UNIVERSITÄT SALZBURG

Inter-Frame H.264/CAVLC Structure-Preserving Substitution Watermarking

Thomas Stütz Florent Autrusseau[a] Andreas Uhl

[a]LUNAM Université, Université de Nantes, France

Technical Report 2013-02 April 2013

Department of Computer Sciences

Jakob-Haringer-Straße 2
5020 Salzburg
Austria
www.cosy.sbg.ac.at

Technical Report Series

Inter-Frame H.264/CAVLC Structure-Preserving Substitution Watermarking

Thomas Stütz, Florent Autrusseau and Andreas Uhl

***Abstract*—In this work we propose a novel H.264/CAVLC structure-preserving substitution watermarking algorithm. The proposed watermarking algorithm enables extremely efficient watermarking by simple bit substitutions. Furthermore our watermarking algorithm can be applied in applications scenarios, that require exact structure-preservation. The quality and robustness of the approach are in depth evaluated and analyzed, the quality evaluation is backed up by subjective evaluations. Comparison to the state-of-the-art indicates a superior performance of our watermarking algorithm.**

I. Introduction

H.264 watermarking has been researched intensely and is of great interest due to its wide applicability in the context of DRM (digital rights management). This paper presents a novel H.264 CAVLC watermarking technique that allows to implement watermarking by simple and efficient bit substitutions of the compressed bitstream (substitution watermarking). Additionally our algorithm is structure preserving, i.e., precisely preserves the length of the bitstream and even of the bitstream's smaller units. In the case of H.264, structure preserving watermarking denotes watermarking algorithms in which the network-abstraction layer units (NAL units / NALUs are small units which form the entire H.264 bitstream) have exactly the same length in the watermarked works and the original / cover work. While there is a considerable amount of scientific literature on structure-preserving watermarking for the MPEG-2 format [6][12][14], the literature on H.264 CAVLC structure-preserving watermarking is far less extensive, only one approach can be employed for structure-preserving H.264 CAVLC watermarking [23] and only one structure-preserving CABAC watermarking [22]. The preservation of the exact bitstream structure for H.264 is required for the watermarking of Blu-Ray content, while structure-preserving MPEG-2 watermarking can be employed for the watermarking of DVD content. The length preservation is required as the video has to fit on a Blu-Ray disc / DVD disc. The internal structure has to be preserved as often byte-based addressing schemes are employed in production and presentation, e.g., the meta-data on Blu-Ray employs byte-based addressing schemes and even more import additional content, that can be downloaded to enhance the Blu-Ray content, employs byte-based addressing (BD-J) as well. So there are important applications, which require structure-preserving H.264 watermarking. In our proposed watermarking algorithm the embedding stage is split into an analysis and a substitution stage; analysis must only be conducted once, afterwards the embedding of different marks requires only extremely light-weight bit substitutions. Thus the embedding of numerous marks in real-time with very low computational complexity is possible; of utmost importance for streaming individually marked content to numerous clients. Our main contribution is the proposal of a new informed (non-blind) structure preserving H.264 CAVLC watermarking approach. A further contribution is the thorough analysis of the approach with respect to robustness and quality. The quality evaluation not only employs state-of-the-art quality assessment tools, but also conducts actual subjective quality evaluation. Our evaluations focus on 720p content (the leading mobile phone's resolution and also occasionally employed for Blu-Ray content).

In section II an overview of H.264 is given, while section III briefly summarizes quality evaluation of visual data. Our structure-preserving H.264 CAVLC watermarking approach is presented in section IV. Details on the detection process are explained in section V. Experimental results with respect to quality and robustness are presented in section VI. A brief review and comparison to the state-of-the art of H.264 watermarking with a focus on structure-preserving watermarking is presented in section VII. Finally section VIII concludes the paper.

T. Stütz and A. Uhl are with the Department of Computer Sciences, University of Salzburg, 5020 Salzburg, Austria, e-mail: thomas.stuetz@fh-salzburg.ac.at, uhl@cosy.sbg.ac.at

F. Autrusseau is with the LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597, Polytech Nantes, rue Christian Pauc, BP 50609, 44306 Nantes, France

II. Overview of H.264

The design of H.264 follows the classic hybrid video coding approach. The frames are processed in 16x16 macroblocks. Each macroblock can be predicted using previously processed macroblocks of the same frame (intra-prediction) or other frames (inter-prediction). The macroblocks can be further subdivided (sub-macroblock partitions), the smallest block size is 4x4. A coded video sequence always starts with the coded data of an intra-predicted frame (I frame) The distortions of I frames spread on all subsequently decoded frames due to inter prediction. Following an I frame inter-predicted frames that may use one reference frame (P frame) or two reference frames (B frame) follow. Inter-prediction is conducted by motion estimation and motion compensation, which are conducted with quarter pixel accuracy. The motion vectors (MVs) of a block are predicted by neighbouring blocks (a detailed description can be found in [10]) and the motion vector difference (MVD) is actually coded in the bitstream (which codes quarter pixel differences). There are two distinct coding modes in H.264, namely CAVLC and CABAC. CAVLC is computationally less

Index	Codeword	MVD
0	1	0
1	010	1
2	011	-1
3	0010	2
4	0011	-2
5	0010	3
6	0011	-3
7	00010	4
8	00011	-4
9	00010	5
10	00011	-5
11	00010	6
...	...	...

TABLE I
CAVLC: CODING OF MVDS

expensive (at the cost of a lower compression performance) and thus is employed in cases where computational complexity constraints outweigh the compression performance. Typical applications are in the context of mobile devices (720p has become the resolution of the lead devices), where power and computational constraints outweigh compression as well as the coding of 720p content for Blu-Rays, which simply does not require higher compression as 720p typically fits on a Blu-Ray anyway. In H.264/CAVLC MVDs are not coded context-adaptively, but with variable-length signed exponential Golomb codes. Table I shows the coding of MVD values, each MVD (last column) is coded by an exponential Golomb code (in the column labelled "Codeword"). A separate MVD is coded for the x- and y-direction.

III. Quality Assessment

As briefly mentioned in the introduction, this work mainly focuses on evaluating the robustness and quality performances of an H.264 CAVLC watermarking operating within the compressed bitstream. In this section, we present both subjective and objective quality assessment of watermarked contents.

A. Subjective Quality Assessment

Ultimately, the quality of the marked content will be judged by human observers, and thus, running a subjective experiment is the best way to evaluate the impact of the watermark on the quality of the protected image/video. During subjective tests, quality (or annoyance) scores are collected from human observers within a controlled environment. Subjective experiments have been of high interest for many decades among the scientific community. Early experiments were conducted to determine an optimal viewing distance on television monitors [11], or the detection threshold of a simple spot on a CRT screen [1], and led the researchers to do an attempt to estimate the subjective quality [2]. Evidently, subjective experiments had to be standardized, in order for other researchers to be able to reproduce and/or compare the results. Thus, the International Telecommunication Union (ITU) has published various reports and recommendations for conducting subjective experiments. Both the Radiocommunication (ITU-R) and Telecommunication (ITU-T) sectors of the ITU are regularly issuing some recommendations for quality assessment (both Objective and Subjective) of digital images and videos. Two recommendations of particular interest are [8] and [9]. The recommendation [9] notably specifies the viewing conditions, monitor settings (resolution, contrast), the importance of anchoring is highlighted, it is for instance advised to use at least 15 non expert observers. Some advices are given on the duration of the experiment, and on the possible protocols to use. Among the most commonly used protocols, we can cite the "Double-Stimulus Impairment Scale" (DSIS) and the "Double-Stimulus Continuous Quality Scale" (DSCQs). In [8], alternative protocols are suggested, the "Absolute Category Rating" (ACR) or the "Pair Comparison" methods are amongst the most common methods. Commonly, the outcome of a subjective experiment is to collect the Mean Opinion Scores (MOS) from the observers for the given input subjective dataset. The MOS are simply computed by averaging the collected scores of all observers for a given content. The alternative to subjective experiments is to utilize Objective Quality Metrics (OQM), which are methods whose goal is to predict the perceived quality. In the upcoming sub-section, we will review the most common types of OQMs.

B. Objective Quality Assessment

Objective Quality Metrics are mainly of two types. On one hand, statistical quality metrics are very widely used, PSNR, RMSE, or SSIM belong to this category. On the other hand, advanced HVS-based OQMs (such as VIF[19], VSNR[5], CPA[17] or C4[3]) exploit some properties of the Human Visual System (such as contrast sensitivity, contrast masking, or luminance adaptation) to provide a prediction of the MOS (predicted Mean Opinion Scores are commonly referred to as MOSp). Thus, once the subjective scores are collected (MOS are gathered), and the MOSp computed for a given set of metrics, a metric performance evaluation is performed. The Video Quality Experts Group (VQEG) issued a report in 2008 [20] providing an analysis of various assessment methods as well as several tools that can be used to evaluate the performances of Objective Quality Metrics. Statistical or advanced HVS metrics could be either full reference (FR) or reduced reference (RR) or even no reference (NR). For a FR metric, the original image is needed as an input, along with the distorted image which needs to be assessed. A RR metric needs the image to be assessed along with a reduced set of features from the original image to compute the MOSp. Finally, NR metrics only need the distorted image as an input in order to provide a prediction. Usually, FR metrics exhibit better performances at predicting the MOS. In the following, only FR metrics are used.

IV. A Novel Inter Frame Substitution Watermarking Algorithm

Our proposal for a novel H.264 CAVLC watermarking algorithms takes advantage of MVD modifications. MVDs

are coded with signed exponential Golomb codes as illustrated in table I. There are groups of equal-length signed exponential Golomb code words for MVDs, which can be further subdivided in MVDs with positive or negative sign. Our watermarking algorithm modifies original MVDs such that the MVD of the watermarked work has equal length and equal sign as the original MVD.

The developed watermarking algorithm is robust, informed (a small text file is required for detection), and zero-bit (only the presence of a watermark will be detected) [7]. In this paper we only present results for watermarking non-reference inter frames.

A. Embedding

The embedding consists of two stages: an analysis stage and a substitution stage (see fig. 1). Input to the embedding are the H.264 bitstream to be watermarked and the watermarking parameters, such as the key for random watermarking bit generation and a quality control parameter MbDist, which is used to control the embedding distortion in terms of MSE. Output of the process are the watermarked bitstream and a small text-file for detection ("Detection Info" in fig. 1). In the analysis stage each macroblock is checked whether is is suitable for watermarking, thereby several conditions have to be met such that a macroblock is employed for watermarking. Only inter predicted macroblocks are considered for watermarking, and each length-preserving and sign-preserving MVD change is evaluated. Table I shows equal length and equal sign MVD difference codes (exponential Golomb codes). Only codes with the same sign (either codes contained in dashed red boxes or solid blue boxes) and same length are evaluated. First the quality is checked (after application of the change) which is done by computing the MSE (mean squared error) between the original macroblock (with the original MVD) and the modified macroblock. Only if the obtained MSE is below the quality control parameter MbDist the change is considered valid. With the parameter MbDist the embedding strength can be adjusted. Second the impact on the feature (avg. luminance) is computed, which is later used in the detection process. If a macroblock has two sets of changes (one that increases the feature and one that decreases the feature), it is employed for watermarking. The change with the strongest increase is employed to encode a 1, while the change with strongest decrease is employed to encode a 0. The macroblock position and frame number and the original feature are recorded in detection info. The watermark embedding algorithm is summarized briefly as follows.

For each inter-predicted macroblock:

- Evaluate original block's feature (avg. luminance)
- Apply length-and-sign-preserving MVD change and check
  - Embedding distortion (MSE, MbDist)
  - Feature (avg. luminance)
- If there are two groups of changes (increase feature, decrease feature) use the 'decrease feature' change to encode a zero, and the 'increase feature' change to encode a one.

B. Detection

As the presented approach is informed (non-blind), we can assume perfect registration / alignment (temporal and spatial). The actual implementation of registration is well covered in computer vision literature and can for example be solved by storing SIFT features [13] of the watermarked frames as well as the detection information. These features can later be used to register the content.

The detection process can be divided into three distinct tasks, bit extraction, correlation and decision. The bit extraction takes advantage of the detection info, it computes the feature (avg. luminance) of a possibly watermarked macroblock and compares it to the recorded original feature. If the computed feature is larger, a 1 is extracted, if it is smaller a 0 is extracted. In the correlation step, the extracted bit sequence $\vec{e}$ is compared to the possibly embedded watermark bit sequence $\vec{w}$. More precisely, the detector response z is computed by $z = 1/n_{bits} \sum_i ((e_i - 0.5) \times (w_i - 0.5))/0.25$. Finally depending on the detector response and a user-defined probability of false alarm a decision is made. Thereby the detector response is compared to a detection threshold $\mathcal{T}(p_{fp})$, and if the detector response is larger than $\mathcal{T}$ the watermark is decided to be present. The user-defined probability of false alarm (also referred to as false positive probability) determines how likely is it to detect the watermark in a work that has not been watermarked. Overall the algorithm for detection can be briefly summarized as follows:

For each possibly watermarked macroblock in the image domain:

- Compute the block feature (avg. luminance) of the possibly watermarked macroblock and compare the block feature to the original feature and return 0 for decrease and 1 increase.
- Compute detection statistic (a measure of correlation between embedded and extracted sequence).
- Decide whether watermark is present or not.

V. Detector Response Analysis / Decision

After embedding and without any distortions (e.g. recompression) the detector response will always be 1, as we do not have any inaccuracies / randomness in the embedding and detection processes and thus each bit will be correctly extracted. The watermark bits W_i can be viewed as a random variable which is drawn from a uniform random distribution on $\{0, 1\}$ (the watermark bit sequence is generated by a random number generator). If the detector is run on the same work that has been watermarked with another key (or has not been watermarked at all), the extracted bits E_i can also be viewed as random variable that is uniformly distributed on $\{0, 1\}$. If any of the two or both (watermarking bits or extracted bits) are interpreted as random variables the detection response z can also be modeled by a random variable Z of which we can determine the distribution. The distribution of Z can be divided into two cases, the watermark has not been embedded ($\mathcal{H}_0$, no watermark embedded) and the watermark has been embedded ($\mathcal{H}_1$, watermark embedded).

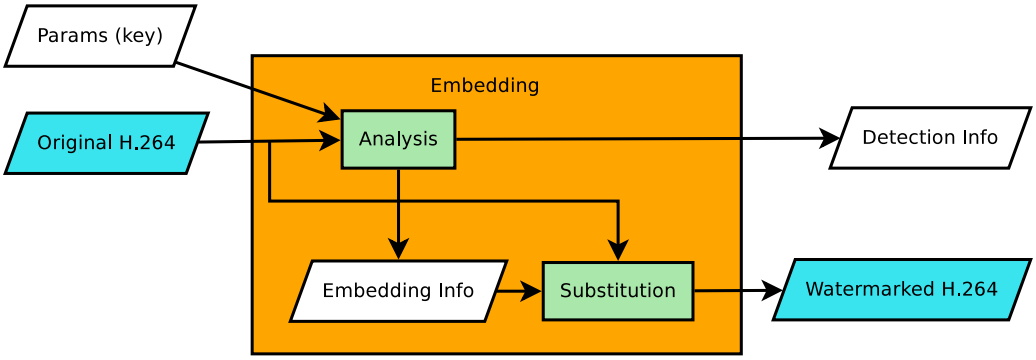


Fig. 1. Watermark embedding

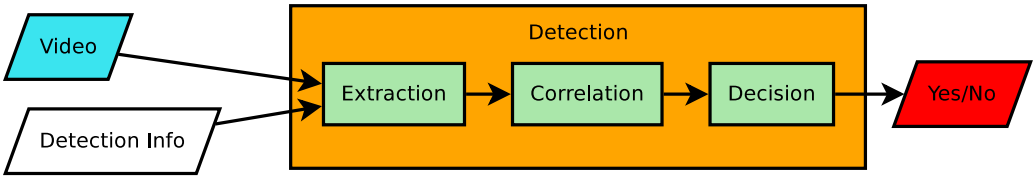


Fig. 2. Watermark detection

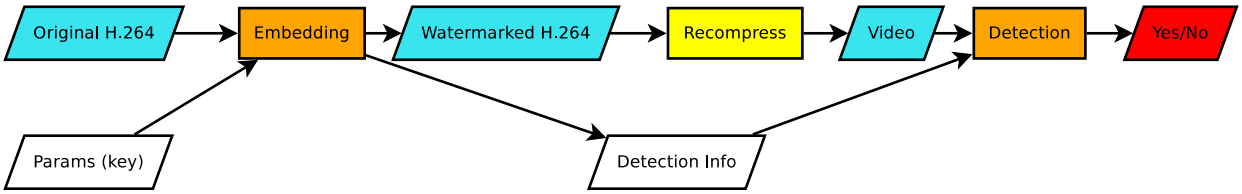


Fig. 3. Watermarking overview

If there are no distortions, the distributions of Z under $\mathcal{H}_1$ is simple (always 1), but distortions from recompression and other signal processing operations will introduce errors, that shift the mean detector response for watermarked content towards 0. Thus we need to determine a threshold for a given probability of a false positive and thus the accurate modelling of Z under $\mathcal{H}_0$ is necessary. For our detection statistic

$$Z = \frac{1}{n_{bits}} \sum_{i=1}^{n_{bits}} \frac{(E_i - 0.5) \times (W_i - 0.5)}{0.5 \times 0.5}$$

the distribution under $\mathcal{H}_0$ is closely related to a Bernoulli distribution with $p = 0.5$, the normalization is done for convenience as the watermarking community is used to detector responses in the range of [-1, 1] as obtained by normalized correlation or the linear correlation coefficient [7, p.103, last sentence]. These correlations are also very similar, but we do not estimate μ and σ of E_i and W_i (0. and 0.5 respectively) and thus the analysis of the distribution of the detection statistic Z remains simple and elegant.

The detection problem can be viewed as a hypothesis testing problem:

$\mathcal{H}_0$: The work is not watermarked.
$\mathcal{H}_1$: The work is watermarked.

In our application the control of the false positive probability is of ultmost importance, thus we need to determine the PDF of Z under $\mathcal{H}_0$, denoted by $p_0(z)$. The probability of a false positive, p_{fp} is then simply obtained as follows:

$$p_{fp} = \int_{\mathcal{T}}^{\infty} p_0(z)$$

Thereby $\mathcal{T}$ denotes a threshold which is used in the final decision whether the watermark is present or not (`T` in the pseudo code below). Conversely the probability of a false negative can be obtained with the aid of the PDF Z under $\mathcal{H}_1$, denoted by $p_1(z)$.

$$p_{fn} = \int_{-\infty}^{\mathcal{T}} p_1(z)$$

The parameter that shapes the distribution of Z under $\mathcal{H}_0$ is the number of embedded bits (n_{bits}), which is both video source and watermarking parameter dependent (the higher the allowed distortion for embedding, the higher the number of bits, the more movement in the video the more bits can be embedded). More precisely the standard deviation of Z under H_0 can be explicitly given as a function n_{bits} (the mean $\mu_0 = 0$):

$$\sigma_0 = \frac{1}{\sqrt{n_{bits}}}$$

For a given number of bits `n`$= n_{bits}$ and a given probability of false alarm `pfp`$= p_{fp}$ the function `getT` (Python pseudo

code) returns the threshold for which the probability that Z is larger than that threshold under $\mathcal{H}_0$ is lower than `pfp`.

```
def getT(n, pfp):
    """Compute the threshold for a given probability
    of false alarm (pfp) and for a given number of
    bits (n)."""

    c_p = 0.
    c_k = n
    p = 1./2.

    while c_p < pfp:
        c_p = c_p + prob(p, n, c_k)
        c_k = c_k - 1

    return xT(n, c_k + 1)

def prob(p, n, k):
    return comb(n, k) * p**k * (1.-p)**(n-k)

def xT(n, k):
    return 2. / n * k - 1.
```

The function `comb(n, k)` is equivalent to $\binom{n}{k}$. On the other hand, we can for every obtained detector response, give precisely the associated probability of false alarm, i.e., we know exactly how likely a false positive will be if we decide that a watermark is present. The function `getPfp` computes the probability for a given detector response `T` and number of embedded bits `n`.

```
def getPfp(n, T):
    """Compute the Probability of false positives
    if we say that a watermark is present for the
    given detector response T and n bits."""

    p = 1./2.

    cT = 1 + 0.5
    cPfp = 0
    ck = n
    while cT > T:
        cPfp = cPfp + prob(p, n, ck)
        ck = ck - 1
        cT = xT(n, ck)
    return cPfp
```

Thus we can concisely determine the probability of a false positive. While we think that the random-watermark scenario [7] is most appropriate for informed detection, the analysis of the random-work scenario and the random-work-random-watermark scenario is analog (those who have concerns on the assumption of uniform randomness of the extracted bits in the random-work scenario can randomize the extraction process).

Figure 4 gives an overview how the number of embedded bits and the probability of false alarm determine the appropriate threshold for detection. The more bits are embedded the lower the threshold for a given probability of false alarm can be chosen.

VI. Experiments

In the following we present results on the basis of 4 different 720p sequences (Canal, Depart, Ebu, Elephant) with 250 frames. The sequences reflect different natural content, as well as one artificially generated sequence (Elephant). The sequences have been encoded with H.264 CAVLC using an I(BP)* prediction structure with non-reference B-frames, which is a reasonable configuration. The Blu-ray specification even requires B-frames to be non-reference frames. Only P16x16 macroblocks (having no sub-partitions) of B-slices have been employed for watermarking. The search range for feasible MVD changes has +/- 16 in each direction and the MVD change with the maximum feature difference below the distortion threshold was selected. Furthermore we rejected MVD changes that did not modify the average luminance feature by more than 0.25.

Sequence / MbDist	100	25	4
Canal	1540	814	98
Depart	2942	1384	24
Ebu	4072	2773	113
Elephant	2146	2109	1113

TABLE II
Total number of embedded bits for each sequence and embedding distortion

Sequence / MbDist	100	25	4
Canal	6.16	3.256	0.392
Depart	11.768	5.343	0.960
Ebu	16.288	11.092	0.452
Elephant	8.584	8.436	4.452

TABLE III
Average number of embedded bits per frame for each sequence and embedding distortion

Our watermarking algorithm can be employed with different values of the embedding distortion parameter MbDist. Both the MbDist and the source material (video) have an impact on the number of bits that can be embedded (see tables II and III for results on our test sources). The encoder decides on the basis of the source video which and how many macroblocks are encoded as P16x16 blocks. The analysis stage of watermark embedding only chooses MVD changes which result in a macroblock distortion (in MSE) that is below MbDist. Therefore the reduction of MbDist severely reduces the set of feasible MVD changes for highly textured sequences (natural video content, especially for the Depart sequence), while the reduction of MVD changes is far less severe for computer generated content, such as the Elephant sequence. For highly textured sequences slight spatial shifts (the result of a MVD change) lead to higher MSE distortions.

In the next section we will present evidence that even the highest embedding strength offers very good / excellent quality.

A. Quality Evaluation

A subjective experiment was conducted, 42 observers were enrolled, their acuity was checked as well as normal color vision. The ACR (absolute content rating) protocol was used. For this protocol, a single video sequence is displayed in the center of the screen, and the observer is asked for a quality score after every displayed sequence. The resolution of the tested video sequences was 1280×720. The quality score was (5: Excellent, 4: Good, 3: Fair, 2: Poor, 1: Bad), MOS were computed across the 42 observers, as well as standard deviations for every content across observers, the

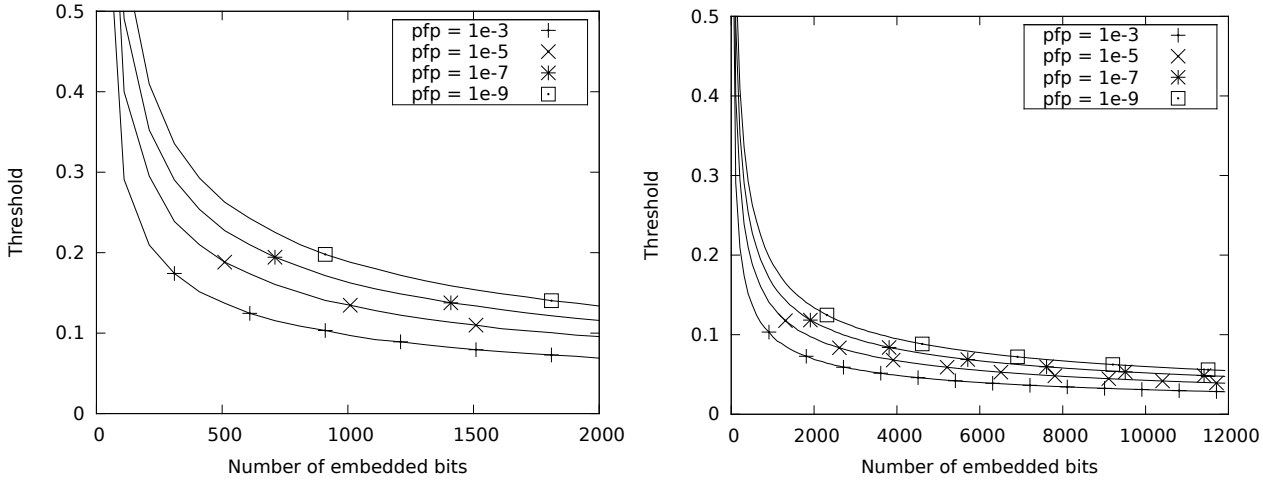


Fig. 4. The threshold for various probabilities of false alarm as a function of the number of embedded bits

maximum standard deviation was below 1 (0.93). During the experiment, fifty six videos were evaluated, the database was build as follows. Among the four input sequences three were collected from the VQEG datasets (Canal, Depart and Ebu) and one sequence was an artificial (cartoon) sequence (Elephant). Every sequence was either watermarked and re-encoded, or only reencoded. The watermarking technique was presented in section IV. Six encoding parameters were used (Qp = 24, 28, 32, 36, 40, 44) for both watermarking and re-encoding scenarios. The original input sequences and watermarked sequences were also considered in the experiment (using a QP of 13). Overall 4 original sequences have been subjected to two distortions (watermarking and coding or coding only) and each resulting sequence has been coded with 7 quantization parameters. In summary this results in a dataset of 56 sequences. Each sequence was 10 seconds long (250 frames). For each observer, the experiment duration was in between 15 and 20 minutes.

The main objective of splitting the dataset into two parts: watermarking and coding was to analyse any perceptual quality loss due to the watermark embedding. To this end, figure 5 shows a histogram representing the difference between watermarked and coded sequences. As we can notice on this figure, the histogram bins are uniformly distributed around zero. Positive x-axis values means that the watermarked sequences presented a higher quality score than the coded version (and negative values along the x-axis means the coded sequence had a higher quality score). The y-axis simply counts the number of occurrences for all observers and for all sequences.

Figure 6 shows the Mean Opinion Score as a function of the quantization parameter for all tested sequences. The gray lines represent the differences between marked and coded sequences. It is interesting to notice that these differences are centered around zero, which means that depending on the sequences and the Qp, the allocated quality score could either be higher for the watermarked sequence or for the coded sequence. On this figure, the symbols (arbitrarily positioned at QP=22) represents the original marked (squares) and coded (diamonds) sequences.

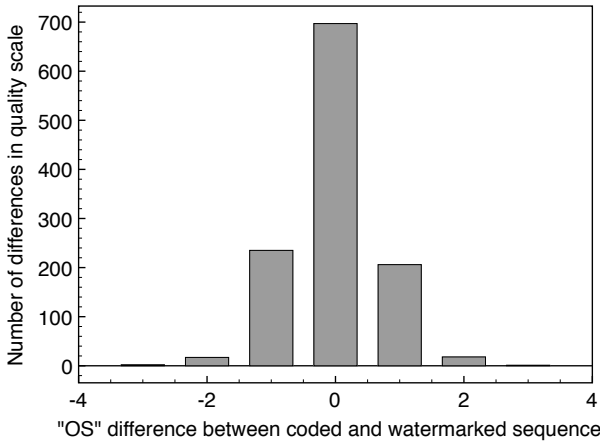


Fig. 5. Watermarked versus coded: differences in opinion scores

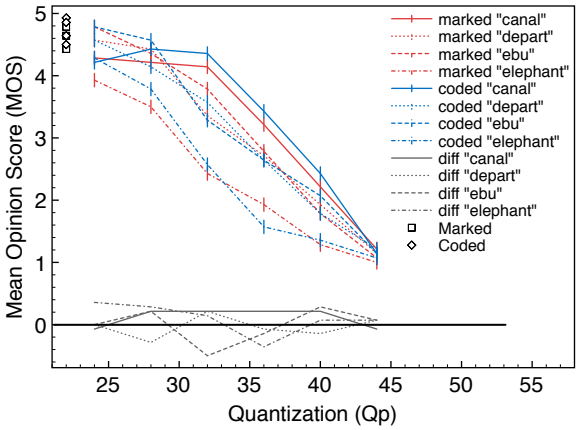


Fig. 6. Subjective mean opinion scores for three various coding parameters (Qp)

	wRMSE	RMSE	RankCorr	OR	Kappa	LinCorr
PSNR	13.6665	0.9419	0.6792	0.7321	0.1754	0.6679
SSIM	12.2094	0.8328	0.7695	0.6071	0.3345	0.7544
CPA1	7.6107	0.7189	0.8090	0.4643	0.4393	0.8229
CPA2	7.7724	0.6481	0.8666	0.4464	0.5186	0.8589
VIF	1.1941	0.2422	0.9621	0.1607	0.8146	0.9815

TABLE IV
METRICS PERFORMANCES

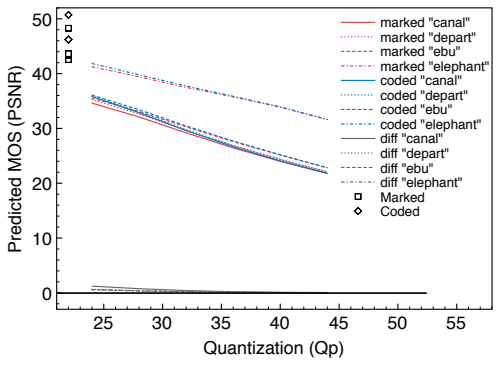
Moreover, five Objective Quality Metrics were tested on this subjective dataset (PSNR, SSIM[21], CPA1[4], CPA2[17] and VIF[19]). The performances of all 5 metrics were assessed in terms of wRMSE, RMSE, Rank correlation, Outlier Ratio, Kappa coefficient, and Linear correlation. Table IV provides the metrics performances for all 5 metrics. It is obvious from table IV that the VIF metric outperforms all others for the six tested performances tools, and PSNR exhibits the worst overall performances. Thus, in the following our analysis will focus on these two metrics.

Figures 7(a) and 7(b) respectively show the MOSp for PSNR and VIF as a function of Qp. We can observe that neither PSNR, nor VIF can notably differentiate coded sequences and watermarked ones. It is interesting to notice that both metrics disagree concerning the assessment of the Elephant sequence (computer generated sequence). A further analysis showed that for all tested metrics, except VIF, the predicted scores for the Elephant sequence were seen as presenting a significantly higher perceptual quality than the other 3 sequences. This explains the overall low metrics performances shown in table IV. This particular behavior is clearly visible on figure 8(a) representing the PSNR plotted as a function of the MOS. For low to good quality, the PSNR values for the Elephant sequence are about 9dB higher than the remaining sequences, whereas the MOS for this sequence was slightly below others (figure 6). Such a behavior is not apparent for the VIF metric (figure 7(b)), which, presents a linear distribution of MOS versus MOSp, and as we have seen above, is capable of discriminating the Elephant sequence as having an overall lower quality (figure 7(b)).

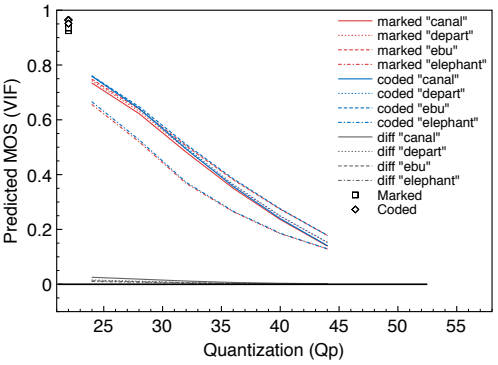
Given our subjective results we conclude that watermarked content can not be visually distinguished from unwatermarked content (when played as video). This may come surprising as the allowed embedding distortion is high with a MbDist of 100 in terms of MSE. However, the watermarking approach implicitly takes advantage of temporal masking effects, as due to the algorithm design the watermark is always embedded in high motion areas, in which distortion are perceived less pronounced by human observers.

B. Robustness Evaluation

The robustness of our watermarking algorithm highly depends on the embedding strength as defined by the parameter MbDist as the number of embedded bits is primarily determined by this parameter. Recompression is the main focus of our robustness evaluation, most importantly recompression with H.263 and H.264. The employed software for

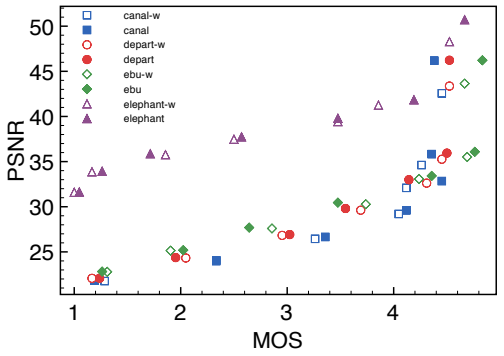


(a) PSNR

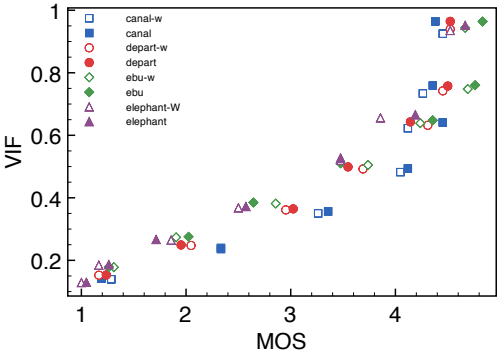


(b) VIF

Fig. 7. OQM predictions on coded or watermarked sequences for various Qp



(a) PSNR



(b) VIF

Fig. 8. OQMs plotted as a function of the mean opinion score

recompression was ffmpeg[1] (vcodec=mpeg4, varying quality scale Qs)for H.263 compression and x264[2] (ultrafast, varying quality parameter Qp) for H.264 compression. The chosen quality ranges correspond to qualities from excellent to bad for both H.264 and H.263.

Additionally results for standard attacks against watermarking algorithms are presented.

1) H.264 and H.263 Compression: Figure 9 plots the detector response against the Qp employed in x264 compression. Results are given for 4 different sequences and different embedding strengths (MbDist). A detection threshold has to be chosen such that it separates the detector responses of un-watermarked content (dashed blue lines in fig. 9) and watermarked content (solid red lines in fig. 9). For MbDist 100 and 25 the selection of detection threshold that separates un-watermarked from watermarked content is obviously possible, and even for a MbDist of 4 the detector response for an un-watermarked work is always below the detector response of the same watermarked work. While this figure can only give a first impression, a better interpretation of the detector values is obtained if we consider the associated probability of false alarm for the obtained detector response. Figure 10 plots the exponent of 1/pfp in basis 10, i.e., a value of 8 corresponds to a probability of false alarm of $p_{fp} = 1/10^8$, against the Qp. Note that the negative exponents of the pfp have been clipped at 15. We notice high robustness to H.264 compression even for very bad quality and even for low embedding strengths, only at MbDist=4 the detection performance decreases significantly (although even this limited robustness may be sufficient for the protection of high quality content). We further have to point out that the ultrafast settings of the x264 encoder can be considered the worst case, as these settings introduce heavy distortions (but are fast). Thus the presented results correspond to a worst case scenario for robustness, other encoders (or other settings for x264) preserve a better quality and thus the watermark is more reliably detected.

Even better is the robustness to H.263 compression as summarized in similar figures (see fig. 11 and 12).

However, single detector responses only represent a single sample from a random experiment; a more thorough analysis has to draw several samples from the random experiment in order to enable a statistical analysis of the underlying distributions. Given that the distortions are computationally expensive (repeated H.264 and H.263 encoding / decoding) the more extensive analysis focuses on medium quality recompression which can be considered the default case in many application scenarios (e.g., illegal file sharing). In the following experiments 50 watermarking keys have been employed both for detection in content watermarked with the same key ($\mathcal{H}_1$) and content that has not been watermarked with the same key ($\mathcal{H}_0$). The experiments have been conducted for different embedding strengths (MbDist). The figures 13 and 14 contain histograms of the detector response under $\mathcal{H}_0$ (on the left, detector responses distributed around 0) and $\mathcal{H}_1$ (on the right). The resulting distribution under $\mathcal{H}_0$ follows approximately a Gaussian distribution, the parameters can be either estimated on the basis of the obtained detector responses (the dashed line in the histograms) or on the basis of our analytical model of sect. V (the solid line with upwards triangles). We notice that the prediction on the basis of our analytical model is very close to the fitted Gaussian distribution. The solid line with downwards triangles is the fitted Gaussian distribution for the detector responses of watermarked content. If the embedding strength is reduced the detection performance slowly decreases, i.e., the distributions of Z under $\mathcal{H}_0$ and Z under $\mathcal{H}_1$ are less and less separated. However, even for an embedding strength of 25 the distributions are clearly separated, and for a embedding strength of 4 many sequences still present well separable distributions. The same information (as contained in histograms) can be plotted in ROC (receiver operation curves). The results are only shown for the Depart and the Elephant sequence as the results of these two are the most different, the performance of the other sequences is in between. The different behaviour of these two sequences is due to the different video characteristics, on the one hand the relatively smooth computer-generated Elephant sequence on the other hand the highly textured Depart sequence, with high, but local and independent motion (it contains a sequence of a cross country running race, each runner moves independently of the others). For the Depart sequence there are simply too few MVD changes that result in a distortion below an MSE of 4. On the other hand the MSE penalty of an MVD change in a smooth sequence, such as Elephant is far less pronounced, resulting in many watermarkable blocks and thus a higher robustness.

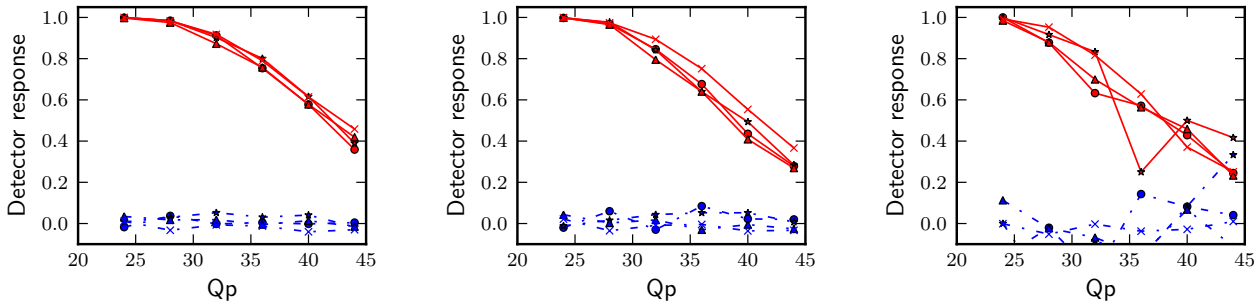
In ROC plots the false positive probability (x-axis) is plotted against the exponent of the false negative probability in base 10 (y-axis). The closer an ROC curve is to the x-axis and the y-axis the better is the performance of the associated watermarking scheme (the proposed scheme with different parameters of MbDist). We plotted results starting from a very high false positive probability of -1 to a a very low false positive probability of -25, which should contain results for most practical systems. The computation of the ROC curves employ approximation with Gaussian distributions (derived parameters from the analytical model for $\mathcal{H}_0$ and estimated parameters for $\mathcal{H}_1$). The schemes with MbDist=100 and MbDist=25 perform excellent for all sequences and both distortions (see figures 15 and 16). Only for a MbDist=4 problems are encountered for sequences with a low embedding capacity. The embedding capacity depends on the source characteristics, MVD changes in highly textured content result in an higher MSE. In a practical system, one would simply need to embed the watermark into more frames, i.e., a longer sequence.

In conclusion, at a MbDist of 25 and above we can extremely reliably detect the presence of the watermark even for extremely compressed sequence (both H.263 and H.264) and even at lower embedding strength many sequences show a very good detection performance.

It is also notable, that the detector responses do not change significantly for the different embedding distortions, only the number of embedded bits increases significantly with higher embedding distortions. The higher the number of embedded

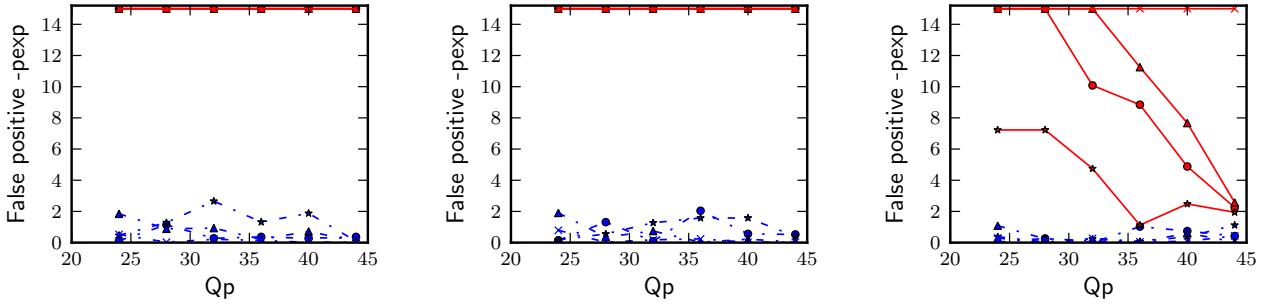
[1]FFmpeg version SVN-r0.5.1-4:0.5.1-1ubuntu1.2, Copyright (c) 2000-2009 Fabrice Bellard, et al

[2]x264 0.85.1448 Ubuntu_2:0.85.1448+git1a6d32-4



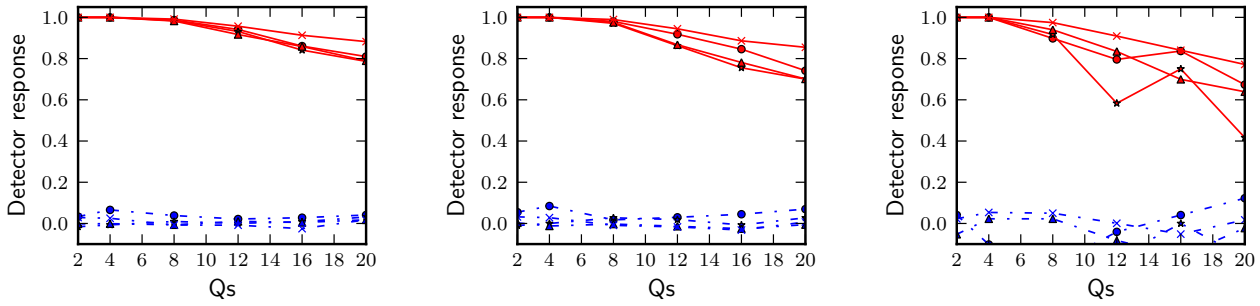
(a) MbDist=100 (b) MbDist=25 (c) MbDist=4

Fig. 9. Detector response for varying Qp of x264 (720p, 250 frames)



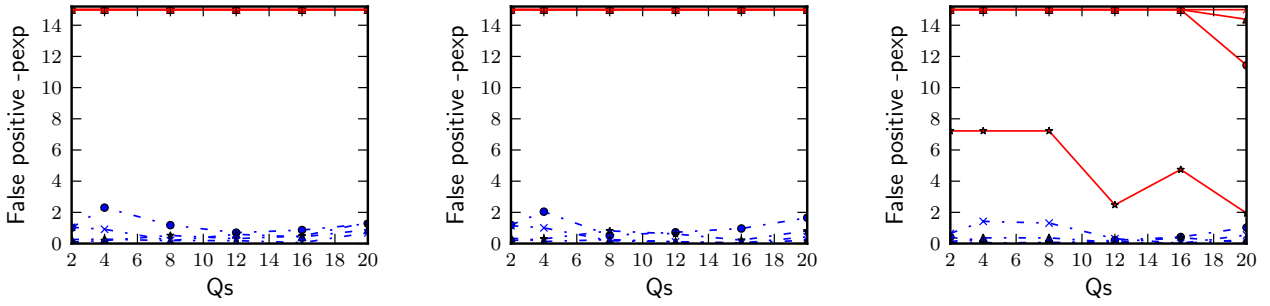
(a) MbDist=100 (b) MbDist=25 (c) MbDist=4

Fig. 10. Probability of a false positive for varying Qp of x264 (720p, 250 frames)



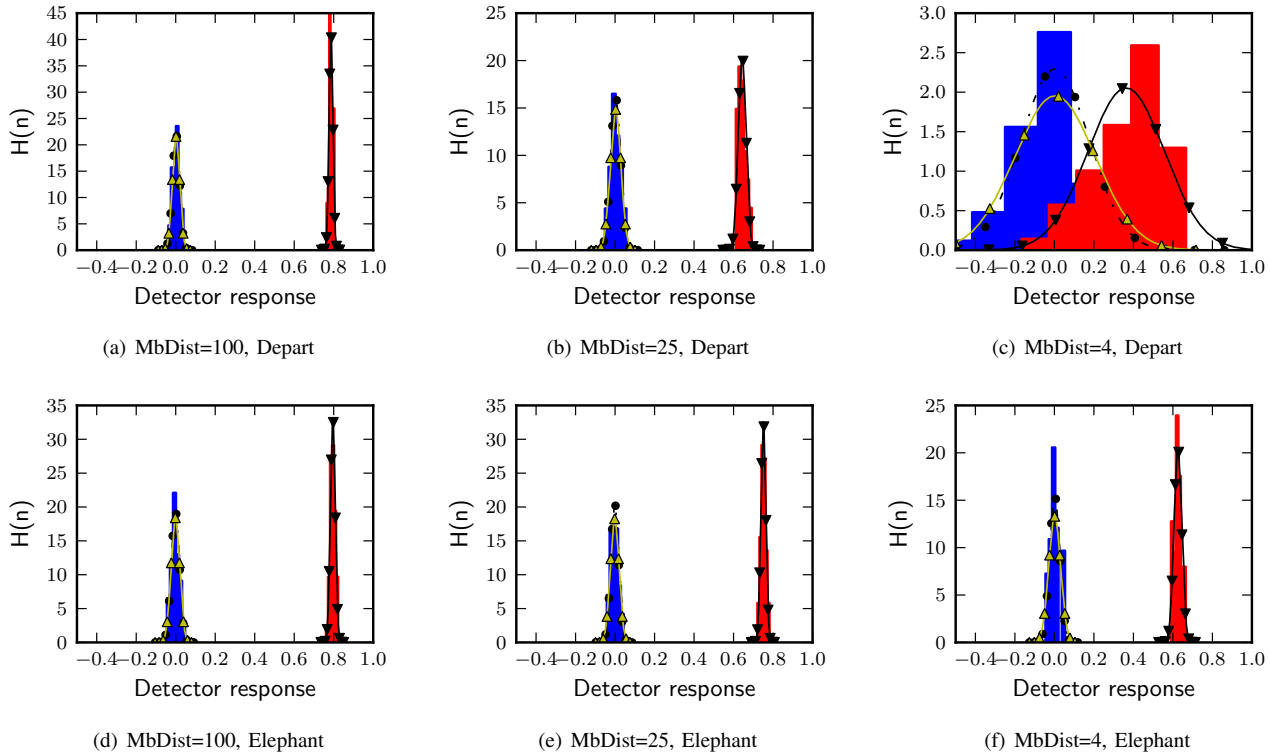
(a) MbDist=100 (b) MbDist=25 (c) MbDist=4

Fig. 11. Detector response for varying Qs for H.263 (ffmpeg, mpeg4, 720p, 250 frames)



(a) MbDist=100 (b) MbDist=25 (c) MbDist=4

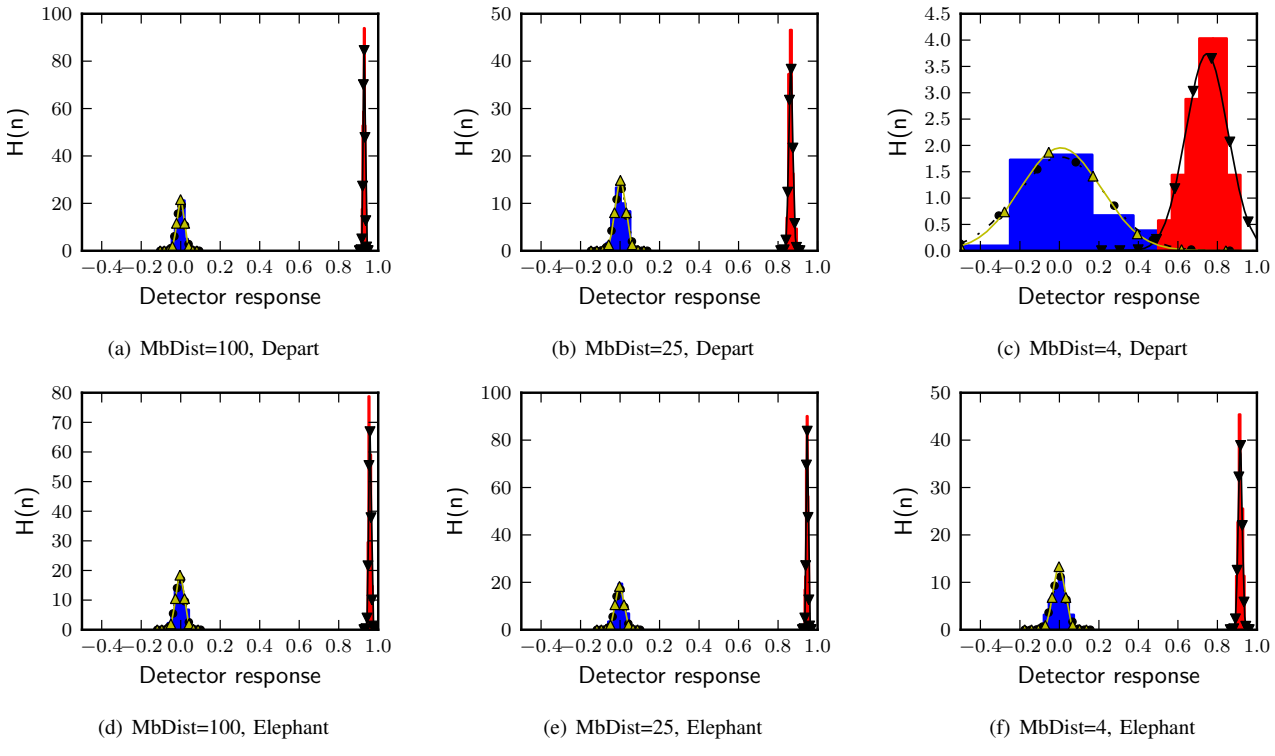
Fig. 12. Probability of a false positive for varying Qs of H.263 (ffmpeg, mpeg4, 720p, 250 frames)



(a) MbDist=100, Depart (b) MbDist=25, Depart (c) MbDist=4, Depart

(d) MbDist=100, Elephant (e) MbDist=25, Elephant (f) MbDist=4, Elephant

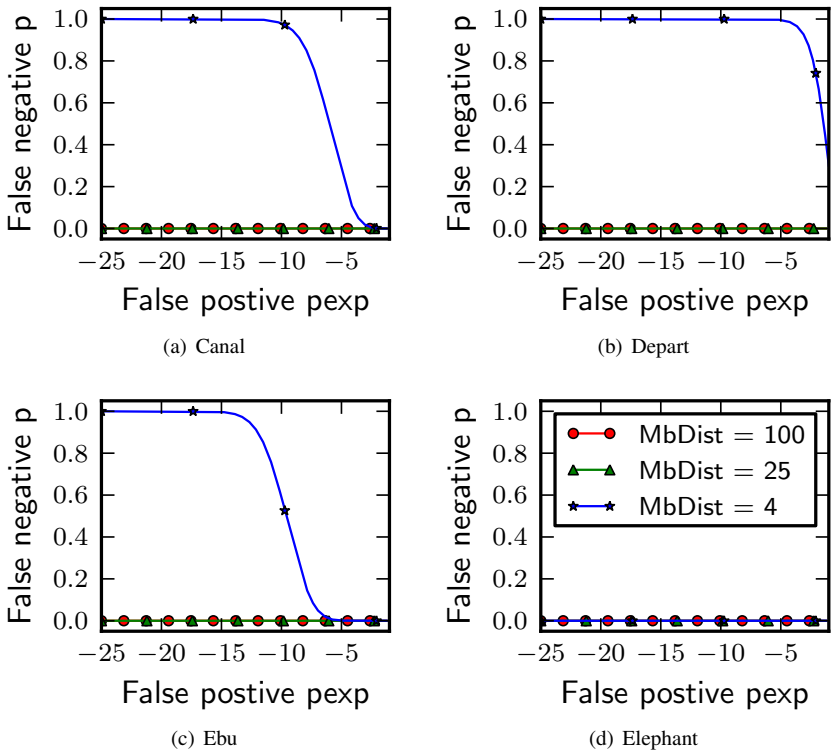
Fig. 13. H.264 (x264, ultrafast, Qp 36): Histogram of 100 detector responses



(a) MbDist=100, Depart (b) MbDist=25, Depart (c) MbDist=4, Depart

(d) MbDist=100, Elephant (e) MbDist=25, Elephant (f) MbDist=4, Elephant

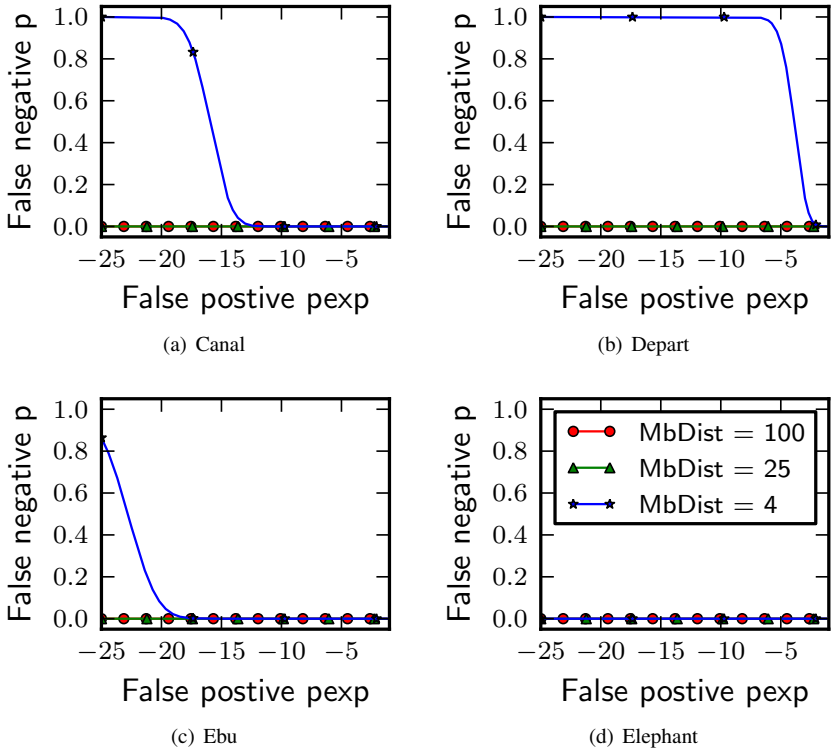
Fig. 14. H.263 (ffmpeg, mpeg4, Qs=12): Histogram of 100 detector responses



(a) Canal (b) Depart

(c) Ebu (d) Elephant

Fig. 15. ROC at H.264 compression (x264, ultrafast, Qp 36)



(a) Canal (b) Depart

(c) Ebu (d) Elephant

Fig. 16. ROC at MPEG-4 compression (ffmpeg, mpeg4, qs 12)

Dist. / Seq.	Canal	Depart	Ebu	Elephant
gaussian1	0.0766	0.095	0.160	0.125
gaussian2	0.0766	0.095	0.159	0.124
medfilt1	0.0961	0.093	0.150	0.117
medfilt2	0.0792	0.101	0.161	0.124
medfilt3	0.1064	0.100	0.155	0.112
sharpen1	0.0870	0.072	0.159	0.144
wiener1	0.0792	0.093	0.158	0.125
wiener2	0.0688	0.092	0.148	0.123
dpr1	0.0805	0.092	0.152	0.119
dpr2	0.0844	0.104	0.146	0.109

TABLE V
CHECKMARK ATTACKS: DETECTOR RESPONSES

Dist. / Seq.	Canal	Depart	Ebu	Elephant
gaussian1	0.00120	1.08-7	5.61e-25	2.34e-9
gaussian2	0.00120	1.08-7	1.07e-24	3.04e-9
medfilt1	0.00007	1.72-8	3.15e-22	2.91e-8
medfilt2	0.00101	2.12-8	2.91e-25	3.04e-9
medfilt3	0.00001	2.12-8	1.01e-23	9.56e-8
sharpen1	0.00028	4.27-5	1.07e-24	8.77e-12
wiener1	0.00101	1.95-7	1.48e-24	2.34e-9
wiener2	0.00371	2.36-7	1.06e-21	6.57e-9
dpr1	0.00071	2.36-7	1.25e-22	1.39e-8
dpr2	0.00050	5.92-9	2.63e-21	1.51e-7

TABLE VI
CHECKMARK ATTACKS: PROBABILITY OF FALSE ALARM

bits the lower the detection threshold can be chosen for a given probability of false alarm. Thus lower embedding distortions solely require to watermark more frames to achieve higher robustness.

2) Watermarking Attacks: In the following we present results for a relevant subset of the Checkmark watermarking evaluation framework [18]. As the embedding distortion mainly influences the number of embedded bits, only results for an embedding distortion of 100 (MbDist) are given in tables V and VI. Although the attacks significantly reduce the detector responses, the decrease is not a problem as long as the the number of embedded bits is sufficient. The number of embedded bits can be simply increased by watermarking more frames (our results are only for 10 seconds video clips).

VII. PREVIOUS WORK

There has been a tremendous interest[3] in H.264 watermarking. However, all but one of the previously presented approaches for H.264 watermarking are not applicable in the stringent application requirements of structure-preserving H.264 CAVLC watermarking. Many schemes exploit the H.264 encoding process to embed the watermark during compression. The main advantage of such approaches [16] is that the error introduced by watermarking is not propagated further (at the cost of some bitrate increase). Other schemes work on the bitstream, mostly to reduce the computational burden of compression-integrated watermarking schemes. It has to be noted that H.264 bitstream watermarking actually performs entropy-decoding, such that the syntax elements can be accessed, watermarked (e.g., the quantized DCT coefficients),

[3]ACM digital library reports 180 publication on H.264 watermarking. IEEE Xplore reports 72 publications on H.264 watermarking.

Zou & Bloom		Our approach			
CALVC	CABAC	Canal	Depart	Ebu	Elephant
0.625	1.319	17.672	41.196	57.420	18.820

TABLE VII
EMBEDDINGS PER FRAME OF 1080P VIDEO, THE RESULTS OF ZOU'S CAVLC [23] AND ZOU'S CABAC [22] COMPARED TO OUR APPROACH FOR DIFFERENT SEQUENCES

and again entropy-encoded. The approaches presented in [15] are examples for H.264 bitstream watermarking. Most related to our application requirements is the setup in the work of Zou and Bloom [24], [23], that discusses substitution watermarking for intra frames of H.264 CAVLC bitstreams, but is not capable to watermark inter coded frames (the vast majority of frames is commonly coded as inter frames, some encoders use intra frames only once at the start of a sequence). Thus methods for substitution watermarking of inter frames are needed. The approach of Zou and Bloom [23] modifies the intra-prediction modes which can be implemented by bit-substitutions of H.264 CAVLC bitstreams. Suitable substitutions have to be found in a complex analysis stage, which has to consider intra and inter drift, while our algorithm has a lightweight analysis stage. Furthermore the capacity of our approach is superior compared to their results (see table VII) and thus a lower threshold can be selected for the same probability of alarm, which increases the overall robustness of the watermarking scheme. A substitution watermarking algorithm for CABAC, based on MVD changes, was presented by the same authors in [22]. However, CABAC and CAVLC are entirely different, and thus the applicable changes are different. MVDs are encoded context-adaptively in CABAC, and thus a computationally complex analysis stage is required in the approach of [22]. Most importantly the number of feasible changes is smaller by an order of magnitude and thus the CAVLC algorithm performs better in terms of capacity, which also leads to an increased robustness.

The comparison of the robustness on bit embedding level reveals that the performance is rather similar, while our robustness criterion (the average luminance feature difference must be larger than 0.25) leads to higher correlations for downsizing, the effect of downsizing and compression are almost the same (see table VIII). A special case again is the computer generated Elephant sequence, which can be compressed very efficiently and thus the highest correlations are observed with this sequence.

The number of embedded bits has a tremendous effect on the threshold selection (for a fixed probability of false alarm) or the probability of false alarm (for a fixed threshold). Figure 17 plots the thresholds for different probabilities of false alarms and for the different approaches against the number of frames. As distortions affect the different approaches almost similarly this allows a fair comparison of the performance of the approaches. The lower the threshold the more robust is the watermarking algorithm against distortions. Our algorithm requires significantly lower detection thresholds compared to the algorithms of Zou [23], [22].

Attack	Zou CAVLC (Full HD)	Our (Canal Full HD)	Our (Depart)	Our (Ebu)	Our (Elephant)
No Attack	0.9685	1.0000	1.0000	1.0000	1.0000
Downsize to 960x540	0.8133	0.9995	0.9998	0.9988	1.0000
Downsize to 480x270	0.2778	0.9488	0.9586	0.9669	0.9974
Downsize to CIF	0.1804	0.8497	0.8572	0.8942	0.9795
Downsize to CIF, 1M	0.1148	0.2643	0.2403	0.2108	0.8826
Downsize to CIF, 780K	0.1181	0.2489	0.2088	0.1710	0.8826
Downsize to CIF, 300K	0.1026	0.0878	0.1043	0.0882	0.6272

TABLE VIII
AVERAGE CORRELATION FOR ZOU'S APPROACH [23] AND OUR ALGORITHM

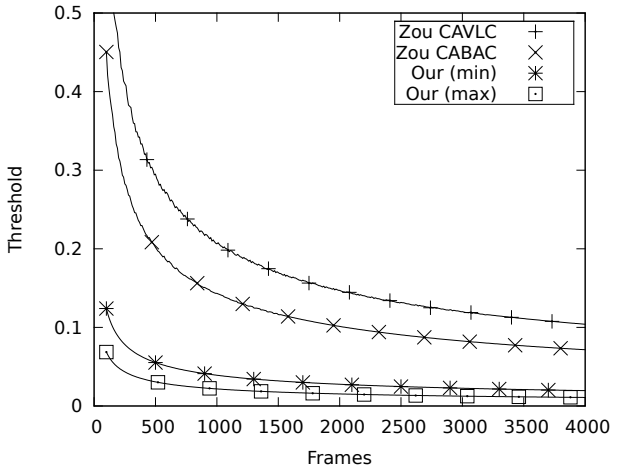


Fig. 17. The thresholds for the approaches of Zou and to our algorithm plotted as a function of frames for a probability of false alarm of 10^{-7}

VIII. CONCLUSION

The proposed H.264/CAVLC watermarking algorithm enables structure-preserving H.264 watermarking. Due to the separation of embedding in an analysis and and a substitution stage, it is able to efficiently embed numerous different watermarks in the same content. Compared to previous work it offers a significantly increased capacity and robustness. While the analysis stage of the algorithm is lightweight compared to previous proposals, it still satisfies the invisibility constraints, which has been shown by subjective experiments. The algorithm offers high robustness to re-compression and sufficient robustness against standard watermarking attacks.

REFERENCES

[1] S. H. Baker and M. E. Carpenter. Correlation of spot characteristics with perceived image quality. *IEEE Trans. Commun. Electron.*, 35:319–324, 1989.

[2] Peter G. J. Barten. Evaluation of subjective image quality with the square-root integral method. *J. Opt. Soc. Am. A*, 7:2024–2031, 1990.

[3] M. Carnec, P. Le Callet, and D. Barba. Objective quality assessment of color images based on a generic perceptual reduced reference. *Signal Processing: Image Communication*, 23(4):239–256, 2008.

[4] Maurizio Carosi, Vinod Pankajakshan, and Florent Autrusseau. Toward a simplified perceptual quality metric for watermarking applications. In *Proceedings of the SPIE conference on Electronic Imaging*, volume 7542, 2010.

[5] D. M. Chandler and S. S. Hemami. Vsnr: A wavelet-based visual signal-to-noise ratio for natural images. *IEEE Transactions on Image Processing*, 16, 2007.

[6] T. Y. Chung, M. S. Hong, Y. N. Oh, D. H. Shin, and S. H. Park. Digital watermarking for copyright protection of MPEG2 compressed video. *IEEE Transactions on Consumer Electronics*, 44(3):895–901, 1998.

[7] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2007.

[8] ITU P.910. Subjective video quality assessment methods for multimedia applications. Technical report, Intl Telecom. Union, April 2008. SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS Audiovisual quality in multimedia services.

[9] ITU-R-BT.500-11. Methodology for the subjective assessment of the quality of television pictures question itu-r 211/11, g. Technical report, Intl Telecom. Union, 2004.

[10] ITU-T H.264. Advanced video coding for generic audiovisual services, November 2007.

[11] L. C. Jesty. The relation between picture size, viewing distance and picture quality. *Proc. Inst. Electr. Eng. Part B*, 105:425–439, 1958.

[12] Gerrit C. Langelaar and Reginald L. Lagendijk. Optimal differential energy watermarking of dct encoded images and video. *IEEE Transactions on Image Processing*, 10(1):148–158, January 2001.

[13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[14] Bijan G. Mobasseri. Watermarking of MPEG-2 video in compressed domain using VLC mapping. In *International Multimedia Conference, Proceedings of the 7th Workshop on Multimedia and Security, MM-SEC '05*, pages 91–94, New York, NY, USA, August 2005. ACM.

[15] M. Noorkami and R. M. Mersereau. Compressed-domain video watermarking for H.264. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '05*, pages 890–893, Genova, Italy, September 2005. IEEE.

[16] M. Noorkami and R. M. Mersereau. A framework for robust watermarking of H.264 encoded video with controllable detection performance. *IEEE Transactions on Information Forensics and Security*, 2(1):14–23, March 2007.

[17] V. Pankajakshan and F. Autrusseau. A multi-purpose objective quality metric for image watermarking. In *IEEE International Conference on Image Processing, ICIP'2010*, pages 2589–2592, 2010.

[18] Shelby Pereira, Sviatoslav Voloshynovskiy, M. Madueno, and Thierry Pun. Second generation benchmarking and application oriented evaluation. In *Proceedings of the 4th Information Hiding Workshop '01*, volume 2137 of *Lecture Notes in Computer Science*, pages 340–353, Portland, OR, USA, April 2001. Springer.

[19] H. R. Sheikh and A. C. Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, May 2006.

[20] VQEG MM. Final report from the video quality experts group on the validation of objective models of multimedia quality assessment, phase I. Technical report, VQEG, 2008.

[21] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[22] D. Zou and J. Bloom. H.264 stream replacement watermarking with CABAC encoding. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '10*, Singapore, July 2010.

[23] Dekun Zou and Jeffrey Bloom. H.264/AVC substitution watermarking: a CAVLC example. In *Proceedings of the SPIE, Media Forensics and Security*, volume 7254, Jan Jose, CA, USA, January 2009. SPIE.

[24] Dekun Zou and Jeffrey A. Bloom. H.264/AVC stream replacement technique for video watermarking. In *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '08*, pages 1749–1752, Las Vegas, NV, USA, March 2008. IEEE.