Real-time Labeling Methods for MPEG Compressed Video

Gerrit C. Langelaar, Reginald L. Lagendijk, Jan Biemond

Department of Electrical Engineering, Information Theory Group
Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands
E-mail: *{gerhard, lagendijk, biemond}@it.et.tudelft.nl*

Abstract

The objective of the SMASH-project, sponsored by the European Union, is to develop a mass multimedia storage device for home usage. Service providers are reluctant to offer video services in digital form because of their fears for unrestricted duplication and dissemination. Therefore a copy protection system based on labeling techniques is proposed for this digital storage device.

In this paper, two new real-time labeling techniques for compressed video are introduced. The first method adds the label directly in the MPEG-bitstream. Although, this labeling procedure can be performed in a fast and simple way, the label can also easily be removed without affecting the quality seriously. The second method is more computationally demanding but also more robust. This method is based on discarding parts of the compressed MPEG-video stream. The label can be extracted even after the MPEG-stream has been re-encoded using a lower bit-rate.

1. Introduction

The objective of SMASH[*]-project, is to develop a mass multimedia storage device for home usage. The development rate and success of such a digital mass storage system depend not only on technical advances, but also on the existence of an adequate copy protection method. Service providers will not offer high quality video services in digital form without a copy protection method which limits duplication of multimedia data.

Therefore a copy protection system for video based on labeling techniques is proposed. This system should be implemented in the storage device and adds a perceptually invisible "copy-prohibit" label to the data which is stored. Before the storage device starts recording, the incoming data is checked for this label. The recorder will only start if the "copy-prohibit" label is not detected. This means that data can only be stored or copied once. The advantage of this system is that the label can not be removed without affecting the quality of the data and the label also remains intact across different data file formats.

The current work in SMASH concentrates among other things on developing real-time labeling methods for compressed video and images. Several labeling methods for images can be found in literature. Most methods add the label in the spatial [1..5] or the Discrete Cosine Transform (DCT) [6,7] domain. To extract a label some of these methods only use the labeled image, where others use the labeled image together with the original one. For copy protection we can only use the first kind of methods. To perform the labeling in real-time on compressed data, the compressed format must be taken into account, because decoding, labeling and finally re-encoding the data is quite computationally demanding. Only the methods presented in [5] and [6] deal directly with compressed data. In [5] a labeling method is described which incorporates the label in the bitstream domain of MPEG-2 coded video, without decoding and full re-encoding the stream. The label can be retrieved from the decoded video. In [6] a labeling technique is proposed which adds the label in the bitstream domain of JPEG coded images, without decoding and full re-encoding the stream. The label can be extracted from the encoded (compressed) image.

In this paper we first discuss the requirements for a label. Subsequently two new real-time labeling techniques for compressed video are introduced. The first method, described and evaluated in section 3, adds the label directly in the MPEG-1 or 2 bitstream by replacing variable length codes. Since decoding and re-encoding the stream is not necessary, the labeling procedure is not computationally demanding. Although, the labeling process can be performed in a fast and simple way, the label can also be removed easily without affecting the quality seriously.

---

[*] This work was funded in part by the EU ACTS program AC18: SMASH (Storage for Multimedia Applications Systems in the Home)

In section 4, the second method is presented and evaluated. This method is more computationally demanding but is also more robust. The method is based on discarding parts of the compressed MPEG-video stream. For the labeling only partial decoding is needed. No re-encoding is required. The label can be extracted even after the MPEG-stream has been re-encoded using a lower bit-rate. Finally, conclusions are drawn in section 5.

2. Requirements

Labeling or watermarking is a technique to embed additional information into audio or video data by modifying the original data under the constraint that the quality is not seriously affected. For a copy protection system as described in the introduction we need a labeling method that:

- allows the extraction of the label without using the original video data,
- can be performed directly on compressed data to add and extract a label in real-time,
- does not increase the size of the compressed bitstream,
- is perceptually invisible,
- is resistant to simple processing techniques, which do not seriously reduce the quality of the image, like re-encoding the MPEG-stream using a lower bit-rate,
- can store enough label bits per second to detect the copy-prohibit bit at every random position in the stream.

The first three requirements are inevitable. A recorder does not have the original data to its disposal to extract the label. A copy protection system in a consumer storage device should not be too complex and expensive and must operate in real-time. If a spatial labeling method [1..4] is used, the MPEG video stream should be locally decoded, labeled and encoded. Re-encoding the data to add the label is too computationally demanding and expensive, therefore the label must be added in another way, circumventing the most time consuming steps, decoding and encoding. If the size of the compressed bitstream is increased by the labeling, the buffers in hardware decoders can run out of space or the synchronisation between audio and video can be disturbed. The other requirements are related with each other and ranked in order of importance. First, the label must be perceptually invisible, because a copy protection system that spoils the video quality is not acceptable. After that, a trade-off should be made between robustness and label size. The more label bits are added per frame, the less robust the label will be against processing techniques.

The methods discussed in section 3 and 4 rely heavily on the MPEG video compression standard. The discussion of this standard is, however, out of the scope of this paper and the reader is referred to [9] for more information.

3. Fast MPEG Labeling Technique in the Bitstream Domain

In this section a real-time labeling system is described and tested, which has about the same strength as the Serial Copy Management System (SCMS) [8]. It has the additional advantage that it can add much more side-information in the label (e.g. copyright information). Unlike SCMS, this side-information remains intact if the video data is copied to a PC or other device, that does not support the copy protection mechanism.

3.1 MPEG Bitstream Labeling Technique

To make the algorithm as fast as possible the motion vector calculation, DCT and IDCT transforms etc. should be avoided. Therefore, the label should be added on the VLC-level (variable length codes). In our case, the label-bitstream is embedded in the MPEG-stream by changing the least significant bit of some VLC's, that represent quantized run-length coded AC DCT-coefficients. To be sure that the change in the VLC is perceptually invisible and the MPEG-bitstream keeps its original size, the VLC has to meet several requirements. Another VLC must exist, that has the same size, the same run length, and a level difference of 1. According to table B.14 and B.15 of the MPEG-2 standard [9], there are enough suitable VLC's. Besides, all fixed-length-coded DCT coefficients following an Escape-code meet the requirements. Some examples of suitable VLC's are listed in table 1. The VLC's in the intra and inter coded macroblocks can be used in the labeling process. However, in our experiments we only used the VLC's in the intra coded macroblocks.

The labeling procedure works as follows. First an intra coded macroblock is located. The VLC's in this macroblock are tested. If a VLC, which meets the requirements mentioned above, is found, this VLC is replaced by another, whose least significant bit represents the label bit. To extract the label, again the intra coded macroblocks are located. If a suitable VLC is found, the value represented by its least significant bit is assigned to the label bit.

Table 1. Example of some suitable VLC's in table B.14 of the MPEG-2 Standard

Variable length code	VLC size	Run	Level	Parity
0010 0110 s	8 + 1	0	5	1
0010 0001 s	8 + 1	0	6	0
0000 0001 1101 s	12 + 1	0	8	0
0000 0001 1000 s	12 + 1	0	9	1
0000 0000 0011 101 s	15 + 1	1	10	0
0000 0000 0011 100 s	15 + 1	1	11	1

The label can be added in a very fast and simple way, but it can also be removed easily for example by decoding the MPEG-stream and encoding it again using another bit-rate or after applying simple processing techniques, like filtering, etc. However, this is a very time-consuming, computationally and memory (disk) demanding method. By watermarking the stream again using another label, the previous label is removed and the quality is not affected. In fact, the stream can be labeled over and over again without seriously affecting the quality any further.

3.2 Experimental Results

We applied the labeling technique to 10 seconds of MPEG-2 video coded with 25 frames per second. The size of the movie is 720 by 576, the GOP-length is 12 and the video contains smooth areas, textured areas and sharp edges. During the 10 seconds of the video there is a gradual frame-to-frame transition and the camera turns fast to another view at the end. The video was coded using different bit-rates.

Table 2. Total number of VLC's and number of suitable VLC's in the intra coded macroblocks of **10 seconds** MPEG-2 video coded using different bit-rates and the maximum label bit-rate.

Video bit-rate	number of VLC's	number of suitable VLC's	Max. label bit-rate
1.4 Mbit/s	334.433	1.673	0.2 Kbit/s
2.0 Mbit/s	670.381	15.733	1.6 Kbit/s
4.0 Mbit/s	1.401.768	42.784	4.3 Kbit/s
6.0 Mbit/s	1.932.917	62.877	6.3 Kbit/s
8.0 Mbit/s	2.389.675	82.397	8.2 Kbit/s

It was possible to store up to 8 Kbit of label information per second in the MPEG streams. Informal subjective tests showed that the labeling does not result in any visible artefacts in the streams coded at 4, 6 and 8 Mbit/s, see Table 2. However, it was not possible to evaluate the quality of the video streams coded at 2 or less than 2 Mbit/s, because the unlabeled MPEG-streams already contained too many coding artefacts. The label bit-rate can be increased by also taking the inter coded macroblocks into account.

In figure 1a the original I-frame of an MPEG-2 coded movie is represented. If the original I-frame is subtracted from the corresponding labeled I-frame and this absolute difference signal is amplified by 60, the images, shown in figure 1b and 1c, are obtained. Since more bits are stored in an I-frame of a stream coded at 8Mbit/s more differences are made (see Figure 1c).

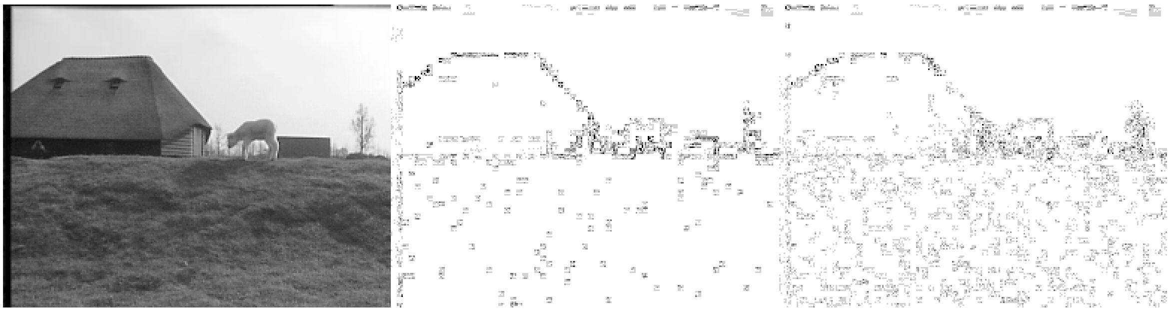

Figure 1. (a) Unlabeled I-frame **(b)** Frame difference (4Mbit/s) **(c)** Frame difference (8Mbit/s)

4. Re-encoding Resistant MPEG Labeling Technique

The labeling algorithm described in this section is more complex, but also more robust against re-encoding the video using a lower bit-rate. The algorithm does also not require re-encoding of the compressed video stream to add the label, since the label is embedded by discarding parts of the compressed stream. A similar approach was proposed in [6] for compressed still images.

4.1 Basic Idea

Each bit out of the label string has its own area in an I-frame For instance the first bit is located in the top-left-corner of the I-frame, in a pixel area of *n* 8x8 pixel blocks, see Figure 2 (*n*=16).

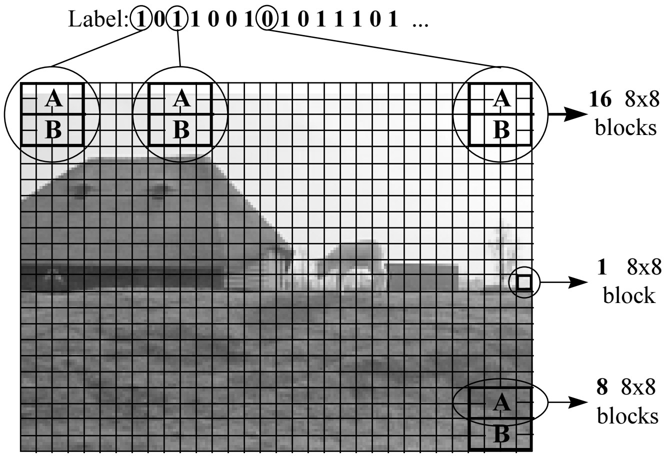

Figure 2. Bit positions and block definitions in a frame.

A label bit is embedded in an area by introducing an "energy" difference in the high frequency DCT coefficients of areas *A* (see Figure 2), the upper half of this pixel area and *B* the lower half. In Figure 3 the procedure to calculate the difference in energy *D* of a 32x32 pixel area is explained. For all 8x8 blocks in *A*, the Discrete Cosine Transform is calculated. Then the squared sum of a particular subset of coefficients is computed.

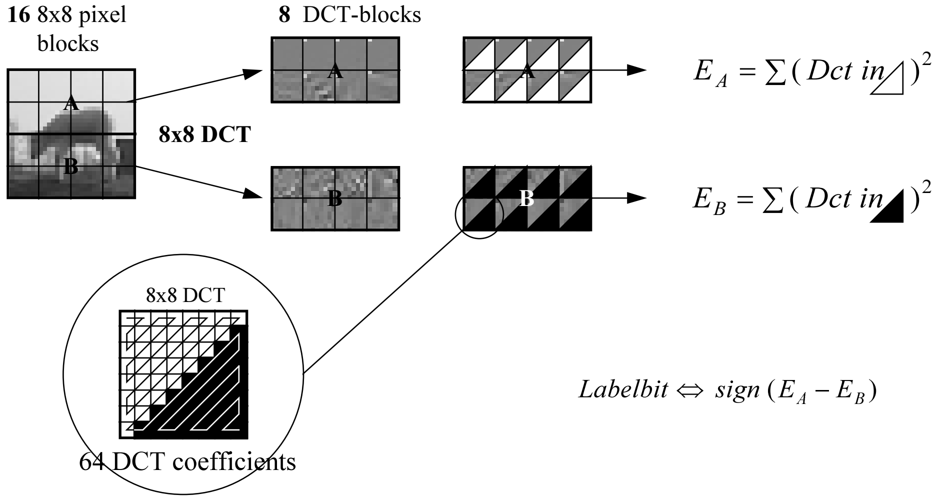

Figure 3. Extracting a label bit from an *n*x*n* pixel area.

This subset is denoted by S_A, and is illustrated in Figure 3 by the white area in the DCT blocks. We define the total energy of S_A, computed over all blocks in area *A*, as:

$$E_A = \sum_{k \in A} \sum_{i,j \in S_A} DCT(i,j)^2$$

Similarly E_B is computed for area B (indicated in Figure 3 as black triangles). S_A and S_B are typically defined according to a cut-off point c in the zig-zag scanned DCT-coefficients. This implies that S_A and S_B have a triangular shape.

The label bit value is now defined by the sign of the difference between the two energies E_A and E_B. Label bit "0" is defined as a positive energy difference and label bit "1" as a negative one.
The label embedding procedure must adapt E_A and E_B to manipulate the energy difference D. If label bit "1" must be embedded, the energy in A is removed and the difference becomes negative:

$$D = E_A - E_B = 0 - E_B = \boldsymbol{-E_B}$$

If label bit "0" must be embedded, the energy in B is removed and the energy becomes positive:

$$D = E_A - E_B = E_A - 0 = \boldsymbol{+E_A}$$

There are several reasons for using this energy difference in the triangularly shaped areas (Figure 3). The most important reason is that calculating D and adapting E_A and E_B can easily be done in the compressed stream. Another advantage is that the human eye is less sensitive to changes in the high frequency coefficients located in those areas. In the compressed bitstream the DCT-coefficients are ordered into a "zig-zag" sequence (see Figure 4) and run-level encoded. Variable-length codes (VLC's) are assigned to these run-level codes. All DCT-coefficients, needed for the calculation of E_A or E_B are conveniently located at the end of the compressed 8x8 DCT block after zig-zag ordering. The coefficients can easily be set to zero to adapt the energy without re-encoding the stream. Simply shifting the end of block marker (EOB) towards the DC-coefficient by skipping the last VLC's of the block changes the energy E (Figure 4).
Because coefficients are "thrown away" to add a label, the labeled compressed stream will be always smaller than the unlabeled stream, and synchronisation and buffer overflow problems are avoided in this way.

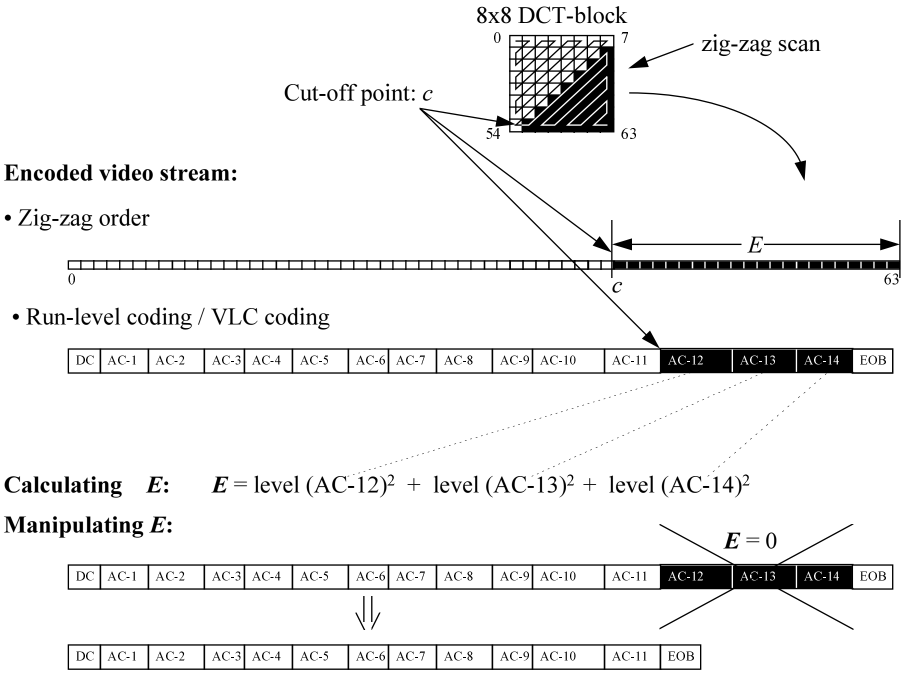

Figure 4. Calculating and adapting the energy in an 8x8 compressed DCT block

4.2 Calculating the Cut-off Point

The question is how many DCT coefficients must be discarded to generate a certain energy difference between the two areas *A* and *B* (see Figure 3.). For smooth areas *A*+*B* a low cut-off point *c* is required to find DCT-coefficients which are non-zero. For more textured areas *A*+*B* a higher *c* is needed, otherwise too many coefficients are set to zero and the label bit will become visible. Problems occur if one area e.g. *A* does not contain any detail and the other contains much texture (see Figure 5a). If a positive energy difference $D = E_A - E_B$ must be generated to embed label bit "0", almost all AC-DCT coefficients in area *B* must be removed by using a very low cut-off point *c* to make $E_A > E_B$. Figure 5 illustrates this situation, clearly the presence of the label becomes visible in area B.

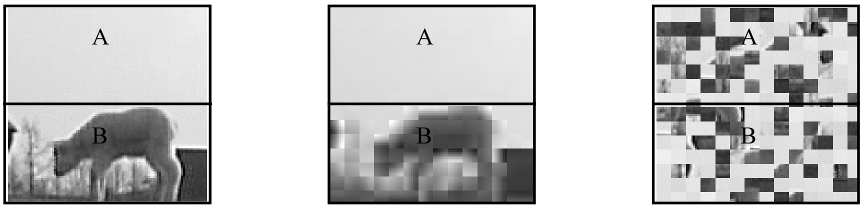

Figure 5. **(a)** smooth and textured area **(b)** positive energy difference enforced $E_A > E_B$ **(c)** shuffled frame

This problem can be solved by pseudo-randomly shuffling all 8x8 DCT-blocks of the I-frame before labeling (Figure 5c). If the I-frame is not shuffled, the cut-off points *c* can range from 1 for smooth blocks, where almost no energy $E_{A(c)}$ or $E_{B(c)}$ can be found, to 63 for textured blocks. If all 8x8 DCT-blocks of the I-frame are first shuffled, smooth 8x8 blocks and textured 8x8 blocks will alternate in blocks *A* or *B*. The high frequency DCT coefficients are now distributed equally over all areas *A* and *B*, and the cut-off points will have a smaller range. The *c*-values in the shuffled frames vary less around the average of the *c*-values than in the unshuffled frames. So, the chance that a very low or a very high cut-off point *c* will be used is smaller. An extra advantage of shuffling is that each label-bit is scattered over the frame, which makes it more difficult for a hacker to locate the label bits.

4.3 The Fully Automated Labeling and Extraction Procedure

Labeling procedure:

The parameter *D* is the energy difference which will be enforced to add a label bit. This parameter influences the amount of coefficients, which are discarded, and determines together with parameter *n* the robustness of the label. *n* indicated the number of 8x8 blocks per label bit and L_i $_{i=0,1,2,...}$ is the label bit string.

1. All 8x8 DCT-Y-blocks of the I-frame are pseudo-randomly shuffled. The label bit counter *i* is set to 0.

2. A block *A*+*B* containing *n* 8x8 Y-DCT-blocks is selected from the I-frame to embed label bit L_i.
   This block is split into an upper half *A* (*n*/2 blocks) and a lower half *B* (see Figure 2).

3. To estimate the cut-off point c_e for this block *A*+*B*, the squared sum $E_{A(c)}$ of all DCT-coefficients after cut-off point $c = 1$ (see Figure 4) in all 8x8 DCT-blocks in *A* is calculated. $E_{B(c)}$ is calculated in the same way for block *B*. This procedure (3) is repeated for $c = 2,3, ..., 63$.

4. c_e is set to the highest cut-off point *c* (1..63) for which ($E_{A(c)} > D$) AND ($E_{B(c)} > D$)

5. If L_i is 1, all coefficients after c_e in all 8x8 DCT-blocks of block *A* are set to zero (coefficients in S_A).
   If L_i is 0, all coefficients after c_e in all 8x8 DCT-blocks of block *B* are set to zero (coefficients in S_B).

6. The label-bit counter *i* is increased by 1. The procedures (2..6) are repeated until all label bits L_i are embedded.

7. All 8x8 DCT-Y-blocks are shuffled back to their original locations.

Label extracting procedure:
For the extraction procedure a fixed threshold T is required. The threshold T influences the determination of the cut-off point c, which is used to extract each label bit. This value must be smaller than the enforced energy difference D.

1. All 8x8 DCT-Y-blocks of the I-frame are pseudo-randomly shuffled. The label bit counter i is set to 0.

2. A block $A+B$ containing n 8x8 Y-DCT-blocks is selected from the I-frame to extract label bit L_i.
   This block is split into an upper half A ($n/2$ blocks) and a lower half B (see Figure 2).

3. To estimate the cut-off point c_e for this block $A+B$, the squared sum $E_{A(c)}$ of all DCT-coefficients after cut-off point $c=1$ (see Figure 4) in all 8x8 DCT-blocks in A is calculated. $E_{B(c)}$ is calculated in the same way for block B. This procedure (3) is repeated for $c = 2,3, ..., 63$.

4. c_A is set to the lowest cut-off point c (1..63) for which ($E_{A(c)} < T$).
   c_B is set to the lowest line number c (1..63) for which ($E_{B(c)} < T$).

5. If ($c_A < c_B$) the label bit L_i embedded in the block $A+B$ is one.
   If ($c_A = c_B$) AND ($E_{A(cA)} < E_{B(cA)}$), L_i is also one.
   Otherwise L_i is zero.

6. The label bit counter i is increased by 1. The procedures (2..6) are applied to all blocks $A+B$ until all bits of the label are extracted.

4.4 Experimental Results

We applied the labeling technique presented in section 4.3 to 10 seconds of MPEG-2 compressed video using different bit-rates. The parameter D was set to 20 and the parameter T to 15. The cut-off points c for each label bit were allowed to vary in the range from 6 to 63.
A similar algorithm using 16 8x8 blocks per bit (n=16) was successfully applied to JPEG compressed images [6]. However, using this value in the MPEG labeling algorithm resulted in blocking artefacts around edges of smooth objects. Informal subjective tests showed that the label, embedded with n=32, was not noticeable in video streams coded at 8 and 6 Mbit/s. If MPEG streams coded at a lower bit-rate, are labeled with n=32 the same artefacts appear. By increasing n further to 64 a perceptually invisible label can also be embedded in an MPEG stream coded at 4 Mbit/s. In Table 3 some results are listed. It appears that not enough high frequency coefficients can be found in the sequence coded at 1.4Mbit/s to create the energy differences D for the label bits.

Table 3. Amount of 8x8 DCT blocks per bit, amount of bits discarded by the labeling process, % label biterrors and label bit-rate for the Sheep-sequence coded with different bit-rates.

Video bit-rate	*n*	Discarded bits	% Bit errors	Label bit-rate
1.4 Mbit/s	64	1.6Kbit/s	24.6	0.21 Kbit/s
2.0 Mbit/s	64	4.7Kbit/s	0.1	0.21 Kbit/s
4.0 Mbit/s	64	3.8Kbit/s	0.0	0.21 Kbit/s
6.0 Mbit/s	32	7.6Kbit/s	0.0	0.42 Kbit/s
8.0 Mbit/s	32	7.1Kbit/s	0.0	0.42 Kbit/s

In figure 6a the original I-frame of an MPEG-2 coded movie is represented. If the original I-frame is subtracted from the corresponding labeled I-frame and this absolute difference signal is amplified by 60, the images, which are shown in figure 6b and 6c, are obtained.

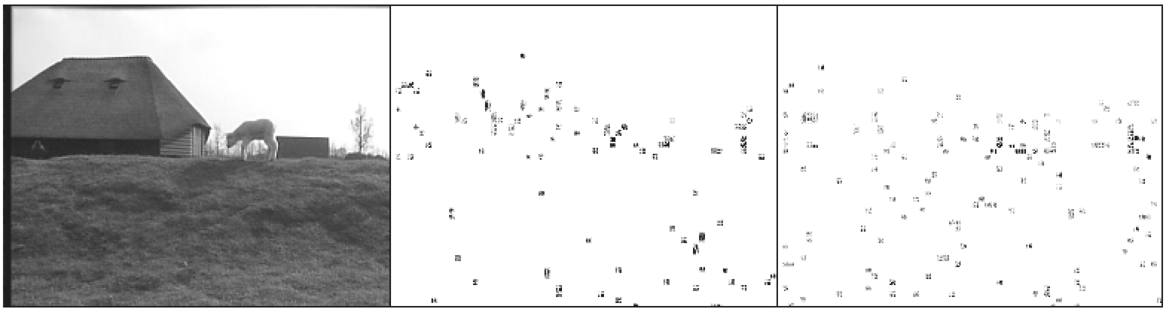

Figure 6. **(a)** Unlabeled I-frame **(b)** Frame difference (*n*=64, 4Mb/s) **(c)** Frame difference (*n*=32, 8Mb/s)

To see if the label is resistant to re-encoding using a lower bit-rate, the following experiment is performed. The sheep-sequence is MPEG-2 encoded at 8 Mbit/s and this compressed stream is labeled (*n*=32). After that, the labeled sequence is decoded and again MPEG-2 encoded using the same GOP-length but with different lower bit-rates. The biterrors introduced by decreasing the bit-rate are represented in Figure 7.

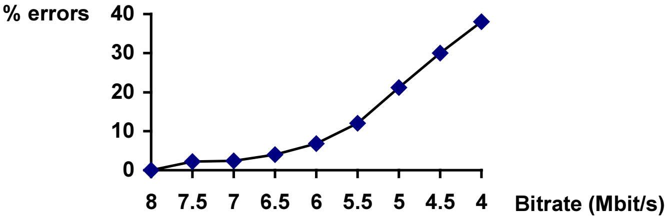

Figure 7. % Biterrors after re-encoding a labeled 8Mbit/s Mpeg-2 sequence with a lower bit-rate.

5. Conclusions and Future Work

In this paper two new techniques for real-time labeling of compressed video are presented. There is clearly a trade-off between complexity, label bit-rate and robustness. The first method is not computationally demanding, has a very high label bit-rate, which can be extended further by taking also the inter-coded macroblocks into account, but is not robust against re-encoding. The second method is more complex and has a lower label bit-rate, but is resistant to re-encoding if the same GOP structure is used. To survive re-encoding using another GOP-structure a different approach should be followed. The label extraction process can for instance take place after decoding in the spatial domain.

References

[1] W. Bender, D. Gruhl, N. Morimoto : "Techniques for Data Hiding", Proceedings of the SPIE, 2420:40, San Jose CA, USA, February 1995

[2] I. Pitas, T. Kaskalis : "Signature Casting on Digital Images", Proceedings IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, June, Greece, 1995

[3] J.R. Smith, B.O. Comiskey, "Modulation and Information Hiding in Images", Preproceedings of Information Hiding, an Isaac Newton Institute Workshop, University of Cambridge, UK, May 1996

[4] G.C. Langelaar, J.C.A. van der Lubbe, J. Biemond : "Copy Protection for Multimedia Data based on Labeling Techniques", 17th Symposium on Information Theory in the Benelux, Enschede, The Netherlands, May 1996

[5] F. Hartung and B. Girod: "Digital Watermarking of Raw and Compressed Video", Proc. European EOS/SPIE Symposium on Advanced Imaging and Network Technologies, Berlin, Germany, Oct. 1996.

[6] G.C. Langelaar, J.C.A. van der Lubbe, R.L. Lagendijk: "Robust Labeling Methods for Copy Protection of Images", Proc. Storage and Retrieval for Image and Video Databases V, February 1997, San Jose (CA), USA

[7] J. Zhao, E. Koch : "Embedding Robust Labels into Images for Copyright Protection", Proceedings of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies, Vienna, Austria, August 1995

[8] Digital Audio Interface, International Standard IEC 958

[9] ISO/IEC 13818-2:1996(E), Information technology - Generic coding of moving pictures and associated audio information:video, International Standard, 1996