect that the aforementioned implementation aspects enable the VTOC implementation on an embedded microprocessor.

An important system aspect of VTOC is the base-layer sub-pictures for the mosaic screen are additionally stored on the recording medium. A good video navigation quality is obtained, by using the VBV-buffer space as a mosaic-screen buffer. This quality corresponds to 15 Mbit/s for a normal video stream. This bit rate was chosen to avoid any visual degradation for large variations in sub-pictures as the sub-pictures are frozen for a number of seconds. With this quality, the additionally required storage capacity is less than 1% of the normal-play storage.

Decoupling of the VTOC from the recording process removes the real-time requirement, thereby relieving possible critical system resources. This offers the possibility to start the generation of a VTOC during recording and to end the VTOC generation after ending of the recording. Assigning a fix bit cost to a mosaic screen simplifies the generation of an MPEG-coded video navigation stream, allowing an embedded MPEG-2 video decoder or even external MPEG-2 decoders to perform the decoding without further notification about the PVR mode of operation.

REFERENCES

[1] C. Buma, R. Brondijk and S. Stan, "DVD+RW: 2-Way Compatibility for Video and Data Applications", *Proceedings of ICCE 2000*, pp. 88-89.

[2] CEI/IEC-61834: "Helical-scan digital video cassette recording system using 6,35 mm magnetic tape for consumer use (525-60, 625-50, 1125-60 and 1250-50 systems", 1998+A1:2001.

[3] P.H.N. de With and A.M.A. Rijckaert, "Design considerations of the video compression system of the new DV camcorder standard", IEEE Trans. Consumer Electronics, Vol. 43, No. 4, pp. 1160-1179, Nov. 1997.

[4] CEI/IEC-60774-1: "Helical-scan video tape cassette system using 12,65 mm (0.5 in) magnetic tape on type VHS, part 1: VHS and compact VHS video cassette system, 1994"

[5] CEI/IEC-60774-1: "Helical-scan video tape cassette system using 12,65 mm (0.5 in) magnetic tape on type VHS, part 5: D-VHS, 2004"

[6] R.J.J. Saeijs, F.J. Jorritsma, "Signal Processing On Information Files So As To Obtain Characteristic Point Information Sequences", Patent application US6871007,1999

[7] ISO/IEC 13818-1: "Information Technology - Generic Coding of Moving Pictures and Associated Audio Recommendation H.222.0 (systems)".

[8] ISO/IEC 13818-2: "Information Technology - Generic Coding of Moving Pictures and Associated Audio Recommendation H.262 (video)".

[9] IEC61883-4: "Digital Interface For Consumer Audio/Video Equipment Part 4: MPEG-TS data transmission".

[10] ISO/IEC 14496-2: "Information Technology Coding of audio-visual objects - Part 2: Visual"

[11] O. Eerenberg, P.H.N. de With, J.P. van Gassel, D.P. Kelly, "MPEG-2 compliant trick play over a digital interface", IEEE Trans. Consumer Electronics, Vol. 51, No. 3, pp. 958-966, Aug. 2005.

[12] ISO/IEC 13818-5: "Information Technology - Generic Coding of Moving Pictures and Associated Audio Recommendation - Software-Simulation".

Onno Eerenberg was born in Zwolle, the Netherlands, in 1966. He graduated from the Polytechnical College in Amsterdam in 1992. He joined Philips Research Laboratories Eindhoven, The Netherlands where he worked in the Magnetic Recording Systems department on digital video and data recording systems. He was involved in several European research projects in this area and was involved in the implementation of e.g. video compression systems. He received a MSc degree in engineering product design in 1998 from the University of Wolverhampton, UK. He is currently working for NXP Semiconductors Research where he is involved in video compression. He holds several US patents and patent applications in the field of digital recording and digital broadcasting.

Peter. H. N de With graduated in Electrical Engineering from the University of Technology in Eindhoven, and received his Ph.D. degree from the University of Technology Delft. Since January 2007 he is IEEE Fellow. He joined Philips Research Labs Eindhoven in 1984, at the Magnetic Recording Systems Dept. and set-up the first DCT-based compression systems. From 1985 to 1993, he was involved in several European research projects on SDTV and HDTV recording. He was the leading video compression expert for the DV camcorder standard in 1989-1993. In 1994, he joined the TV Systems group at Philips Research for leading the design of programmable video architectures, being senior TV systems architect and in 1997, he became full professor at the University of Mannheim, Germany, at the faculty Computer Engineering. From 2000 to 2007, he was principal consultant at LogicaCMG in Eindhoven and he is now with CycloMedia Technology (NL). Since 2000, he is professor at the University of Technology Eindhoven, heading the chair on Video Coding and Architectures. Mr. De With has written and co-authored over 200 international papers, and is holding over 40 international patents. He co-authored several award papers (IEEE CES 1995, 2000, SPIE 2004, etc.). He is program committee member of the IEEE CES, ICIP and SPIE VCIP, vice chair and honorary member of the IEEE Benelux community for IT, advisor to the Dutch imaging school, and board member of various working groups.