Implementation-Free Forensic Watermarking for Adaptive Streaming with A/B Watermarking

Hannes Mareen, Glenn Van Wallendael, and Peter Lambert

Ghent University – imec, IDLab, Department of Electronics and Information Systems
Ghent, Belgium
{hannes.mareen, glenn.vanwallendael, peter.lambert}@ugent.be
http://media.idlab.ugent.be/

Abstract. Forensic watermarking enables the identification of digital pirates after they illegally re-distribute copyright-protected videos. For adaptive streaming, these methods are best combined with A/B watermarking, in which two watermarked versions are created for each video segment, and subsequently mixed in order to create a large number of uniquely-watermarked videos. Although good video quality and low bit-rate are key characteristics of a good watermarking system, existing methods objectively lower the compression efficiency. Additionally, they often require complex implementations. Therefore, this paper proposes an implementation-free, rate-distortion-preserving watermarking technique to be used with the scalable A/B watermarking concept. Even though the embedding is performed during compression, it does not change the existing video encoder implementation. Instead, it only changes the target bit-rate parameter in order to create different compression artifacts. These artifacts represent the watermark, but are not noticeable due to high-quality coding. As such, the rate-distortion performance is nearly equal to that of ordinary, unwatermarked compression (i.e., a BD-rate of 0.02% and −0.10% when applied with a H.264/AVC and H.265/HEVC encoder, respectively). Furthermore, the robustness is equal or better than state-of-the-art methods with comparable embedding complexities. More specifically, in case of recompression attacks, nonzero false-negative rates are only reported when a watermarked video is initially compressed with a high quality and degraded to a very low quality. Consequently, the proposed scheme can be used in practice by adaptive-streaming platforms without a quality decrease, bit-rate increase, or implementation overhead.

Keywords: Forensic watermarking, compression efficiency, imperceptibility, rate-distortion performance, implementation free.

1 Introduction

Digital piracy is a common problem for copyright holders that distribute their content on video-on-demand platforms. In order to prevent direct access to the streamed media by clients, those platforms often apply encryption-based digital

rights management (DRM) security measures. However, these tools require extra resources on the clients' devices and do not fully prevent content from being pirated after its decryption [1]. Therefore, forensic watermarking methods are used to enable the identification of digital pirates after illegal video redistribution, i.e., to perform traitor tracing or active fingerprinting. That is, every user receives a uniquely-watermarked version of the protected video, i.e., the watermarked video is linked to the receiver's identifier (ID). When a malicious user leaks their version of the video, the watermark associated to the culprit can be detected.

Video-on-demand applications typically serve a large number of users. In order to distribute videos to many users with varying internet and device conditions, adaptive-streaming protocols are typically used. More specifically, client devices dynamically react to varying conditions by streaming content representations with other bit-rates and resolutions. In typical adaptive-streaming applications, the streamed videos are encoded once and then distributed to a large number of users. Hence, the visual quality and bit-rate is of utmost importance. For example, popular streaming platforms were downgrading the video bit-rates by 25% to avoid network congestion in March 2020, when Europe went into lockdown due to the coronavirus outbreak [2]. Moreover, because the users may stream videos from a large variety of devices and under different conditions, the videos are accessible in a variety of content representations, potentially compressed with different video encoders. For example, old devices can be limited to the older, low-complexity H.264/Advanced Video Coding (AVC) standard, whereas modern devices can use the newer, more-efficient H.265/High Efficiency Video Coding (HEVC) standard.

Watermarking in adaptive-streaming applications is not straightforward because the server must provide each user with a uniquely-watermarked video. This is not practical when the number of users is large because the watermarking algorithm is too complex for real-time embedding, or because it requires an unpractical increase in storage space. Moreover, it disrupts file caching mechanisms in the content delivery network (CDN). A solution is to distribute the same video to all users and perform watermark embedding at the client device [3, 4]. However, this requires extra DRM software to be installed on the client devices, as well as extra client-side resources to embed the watermark in real-time during video decoding. Additionally, it decreases the security of the system since malicious users may get access to the unwatermarked video that is temporarily stored on their device. As an alternative, the A/B watermarking or two-step watermarking framework is often applied, in which only a small number of watermarked versions are created for every short video segment, using an existing watermarking method [1]. In other words, every segment is not only available in various content representations (e.g. bit-rates, resolutions), but also in different watermarked versions. For example, only two watermarked variants (i.e., version A and version B) can be created for each segment. Before sending the video to a user, the watermarked segments are mixed to create a unique switching pattern that identifies the user. This example is illustrated in Fig. 1.

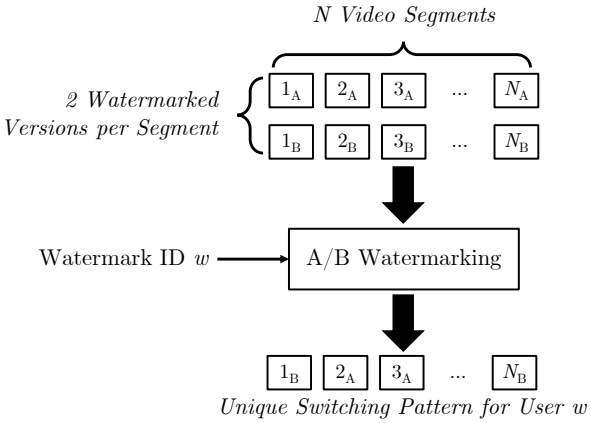


Fig. 1. Example of A/B Watermarking: only two watermarked versions are created for each of the N video segments. By mixing these segments according to a watermark ID w, every user receives a unique switching pattern as a watermarked video.

As such, A/B watermarking is routinely used in adaptive bit-rate applications where the number of users is high [5].

In the last two decades, many watermarking techniques have been created that can be used in combination with the A/B watermarking concept [6, 7]. An important requirement of these techniques is that they are imperceptible while not causing a significant bit-rate overhead. In other words, the compression efficiency or rate-distortion (RD) performance may not be influenced significantly. Moreover, the watermarks should be robust to manipulations that try to destroy it. Since a more perceptible watermark is usually harder to delete, there is a trade-off between imperceptibility and robustness. For example, video encoders are made to remove imperceptible information in order to achieve stronger compression. Thus, video compression subsequent to watermarking is already an (unintentional) attack to the security system.

In order to balance the two requirements, watermarks are often embedded in a transform domain because they have interesting properties regarding perceptibility and robustness. For example, the discrete cosine transform (DCT) and wavelet transform domains are commonly used [8–14]. That is because the human visual system is more sensitive to low-frequency DCT coefficients changes, such that a watermark embedded in those coefficients is perceptible, but hard to remove without decreasing the visual quality even more. In other words, the trade-off between robustness and imperceptibility is controlled by changing either low- or high-frequency coefficients. In order to further balance imperceptibility and robustness, visual masking is commonly applied during embedding [11,15,16]. In this technique, the watermark is masked such that it is more strongly embedded in regions with low noise sensitivity than in those where it is more perceptible. Most importantly, although existing techniques are relatively imperceptible,

they still objectively decrease the visual quality and/or increase the bit-rate of the compressed video.

In an effort to minimize both the bit-rate overhead and quality decrease, our previous work proposed a forensic watermarking method that does not affect the RD performance [17]. That is, the watermarking step is incorporated in the video compression process by varying the quantization parameters (QPs) of coding transform units (CTUs) during encoding. Because the rest of the video encoder is left unchanged, the compression efficiency is preserved. Although this effectively minimizes the RD overhead, it requires the video encoder implementation to be adapted, which is not always possible or practical (e.g. due to proprietary software). Additionally, it is expensive when multiple video codecs are used, which is often the case in adaptive-streaming applications.

As a solution, this paper proposes a forensic watermarking method that preserves the RD performance and that does not change the existing video encoder implementation. Instead, only the input parameters of the video encoder are changed in order to create a watermarked video. More specifically, the target bit-rate is changed. Because of the flexibility of modern video encoders and their rate-control algorithms, a different target bit-rate results in different coding decisions, which on their turn result in different compression artifacts. Although these artifacts are different, the RD performance of the compressed watermarked videos is quasi equal.

The remainder of this paper is structured as follows. First, Section 2 summarizes background information of video coding standards. Then, the forensic watermarking method is proposed in Section 3. Subsequently, Section 4 discusses the experimental results. Finally, the paper is concluded in Section 5.

2 Video Coding

Video compression standards such as H.264/AVC, H.265/HEVC, and the recently-standardized H.266/Versatile Video Coding (VVC) transform an uncompressed video into a compressed bit-stream. The compressed video consists of two main components: the coding information and the residual signal. The coding information involves the block and prediction structure. For example, a frame is typically partitioned into blocks of pixels of various sizes, and every block is predicted using motion vectors and intra-prediction modes. Usually, this prediction is not perfect. Therefore, the residual signal corrects the prediction errors. The residual signal is transformed and quantized with a configurable quantization parameter (QP), which results in less bits to store, but also introduces compression artifacts. The QP is typically determined automatically by a rate-control algorithm, or varied periodically by a constant rate factor (CRF).

Modern video codecs have a large number of flexible tools and potential coding decisions, which rate-control algorithms use to achieve a target bit-rate [18]. The QP is the most effective coding decision to be varied by rate-control algorithms, as it controls the granularity of transformed coefficients of the residual signal. Moreover, they can also change (the granularity of) other coding deci-

sions such as the block partitioning and motion vectors. In adaptive-streaming applications, this is typically done using a two-pass encoding [19]. In a first pass, the content is analyzed and the result is stored in a log file. Then, in the second pass, these results are used to achieve good encoding quality as well as accurate rate allocation.

3 Proposed Watermarking Method

The proposed forensic watermarking method is created to be used in combination with A/B watermarking. Thus, only few watermarked versions are created for each video segment. First, the embedding algorithm is described in Section 3.1, followed by the detection method in Section 3.2.

3.1 Watermark Embedding

The watermark is embedded during video compression, but the proposed algorithm does not require changes to the video encoder. Instead, it only changes the video encoder's input parameter that sets the target bit-rate. More specifically, in order to embed watermark ID w, the default target bit-rate t is changed to $t + w$ kilobit per second (kbit/s), where w is a signed integer.

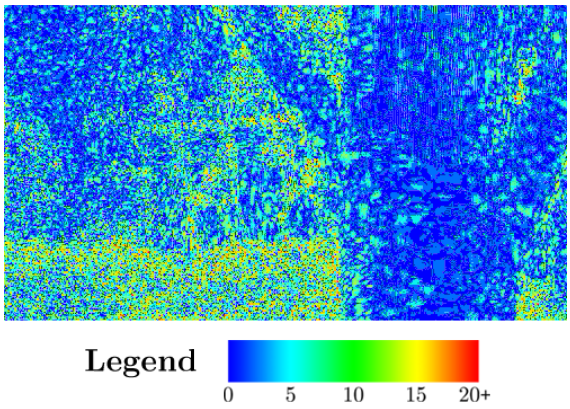
Changing the target bit-rate has an effect on coding decisions made by the video encoder's rate-control algorithm, as explained in Section 2. Changing coding decisions such as the QP introduces different compression artifacts, which are used as a watermark representation. This effect is illustrated in Fig. 2. More specifically, Fig. 2a shows a crop of a compressed frame of the *ParkScene*-sequence, using a two-pass encoding with CRF = 27 and a target bit-rate of $t = 3600$ kbit/s, using the *x265*-encoder. Fig. 2b shows the same crop, watermarked and compressed using the same parameters, except for the target bit-rate which was changed with 1 kbit/s, to $t-1 = 3599$ kbit/s. Although both versions are encoded in a similar way and have a near-equal rate-distortion performance, there are many differences between them, as visualized in Fig. 2c.

In our proposed method, the target bit-rates of each two watermarked videos differ minimally with 1 kbit/s. This value is sufficiently large to cause the creation of a different compression artifacts in typical rate-control algorithms, as validated in the experiments in Section 4.3. Additionally, when the watermark ID is a small number, the difference in target bit-rate is low, such that the output quality and bit-rate are not changed significantly. In any case, the RD performance is preserved because the video compression operates as usual. Moreover, since A/B watermarking requires only few watermarked versions for each segment, the watermark ID always remains small.

The embedding complexity is similar to other uncompressed-domain watermarking algorithms and methods that perform watermarking during video compression. That is, they perform a separate video encoding for each watermarked version of the video. Since the proposed method should be used in combination with A/B watermarking, this high embedding complexity is not a problem, since only few watermarked versions are created for each video segment.

(a) Compressed (3600 kbit/s).

(b) Watermarked (3599 kbit/s).



(c) Visualization of absolute pixel differences.

Fig. 2. Cropped example of (a) a compressed frame, (b) a watermarked frame, and (c) the differences between the compressed and watermarked frame. The crop has a resolution of 500×280 pixels, and a pixel is represented by an 8-bit integer between 0 and 255. Although the watermarked and compressed frame have an approximately-equal quality and bit-rate, they exhibit many differences.

3.2 Watermark Detection

The compression artifacts that represent the watermark are detected by using a correlation-based technique [17,20,21]. More specifically, the observed video $\boldsymbol{o}$ is first compared to all distributed watermarked versions $\boldsymbol{w_i}, i \in \{1, 2, \ldots, N\}$ by calculating the root mean squared error (RMSE). It should be noted that the number of distributed watermarked versions N is very low when used with A/B watermarking. The RMSE is defined in (1), in which p represents the number of pixels of the video.

$$\mathrm{RMSE}(\boldsymbol{o}, \boldsymbol{w_i}) = \frac{1}{p}\sqrt{\sum_{j}^{p}(\boldsymbol{o}_j - \boldsymbol{w_{i,j}})^2} \tag{1}$$

When the observed video does not contain any additional distortions, the RMSE corresponding to one of the watermarked videos is zero, whereas it is nonzero for the other watermarked videos. Nevertheless, when attacks are performed prior to leaking the video, the RMSEs corresponding to all videos are nonzero, due to the distortions introduced by the attack. Because attacks increase the RMSE, it is not possible to perform robust watermark detection by simply comparing the RMSE value to a threshold.

Although a single RMSE value cannot indicate watermark presence, the collection of RMSE values of all watermarked videos can. More specifically, the RMSE value corresponding to the present watermarks is significantly smaller than the values from absent watermarks. For this reason, we convert every RMSE value into a z-score, which represents the number of standard deviations that an RMSE value differs from the mean of the distribution of RMSE values corresponding to absent watermarks. To put it in a simpler way, it performs outlier detection [22,23]. The z-score is defined in (2). In the equation, r_i is the RMSE value corresponding to watermark w_i, and μ_A and σ_A are the mean and standard deviation of the distribution of RMSE values corresponding to absent watermarks, respectively. For simplicity, we assume that all watermarks are absent except for the one corresponding to the lowest RMSE value. Alternatively, the RMSE values corresponding to watermarks that were never distributed can be used as an estimate for all absent watermarks.

$$z(r_i, \mu_\mathrm{A}, \sigma_\mathrm{A}) = \frac{r_i - \mu_\mathrm{A}}{\sigma_\mathrm{A}} \tag{2}$$

The z-score calculation normalizes the RMSE values corresponding to absent watermarks, such that the collection of z-scores corresponding to absent watermarks has a zero mean with unit variance. In contrast, the z-score corresponding to the present watermark has a much lower (negative) value. As such, to detect a watermark's presence, we can compare the corresponding z-score to a threshold. To estimate the threshold T for a certain false positive (FP) probability P_fp, the Gaussian method defined in (3) is used [24,25].

$$T = \sqrt{2}\,\mathrm{erfc}^{-1}(2P_\mathrm{fp}) \tag{3}$$

In the equation, erfc^{-1} is the inverse of the complimentary error function. For example, an FP probability of $P_{\text{fp}} = 10^{-6}$ results in a threshold $T \approx -4.8$. Finally, a watermark's presence is detected when the corresponding z-score does not exceed this threshold.

Note that the detected watermark is a zero-bit watermark. That is to say, it cannot embed or extract a multi-bit payload bit by bit from a single video segment. Instead, we can only detect the watermark's presence or absence. Then, by linking a zero-bit watermark with the user to which the corresponding video was distributed, the leaker is detected. Additionally, a multi-bit payload can optionally be embedded by combining the proposed method with A/B watermarking. In that case, every segment can represent one bit of information, assuming only two watermarked versions per segment are used. In this way, by using multiple segments in the entire video, a multi-bit watermark is embedded.

The proposed method requires all distributed watermarked versions of the video to detect the leaked watermark. Even so, as mentioned before, it should be stressed that there are only few watermarked versions when used in combination with A/B watermarking (e.g., only two). Nonetheless, the method is non-blind. Non-blindness has the advantage that registration techniques can be applied to undo spatial and temporal synchronization attacks, increasing the robustness of the method. Moreover, the complexity of extraction increases linearly with the number of distributed watermarked videos. Again, although the order of the asymptotic complexity may be larger than other existing methods, it should be stressed that there are only few distributed watermarked videos in A/B watermarking. Hence, in that case, the total computational complexity is still very low.

Finally, one may note that when only two watermarked versions of a video segment are created in A/B watermarking, the proposed detection method does not work as is. That is because there is only a single absent watermark, resulting in a the standard deviation σ_{A} of zero, i.e., a z-score of (negative) infinity. As a solution, few non-distributed watermarked videos can be created as extra watermarked videos for accurate detection. These limitations are the cost of an implementation-free, rate-distortion-preserving watermarking algorithm.

4 Evaluation

This section experimentally analyzes the proposed method. First, Section 4.1 describes the setup that was used during the experiments. Then, Section 4.2 analyzes the compression efficiency, i.e., the visual quality and bit-rate. Lastly, the robustness is evaluated in Section 4.3.

4.1 Experimental Setup

We evaluated the proposed method with five 10-second sequences that have a resolution of 1920×1080 pixels: BQTerrace, Cactus, ParkJoy, Kimono1, and ParkScene [26]. These sequences consist of 600, 500, 500, 240, and 240 frames, respectively.

The watermarking method was used in combination with two existing encoders, namely the *x264*-encoder (version 0.157) and the *x265*-encoder (version 3.2), which use the H.264/AVC and H.265/HEVC standard, respectively. These open-source implementations allow a minimum target bit-rate change of 1 kbit/s. The rate-control algorithms' buffer sizes were set to the same value as the target bit-rate, and no parallelization was enabled such that the encoders behavior were deterministic and reproducible. To adapt the encoder settings to typical adaptive-streaming applications, a two-pass encoding was performed. The first pass is equal for all watermarked versions of a video, in which a video is encoded with a fixed CRF value of 22, 27, 32, or 37 (further represented as CRF_w). Then, the average bit-rate of the first pass encoding is calculated and rounded to the nearest 100 kbit/s. In the second pass, this rounded bit-rate is used as the default target bit-rate t, as well as for the video buffering verifier (VBV) size (i.e., the buffer is approximately one second). Lastly, an intra-period of 10 seconds was set. This means that only the first frame of each sequence is an intra-frame, whereas all subsequent frames are inter-frames. The other encoding parameters were left to their default values. For each sequence and each CRF_w, 20 watermarked videos were created with IDs $-9, -8, ..., 9, 10$. In other words, the default target bit-rate was changed with a maximum of 10 kbit/s.

We compare the experimental results to various recent state-of-the-art watermarking methods with similar embedding complexities as the proposed technique. In other words, systems in which each watermarked video is separately compressed. First, both the results of compression efficiency and robustness are compared to our previous work, i.e., the RD-preserving method by Mareen *et al.* [17]. Although this work is also RD preserving, its implementation requires complex modifications to the video encoders. Secondly, we compare the results against the uncompressed-domain watermarking method called symmetric dynamic level detection (SDLD), by Asikuzzaman *et al.* [11]. For this method, the parameters are equal to those in the original method. Thirdly, we compare with an adapted version [9] of the non-blind uncompressed-domain spread-spectrum (SS) watermarking method by Cox *et al.* [8]. That is, the watermark length, the scale factor, and the number of skipped coefficients are 1000, 0.1, and 1000, respectively. The length and scale factor are the same that were used for the evaluation of the originally-proposed method [8]. Additionally, the number of skipped coefficients is chosen to be as large as the length of the watermark, as was done in the evaluation of the adaptation by Barni *et al.* [9]. Since the watermarks of the SDLD and SS method are embedded in the uncompressed domain, we encoded the watermarked videos with the same four CRF_w values as this paper. Lastly, we compare the compression efficiency to the scheme by Meerwald and Uhl [12] and the system proposed by Buhari *et al.* [14].

Table 1. Compression efficiency results and comparison with state of the art.

Method	BD-rate (%)
Asikuzzaman *et al.* [11]	8.72
Cox *et al.* [8]	8.71
Meerwald and Uhl [12]	8.23
Buhari *et al.* [14]	1.38
Mareen *et al.* [17]	0.03
Proposed (*x264*)	0.02
Proposed (*x265*)	-0.10

4.2 Compression Efficiency: Visual Quality & Bit-rate

The imperceptibility or visual quality is often measured using the Peak Signal-to-Noise Ratio (PSNR). However, care should be taken when interpreting reported PSNR values from watermarking methods, because a high PSNR is not necessarily good when the bit-rate increase is also high. Therefore, we measure the visual quality jointly with the bit-rate increase, using the Bjøntegaard-Delta rate (BD-rate) [27]. The BD-rate estimates the difference in average bit-rate that two encoders need to create a video of the same quality. More specifically, the BD-rate is a percentage that represents the bit-rate overhead to compress a watermarked video at the same quality as the unwatermarked encoding. A BD-rate of 0% indicates that the relationship between the quality and bit-rate is equal for both the watermarked and unwatermarked encoding. For the BD-rate calculation in this paper, we used the PSNR and bit-rate of 4 quality levels (CRF_w of 22, 27, 32, and 37). These values were calculated for an encoding using the default target bit-rates as the unwatermarked encodings, and for all 20 modified target bit-rates (per sequence, per default target bitrate) as the watermarked encodings. As such, the unwatermarked encodings can be compared to the watermarked encodings.

Table 1 presents the obtained BD-rates of the proposed and state-of-the-art techniques. The results prove that the proposed method approximately preserves the compression efficiency, since the BD-rates are very close to zero, namely 0.02% and -0.10% for the *x264* and *x265*-encoder, respectively. In other words, these results report a negligible loss in RD performance when using the *x264*-encoder, and a small but negligible increase in compression efficiency when using the *x265*-encoder. This close-to-zero values are as expected, since the watermarked encodings are using an unmodified video encoder from which they only change an input parameter. Our previous rate-distortion-preserving work [17] reported a similar BD-rate of 0.03%, but requires a complex implementation integrated with the used video encoders. Lastly, other state-of-the-art algorithms report much larger BD-rates of up to 9%. Thus, the state-of-the-art is outperformed by the proposed method in terms of compression efficiency.

Table 2. Robustness results and comparison with state of the art.

CRF_w	False Negative Rate (%)																	
	CRF_a =	22	27	32	37	42	47	22	27	32	37	42	47	22	27	32	37	42	47
		Asikuzzaman *et al.* [11]						Cox *et al.* [8]						Mareen *et al.* [23]					
22		0	0	0	20	100	100	0	0	0	0	100	100	0	0	0	0	0	37
27		0	0	0	20	100	100	0	0	0	0	100	100	0	0	0	0	0	19
32		0	0	0	20	100	100	0	0	0	0	100	100	0	0	0	0	0	0
37		20	20	20	40	100	100	0	0	0	65	100	100	0	0	0	0	0	0
		Proposed (*x264*)						Proposed (*x265*)											
22		0	0	0	0	1	39	0	0	0	0	19	20						
27		0	0	0	0	0	6	0	0	0	0	0	3						
32		0	0	0	0	0	1	0	0	0	0	0	0						
37		0	0	0	0	0	0	0	0	0	0	0	0						

4.3 Robustness

The robustness is evaluated against recompression attacks in this section. The recompression attacks were performed by re-encoding all watermarked videos with the *x265*-encoder, using 6 different CRF values: 22, 27, 32, 37, 42, and 47 (further represented as CRF_a). In other words, $5 \cdot 20 \cdot 4 \cdot 6 = 2400$ recompressions were done in total, i.e., for 5 tested sequences, 20 watermark IDs, 4 CRF_w values, and 6 CRF_a values.

For detection, the threshold was set to $T = -4.7534$, which corresponds to a FP probability of $P_{\text{fp}} = 10^{-6}$. For this reason, the false positive rates (FPR) are not further presented in this paper. For the robustness evaluation, we use the false negative rate (FNR) defined in (4). In the equation, a false negative (FN) detection is when the present watermark is not detected in the attacked video. Finally, it is important to note that a smaller FNR is better.

$$\text{FNR} = \frac{\#\text{FN Detections}}{\text{Total \# Detections}} \tag{4}$$

Table 2 shows the calculated FNR values for the proposed method, in addition to the results of the state-of-the-art methods by Asikuzzaman *et al.* [11], by Cox *et al.* [8], and our previous RD-preserving work [17]. The methods by Meerwald and Uhl [12] and Buhari *et al.* [14] are not present in the comparison because their implementations were not available, and the robustness results for the recompression attack is not present in scientific literature. The FP probability of the state-of-the-art methods was set also to $P_{\text{fp}} = 10^{-6}$.

For the proposed technique, the FNRs for almost all CRF_w and CRF_a values are zero, regardless of the used encoder. Only watermarks initially encoded with a higher quality (i.e., a low CRF_w) result in a nonzero FNR after recompression

Fig. 3. Attacked version of the watermarked cropped frame in Fig. 2b, recompressed with $\mathrm{CRF}_a = 42$. Although the attacked frame has a low quality, the watermark's presence is still detected.

to a low quality (i.e., $\mathrm{CRF}_a \geq 42$). In contrast, when the quality of the initial encoding are medium to high (i.e., $\mathrm{CRF}_w = 27$ or $\mathrm{CRF}_w = 32$), the watermarked videos are robust for recompression with $\mathrm{CRF}_a \leq 42$. These results are similar to the method by Mareen *et al.* [17]. It should be noted that recompression with $\mathrm{CRF}_a \geq 42$ severely decreases the quality, to a point where users do not enjoy watching it anymore. For example, Fig. 3 shows an example of a cropped attacked frame, recompressed with $\mathrm{CRF}_a = 42$, which shows many disturbing artifacts.

For the state-of-the-art methods by Asikuzzaman *et al.* [11] and by Cox *et al.* [8], one can observe worse robustness results. That is, the method by Asikuzzaman *et al.* reports nonzero FNRs for all $\mathrm{CRF}_a \geq 37$ and for $\mathrm{CRF}_w = 37$. Similarly, the method by Cox *et al.* is not robust for large CRF_a values. Most interestingly, traditional state-of-the-art methods perform worse when initially compressed with a high CRF_w, whereas the proposed method achieves better results in that case. That is because the initial encoding indirectly creates the zero-bit watermark. Since a high CRF_w creates more perceptible compression artifacts, the watermark is more robust than when compressed with a low CRF_w.

5 Conclusion

Forensic watermarking in adaptive streaming is often performed using the A/B watermarking concept, which can be combined with existing watermarking methods. Because visual quality and the compressed video's bit-rate are of utmost importance in adaptive-streaming applications, this paper proposed a rate-distortion-preserving forensic watermarking method which is intended to be used in combination with A/B watermarking. Additionally, although the proposed method integrates the watermarking step with video compression, it does not require changes to existing video encoders. Instead, it only changes the input parameters of the rate-control algorithm of the video encoder. In this way, the proposed scheme does not require a complex implementation, and can hence be used in combination with various video codecs without implementation modification overhead.

The results prove that our proposed approach has a negligible impact on the RD performance, i.e., on the relationship between visual quality and bit-rate. More specifically, the BD-rate is only 0.02% and −0.10% when combining the proposed method with an H.264/AVC and H.265/HEVC-encoder, respectively. Moreover, the watermarks are robust against attacks that severely lower the quality of the video. For example, when the video is initially compressed with a medium-to-high quality, the system is robust against recompression with $\mathrm{CRF}_a \leq 42$. These robustness results are similar or better than state-of-the-art methods with a comparable embedding complexity. In conclusion, when combined with A/B watermarking, the proposed scheme provides a practical, implementation-free, rate-distortion-preserving forensic watermarking solution for adaptive-streaming platforms.

Acknowledgment

This research was supported by the Research Foundation – Flanders (FWO) under Grant 1S55218N. Additionally, it was supported by by IDLab (Ghent University – imec), Flanders Innovation & Entrepreneurship (VLAIO), and the European Union. Moreover, the computational resources (STEVIN Supercomputer Infrastructure) and services used for the evaluation of our watermarking approach were kindly provided by Ghent University, the Flemish Supercomputer Center (VSC), the Hercules Foundation and the Flemish Government department EWI. Furthermore, we would like to thank Martijn Courteaux for his valuable input during brainstorming sessions.

References

1. Jarnikov, D., Hietbrink, E., Arana, M., Doumene, J.M.: A watermarking system for adaptive streaming. In: Proc. IEEE Int. Conf. Consum. Electron. (ICCE). pp. 375–377 (Jan 2014)

2. Mathew, R., Cushing, C.: Netflix to slash traffic across Europe to relieve virus strain on internet providers. https://www.reuters.com/article/idUSKBN21906P, posted on 2020-03-22, last accessed on 2020-08-17.
3. Kim, K.S., Lee, H.Y., Im, D.H., Lee, H.K.: Practical, real-time, and robust watermarking on the spatial domain for high-definition video contents. IEICE Trans. Inf. Syst. E91-D(5), 1359–1368 (May 2008)
4. Piva, A., Bianchi, T., Rosa, A.D.: Secure client-side ST-DM watermark embedding. IEEE Trans. Inf. Forensics Security 5(1), 13–26 (Mar 2010)
5. Fautier, T.: Ultra HD Forum guidelines, v2.2. Tech. rep., Ultra HD Forum (2020)
6. Jian Liu, Xiangjian He: A review study on digital watermarking. In: Proc. Int. Conf. Inf. Commun. Technol., ICICT 2005. pp. 337–341 (2005)
7. Asikuzzaman, M., Pickering, M.R.: An overview of digital video watermarking. IEEE Trans. Circuits Syst. Video Technol. 28(9), 2131–2153 (Sep 2018)
8. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. 6(12), 1673–1687 (June 1997)
9. Barni, M., Bartolini, F., Cappellini, V., Piva, A.: A DCT-domain system for robust image watermarking. Signal Process. 66(3), 357–372 (May 1998)
10. Islam, S.M.M., Debnath, R., Hossain, S.K.A.: Dwt based digital watermarking technique and its robustness on image rotation, scaling, jpeg compression, cropping and multiple watermarking. In: Proc. Int. Conf. Inf. Commun. Technol., ICICT 2007. pp. 246–249 (2007)
11. Asikuzzaman, M., Alam, M.J., Lambert, A.J., Pickering, M.R.: Imperceptible and robust blind video watermarking using chrominance embedding: A set of approaches in the DT CWT domain. IEEE Trans. Inf. Forensics Security 9(9), 1502–1517 (Sep 2014)
12. Meerwald, P., Uhl, A.: Robust watermarking of H.264/SVC-encoded video: Quality and resolution scalability. In: Proc. Int. Conf. Digital Watermarking. pp. 159—169. IWDW'10, Springer–Verlag, Berlin, Heidelberg (2011)
13. Aparna, J., Ayyappan, S.: Image watermarking using diffie hellman key exchange algorithm. Procedia Computer Science 46, 1684 – 1691 (2015), proc. Int. Conf. Inf. Commun. Technol., ICICT 2014
14. Buhari, A.M., Ling, H.C., Baskaran, V.M., Wong, K.: Fast watermarking scheme for real-time spatial scalable video coding. Signal Process.: Image Commun. 47, 86 – 95 (2016)
15. Coria, L.E., Pickering, M.R., Nasiopoulos, P., Ward, R.K.: A video watermarking scheme based on the dual-tree complex wavelet transform. IEEE Trans. Inf. Forensics Security 3(3), 466–474 (Sep 2008)
16. Chen, W., Shahid, Z., Stütz, T., Autrusseau, F., Le Callet, P.: Robust drift-free bit-rate preserving H.264 watermarking. Multimedia Syst. 20(2), 179–193 (Mar 2014)
17. Mareen, H., Courteaux, M., De Praeter, J., Asikuzzaman, M., Van Wallendael, G., Lambert, P.: Rate-distortion-preserving forensic watermarking using quantization parameter variation. IEEE Access 8, 63700–63709 (2020)
18. Li, B., Li, H., Li, L., Zhang, J.: λ domain rate control algorithm for high efficiency video coding. IEEE Transactions on Image Processing 23(9), 3841–3854 (2014)
19. Zupancic, I., Naccari, M., Mrak, M., Izquierdo, E.: Two-pass rate control for improved quality of experience in UHDTV delivery. IEEE Journal of Selected Topics in Signal Processing 11(1), 167–179 (2017)

20. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edn. (2008)
21. Mareen, H., De Praeter, J., Van Wallendael, G., Lambert, P.: Traitor tracing after visible watermark removal. In: Yoo, C.D., Shi, Y.Q., Kim, H.J., Piva, A., Kim, G. (eds.) Digital Forensics and Watermarking. pp. 110–123. IWDW'18, Springer International Publishing, Cham (2019)
22. Urvoy, M., Autrusseau, F.: Application of Grubbs' test for outliers to the detection of watermarks. In: Proc. ACM Workshop Inf. Hiding Multimedia Security (IH&MMSec). p. 49–60. Association for Computing Machinery, New York, NY, USA (2014)
23. Mareen, H., De Praeter, J., Van Wallendael, G., Lambert, P.: A novel video watermarking approach based on implicit distortions. IEEE Trans. Consum. Electron. 64(3), 250–258 (Aug 2018)
24. Kalker, T., Linnartz, J., Depovere, G., Maes, M.: On the reliability of detecting electronic watermarks in digital images. In: Proc. Eur. Signal Process. Conf. (EUSIPCO). pp. 1–4 (Sep 1998)
25. Miller, M.L., Bloom, J.A.: Computing the probability of false watermark detection. In: Proc. Int. Workshop Inf. Hiding. pp. 146–158 (1999)
26. Bossen, F.: Common test conditions and software reference configurations. Tech. Rep. JCTVC-L1100, ITU-T Joint Collaborative Team on Video Coding (JCT-VC) (Jan 2013)
27. Bjøntegaard, G.: Calculation of average PSNR differences between RD-curves. Tech. Rep. VCEG-M33, ITU-T Video Coding Experts Group (VCEG) (Apr 2001)