Watermarking for Large-Scale Video Distribution through Out-of-the-Loop Frame Replacement

Hannes Mareen, Francis Valcke, Aaron Hallaert, Glenn Van Wallendael, and Peter Lambert
IDLab, Ghent University – imec, Ghent, Belgium
E-mail: firstname.lastname@ugent.be

***Abstract*—Forensic watermarking is used in large-scale video-distribution applications to track digital pirates after they illegally leak videos. Existing methods either have a high embedding complexity or have troubles providing imperceptibility and robustness. Therefore, this paper proposes a novel watermarking method that has a low embedding complexity, as well as sufficient imperceptibility and robustness. The main novelty is that frames are replaced out of the loop, which has a very low complexity. This is inspired by A/B watermarking, yet applied on frame level rather than video-segment level. The resulting drift-error artifacts are used for robust watermark detection. Additionally, a faster watermark detection method is proposed at the cost of a decrease in embedding capacity. We show that the drift errors caused by frame replacement are imperceptible, and that our method has a negligible impact on the video bitrate. Moreover, we demonstrate a high level of robustness. These interesting properties come at the cost of a non-blind detection with a relatively high complexity, that is partially solved by the proposed fast detection method. Additionally, a high detection complexity is not necessarily a problem since detection occurs infrequently and recent speed-up solutions exist. In conclusion, the proposed method is a novel and scalable solution that enables secure large-scale video distribution.**

***Index Terms*—Forensic watermarking, out of the loop, compressed domain, A/B watermarking, H.265/HEVC**

I. Introduction

Forensic watermarking methods aim to track down digital pirates when they illegally leak videos, and exhibit three main challenges [1]. First, the watermark should be imperceptible (i.e., invisible). Second, the watermark should be robust, meaning that it should survive attacks made by pirates in an attempt to remove the watermark. Finally, the system should be scalable, meaning that the embedding complexity should be low as to enable large-scale video distribution.

Existing robust and imperceptible watermarking solutions are often embedded in the uncompressed domain or during compression (i.e., in the loop) [1]. Although these methods are very robust and imperceptible, they are not scalable because every watermarked video needs to be compressed separately before distribution. Although a solution exists to reduce the complexity of compression of watermarked videos, it also reduces their compression efficiency [2]. Moreover, compressing the video after watermark embedding is already a first unintentional attack on the watermark. An alternative is to embed the watermark on the client's device, after decoding, in the uncompressed domain. However, in this case, digital rights management software should be installed on the client's device to prevent access to the unwatermarked source.

When one wants to apply uncompressed domain watermarking on the server's side in a scalable way, A/B watermarking or two-step watermarking can be applied [3], [4]. That is, only a small number of watermarked donor videos are created. Then, the video segments are multiplexed in order to send a unique video (with a unique switching pattern) to each user. Although this solution makes the watermarking scheme scalable, the embedding capacity is very small since only a few bits of information can be embedded per video segment, which is typically multiple seconds long.

Finally, compressed-domain or out-of-the-loop watermarking is a scalable solution that does not require changes to the client's device, and enables high embedding capacities [5]–[8]. They make changes in the compressed bitstream, which may propagate and reduce the quality of the video. Hence, care should be taken to keep the drift-error propagation imperceptible, as well as providing a high level of robustness.

This paper proposes a novel out-of-the-loop watermarking method that is inspired by A/B watermarking. The main novelty is that *frames* are replaced according to the embedding bits, rather than entire video-segment replacements. More specifically, the video's frames are replaced in the compressed domain by corresponding frames of watermarked donor videos. These replacements bring the challenge of dealing with drift-error artifacts, but we demonstrate that these remain imperceptible. Moreover, the artifacts are used to our advantage in robust watermark detection. As such, our main contribution is providing a new method that is imperceptible, robust and, most notably, scalable.

II. Proposed Method

A. Watermark Embedding

The proposed method applies the concept of A/B-watermarking on a frame level, rather than on segment level. First, a small number b of watermarked donor videos $A, B, C, \dots$ are required. Additionally, the watermark ID to be embedded is transformed to a sequence w with base b and a length of d digits. In this way, our method can represent w using (frames of) the b donor videos. To create the donor videos, this paper utilizes the in-the-loop rate-distortion-preserving watermarking method by Mareen *et al.* [9]. This donor method has the advantage of not significantly impacting

This work was funded in part by the Research Foundation – Flanders (FWO) under Grant 1S55218N, IDLab (Ghent University – imec), Flanders Innovation & Entrepreneurship (VLAIO), and the European Union.

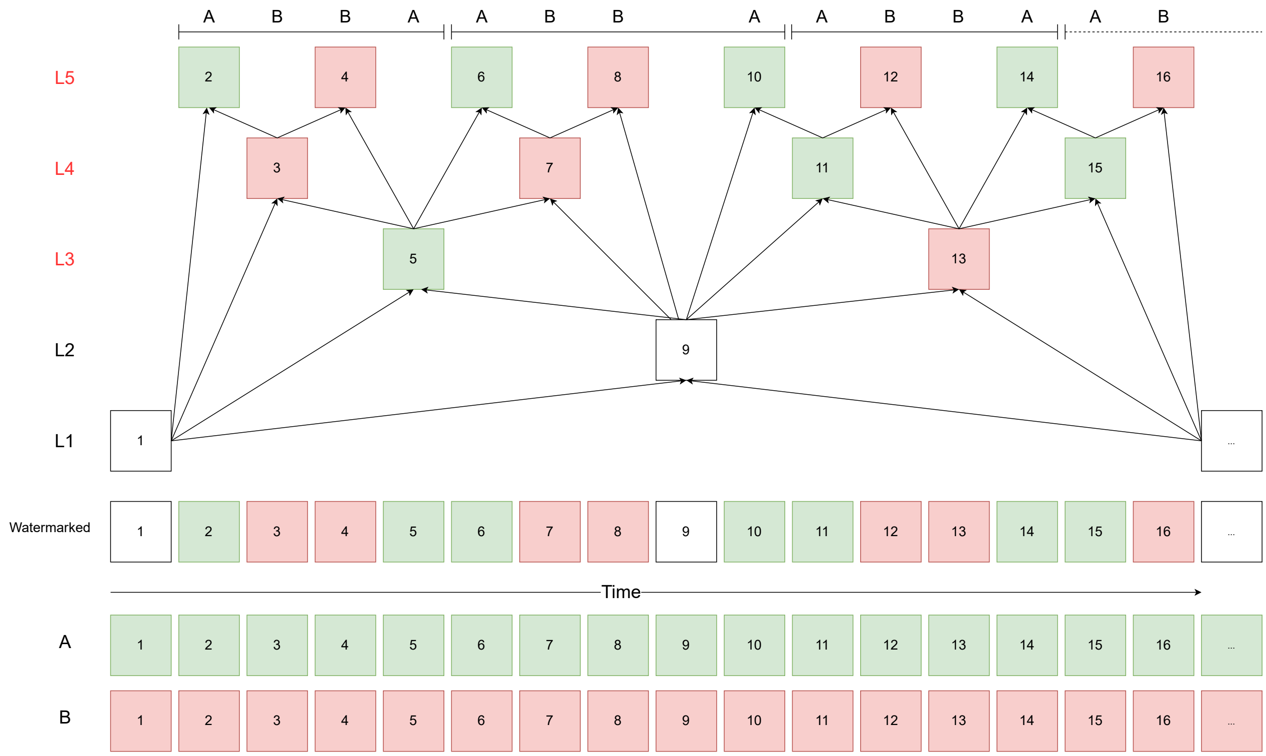

Fig. 1. Watermark embedding example in a five-layer GOP structure. Frames from layers L5, L4, and L3 (i.e., $l = 3$) are replaced with corresponding frames from the donor videos A and B, according to the repeated watermark ABBA (i.e., watermark ID 0110 encoded in base $b = 2$ with length $d = 4$ digits).

the bitrate and quality of the watermarked frames, although other methods can be used as well.

The watermark w is embedded by replacing f selected frames of the source video by corresponding frames from the watermarked donor videos. The watermarked donor video is selected according to the watermark sequence w. If the number of frames to be replaced f is larger than the watermark length d, then the w is repeated. By repeating the watermark, we either get a longer embedding capacity for full detection or a stronger watermark for fast detection (see Section II-B). This frame replacement is done by swapping the Network Abstraction Layer Units (NALUs) from the corresponding frames in the source and donor videos. As such, the embedding complexity is very low because there is no need for entropy decoding or recompression.

Replacing frames from the source video with corresponding frames from the donor videos creates drift-error artifacts, as the replaced frames expect different reference frames. That is, replaced frames expect neighboring frames from the donor video as reference. However, they instead have the corresponding frames from the source video as reference (or frames from other donor videos that were inserted in the source video). As a result, small errors are introduced by frame replacement. Additionally, these errors may drift to neighboring spatial regions and frames that use the replaced frame as reference. Drift errors from compressed-domain frame replacement remain imperceptible, as was observed in related work for random-access and packet-loss-repair applications [10]. More notably, these errors are used to our advantage in the proposed detection methods (described in Section II-B).

The selection of frames that can be replaced is a trade-off between embedding capacity, on the one side, and imperceptibility on the other side. More specifically, only frames in layer l or higher are selected. For simplicity, we assume a random-access Group of Pictures (GOP) structure of 5 levels of reference. Fig. 1 visualizes such an embedding example, where the sequence ABBA is repeatedly embedded in the frames from the top three layers (i.e., $l = 3$). The top layer (L5 in Fig. 1) contains half of the frames of the segment, which are all non-reference frames. Hence, they can be swapped with the frames from the donor videos without causing inter-frame drift. In contrast, when frames from lower layers (e.g., L4 or L3 in Fig. 1) are replaced in the source video, inter-frame drift can occur in higher layers. As such, the layers in which frames are replaced can be tuned as desired to fit the use-case. In this way, a watermark consisting of many bits can be embedded in a very low complexity.

B. Watermark Detection

To detect which watermark ID was embedded in a leaked video (that was potentially distorted), two similar detection methods based on the donor method of Mareen *et al.* [9] are proposed. The first full detection method compares the leaked video L to all watermarked videos $W_i, i \in \{1, 2, \ldots, N\}$, with N the number of watermarked videos. The second fast detection method compares the leaked video L to only the watermarked donor videos $W_i, i \in \{1, 2, \ldots, b\}$ (i.e., these were previously notated as $A, B, C, \ldots$), with b the number of watermarked donor videos (which is typically much smaller than N).

In the full detection method, a single group of frames containing all frames of the video is created (i.e., creating $g = 1$ groups). In contrast, the fast detection method creates $g = d$ groups of frames corresponding to the same digit in the repeated watermark sequence w, with r frames per group (i.e., the number of times that the watermark was repeated during embedding). Then, the MSE is averaged for each group. In both proposed detection methods, the Mean-Square Error (MSE) m_j is first calculated between the leaked video L and each watermarked video $W_{g,j}$ of a group g, as defined in (1). In the equation, P is the total number of pixels in the group.

$$m_{g,j}(L, W_{g,j}) = \frac{1}{P}\sum_{p}^{P}(L_p - W_{g,j,p})^2 \tag{1}$$

The MSE corresponding to the watermark that is present in L should be much lower than the MSEs corresponding to absent watermarks in the group. Therefore, outlier detection is performed on each group of MSE values. This is done by calculating the z-score $z_{g,j}$ for each MSE $m_{g,j}$, as defined in (2). The z-score $z_{g,j}$ represents the number of standard deviations $\sigma_{g,\mathrm{abs}}$ that the MSE $m_{g,j}$ differs from the mean $\mu_{g,\mathrm{abs}}$ of MSEs of absent watermarks. For simplicity, all watermarks are considered absent except for the one corresponding to the lowest MSE value.

$$z_{g,j}(m_{g,j}, \mu_{g,\mathrm{abs}}, \sigma_{g,\mathrm{abs}}) = \frac{m_{g,j} - \mu_{g,\mathrm{abs}}}{\sigma_{g,\mathrm{abs}}} \tag{2}$$

The z-scores corresponding to absent watermarks should be close to zero, whereas the z-score corresponding to the present watermark should be significantly lower than zero. This is because the corresponding MSE should be an outlier that is significantly lower than the average MSE. In order to detect the watermark, it is compared to a threshold T that is estimated using the approximate Gaussian method [11], [12], and defined in (3). In the equation, erfc is the complementary error function and P_{fp} is the false-positive (FP) probability. For example, for an FP probability of $P_{\mathrm{fp}} = 10^{-6}$, the threshold is $T \approx -4.8$. If a z-score $z_{g,j}$ is lower than the threshold T, the corresponding watermark $w_{g,j}$ is detected.

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

The detection is non-blind since it requires all watermarked videos W_i (it, however, does not require the original cover video). Although this may be considered a disadvantage in certain applications, it has the advantage that it is more robust to synchronization attacks such as camcording [13]–[15].

The detection complexity is linear in the number of watermarked videos N for the first full detection method, and linear in the number of donor videos b for the second fast detection method. Since b is typically much smaller than N, the fast detection method has a much lower complexity, independent from the number of users in the application. The complexity is further discussed in Section III-F.

In summary, the watermark is introduced by out-of-the-loop frame replacement in a low embedding complexity, creating drift-error artifacts that represent the watermark. The watermark is detected by comparing a leaked video to all N watermarked videos, or all b donor videos. As the leaked video should be most similar to a certain watermarked/donor video, outlier detection is used to detect the corresponding watermark's presence. As such, drift-error artifacts are used to make the proposed method robust.

III. Experimental Results

A. Experimental Setup

To create the donor videos, an adapted version of the in-the-loop rate-distortion-preserving watermarking method by Mareen *et al.* [9] was used. This was implemented using the H.265/HEVC reference Model (HM) v16.5, although the proposed concepts can be applied on other standards as well. The videos were encoded with a random-access configuration that contains an I frame every 32 frames, and have a GOP structure with 5 layers as visualized in Fig. 1. That is because the random-access configuration is typically used in large-scale video-on-demand applications. Future work will investigate the usage of the proposed method in a low-delay configuration, which is more commonly used in live streaming. Additionally, Temporal Motion Vector Prediction (TMVP) was disabled, as out-of-the-loop swapping of frames encoded with this feature results in perceptible artifacts [10].

For a thorough evaluation, we analyzed the effect of an increasing minimum embedding layer $l = 5, 4, 3, 2$. Additionally, we created $b = 9$ donor videos to embed a watermark sequence with base $b = 9$ and a length of $d = 9$ digits (comparable to a binary sequence of $log_2(b^d) = log_2(9^9) \approx 28$ bits). The choice for these parameters is arbitrary, yet chosen such that there is a feasible overhead in terms of donor videos that should be created, and resulting in a sufficiently large embedding capacity of 28 bits for the fast detection method. Moreover, we investigated the effect of the Quantization Parameter (QP) of the source and donor videos, further denoted as QP_w ($\mathrm{QP}_w = 22, 27, 32, 37$). For simplicity, we keep the QP of the source and donor videos equal. Future work will explore the effect of changing the QP of the replaced frames.

We used five video test sequences with a resolution of 1920×1080 pixels: BQTerrace, Cactus, Kimono1, ParkJoy, and ParkScene [16]. These are 10 seconds long and contain between 240 and 600 frames. From each of the sequences and configurations, ten watermarked videos were created.

The experimental results are compared to three existing out-of-the-loop schemes (Liu *et al.* [5], Gaj *et al.* [6], and Zhou *et al.* [7]). These all have a similar embedding complexity as the proposed method, as they also embed the watermark in the compressed domain and thus do not require recompression. All presented state-of-the-art results are extracted from the work by Zhou *et al.* [7], and were created with

TABLE I
IMPERCEPTIBILITY AND BIR RESULTS.

QP_w	Orig. PSNR	PSNR Decrease (dB) $l=$ 5	4	3	2	Orig. VMAF	VMAF Decrease (dB) $l=$ 5	4	3	2	BIR (%) $l=$ 5	4	3	2
22	39.25	0.09	0.26	0.45	0.74	97.31	0.15	0.43	0.76	1.32	0.01	0.00	0.05	0.10
27	36.52	0.05	0.15	0.29	0.53	91.07	0.28	0.80	1.53	2.80	0.03	0.14	0.25	0.38
32	34.10	0.04	0.11	0.23	0.41	78.23	0.28	0.81	1.62	2.98	0.03	0.11	0.23	0.38
37	31.82	0.03	0.09	0.17	0.30	61.23	0.22	0.61	1.17	2.10	0.02	0.04	0.13	0.34

TABLE II
RESULTS COMPARISON WITH STATE OF THE ART.

Measure	[5]	[6]	[7]	Proposed Embedding $l=5$	$l=4$	$l=3$	$l=2$	
PSNR Decrease (dB)	1.20	0.15	0.20	**0.09**	0.26	0.45	0.74	
BIR (%)	2.05	36.20	3.30	0.01	**0.00**	0.05	0.10	
Embedding Time (s)	3.38	3.42	3.32	**0.87**				
Measure	**[5]**	**[6]**	**[7]**	**Proposed Detection: Full $l=5$**	**Full $l=4$**	**Full $l=3$**	**Full $l=2$**	**Fast**
Used Embedding Capacity (bits) *(in a 500-frame video)*	**1500**	**1500**	**1500**	789.3	1185.6	1382.1	1480.4	$log_2(b) * d \approx 28$
Detection Time (s)	**0.17**	**0.17**	0.19	$13.8 \cdot N$				$13.8 \cdot b$

QP $= 16$ and embedded 100 bits per I-frame. In Table II, the existing methods are compared to our proposed method with parameters $\mathrm{QP}_w = 22$, and $l = 5, 4, 3, 2$.

B. Imperceptibility

To calculate the imperceptibility, we measured the decrease in Peak Signal-to-Noise Ratio (PSNR) and Video Multi-Method Assessment Fusion (VMAF) from an unwatermarked to a watermarked video. Table I shows the original PSNRs and VMAFs from the unwatermarked video, as well as their decrease for all evaluated QPs and layers. We observe that a larger quality decrease is observed when utilizing more layers for the embedding, which is as expected. In general, the observed imperceptibility is similar to other out-of-the-loop schemes from the state of the art, as shown in Table II. Most importantly, all PSNR and VMAF decreases are low. For example, all VMAF scores are below 6 points (which is the claimed threshold for a just-noticeable difference [17]). The imperceptibility was also confirmed by manual subjective inspection of the watermarked videos.

C. Bit Increase Rate

The Bit Increase Rate (BIR) is the percentage that the file size of the watermarked video is larger than the unwatermarked video. Table I gives these results for the proposed method. We can observe that using more layers results in a slightly larger BIR. However, most importantly, the BIR is very low or negligible (i.e., close to 0%) in all cases. The low BIR of the proposed method is thanks to the watermarking method of the donor videos that are rate-distortion preserving and do not significantly alter the file size [9]. This is in contrast to the BIRs of the state of the art, given in Table II, which are 2 to 3 orders of magnitude higher.

D. Embedding Capacity

When using the full detection method, the embedding capacity is $c = log_2(b) \cdot f$ bits, where b is the number of donors and f the number of replaceable frames (dependent on the minimum layer l). For example, assume a video of 500 frames and a five-level GOP structure. Using only the top layer ($l = 5$), we can replace $f = 249$ frames each with $b = 9$ donor versions, i.e., $c = log_2(9) \cdot 249 \approx 789.3$ bits in total. Similarly, using four layers ($l = 2$), we can replace $f = 467$ frames, i.e., $c = log_2(9) \cdot 467 \approx 1480.4$ bits in total.

In the fast detection method, the capacity could be fully utilized by setting the watermark length d equal to f, but it should be noted that increasing d may reduce the robustness. That is because the fast detection then only has a single frame to detect each embedded digit. Therefore, we used a shorter watermark length $d = 9$ in our experiments, resulting in a utilized capacity $c = log_2(b) \cdot d \approx 28$ for this scenario with fast detection. In contrast, in the full detection method, we can increase d to f without reducing the robustness, and hence it is reported as such in Table II. In any case, the embedding capacity is much higher than the $log_2(b) \approx 3$ bits that can be embedded in a video segment in traditional segment-level A/B watermarking [3].

Table II shows the embedding capacities calculated for a 500-frame video encoded in the proposed method and the three state-of-the-art methods. As explained in Section III-A, $b = 9$

donor videos were used to calculate these values. Additionally, since the state-of-the-art methods embedded 100 bits in every I-frame of the video in their experimental setup, and there are 15 I-frames in a 500-frame video encoded with the random-access GOP structure of our experimental setup, they have a total embedding capacity of $c = 100 \cdot 15 = 1500$ bits. This is comparable to our capacity with $l = 5$ and $b = 9$.

E. Robustness

We performed recompression experiments using the *x264*-encoder with QP values of 22, 27, 32, 37, 42, and 47 (further denoted as QP_a). As a robustness measure, the False-Negative Rate (FNR) was calculated for each attack type. The FNR is defined in (4) and signifies the fraction of false negative (FN) detections. An FN detection means that the present watermark was not detected in the attacked video. Note that a smaller FNR value is better.

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

Table III gives the FNR values for both detection methods, for $l = 5$, $l = 4$, $l = 3$, $l = 2$. A relatively strong robustness is observed in both proposed methods because the FNR is zero for most attacks. Non-zero FNRs were obtained only for the highest QP_a values. This means that watermark detection can fail when strong, quality-reducing recompression is performed.

The robustness also decreases when more layers are used for frame replacement (i.e., smaller l). This is in contrast to what we expected, since Section III-B showed that using more layers results in a more perceptible watermark. In other words, using more layers is only advantageous to increase the embedding capacity, but is a disadvantage in both imperceptibility and robustness. Future work should further investigate this unexpected observation. However, most importantly, the proposed methods have a satisfiable robustness in all cases.

The state of the art measured the robustness with the Bit Error Rate (BER) measure, calculated by dividing the number of bit errors in the extracted watermark with the total number of bits. Although it is a different measure than the FNR, these results are used to highlight a lack of robustness against recompression attacks in the three existing out-of-the-loop methods that we compare to. That is, Fig. 2 shows high BERs for relatively low QP_a values. In other words, the state of the art that we compare to has troubles to provide robustness against recompression with larger QP_a values, whereas the proposed method demonstrated a higher level of robustness.

Future research could consider other types of attacks as well, such as spatial synchronization and camcording attacks. It should be noted, though, that related work based on a similar non-blind detection method already demonstrated a good robustness against these attacks [14]. Most interestingly, future work should look into robustness against collusion attacks, to which also the traditional segment-level A/B watermarking method are sensitive to.

TABLE III
ROBUSTNESS RESULTS OF RECOMPRESSION ATTACKS.

Detection Method	l	QP_w	FNR (%) $\mathrm{QP}_a = 22$	27	32	37	42	47
Full	5	22	0	0	0	0	0	40
		27	0	0	0	0	0	4
		32	0	0	0	0	0	4
		37	0	0	0	0	0	8
	4	22	0	0	0	0	20	66
		27	0	0	0	0	0	12
		32	0	0	0	0	0	0
		37	0	0	0	0	0	0
	3	22	0	0	0	0	14	54
		27	0	0	0	0	0	10
		32	0	0	0	0	0	2
		37	0	0	0	0	0	0
	2	22	0	0	0	12	18	44
		27	0	0	0	0	0	0
		32	0	0	0	0	0	0
		37	0	0	0	0	0	0
Fast	5	22	0	0	0	0	0	66
		27	0	0	0	0	0	32
		32	0	0	0	0	0	40
		37	0	0	0	0	0	0
	4	22	0	0	0	0	0	42
		27	0	0	0	0	0	32
		32	0	0	0	0	8	40
		37	0	0	0	0	0	32
	3	22	0	0	0	0	0	8
		27	0	0	0	0	0	0
		32	0	0	0	0	0	6
		37	0	0	0	0	0	0
	2	22	0	0	0	0	0	10
		27	0	0	0	0	0	14
		32	0	0	0	0	0	0
		37	0	0	0	0	0	0

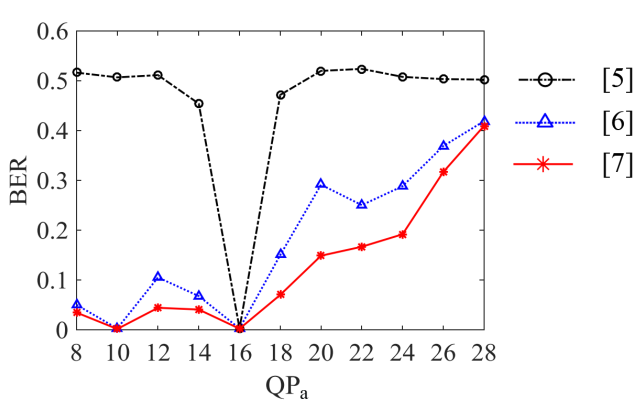

Fig. 2. Robustness results of state of the art in terms of BER.

F. Embedding & Detection Time

The main advantage of the proposed method is its low embedding time. Table II shows the average embedding time of the proposed method measured for a 500-frame video of resolution 1920×1080 pixels. The embedding times of the existing methods are given as well, yet for a 500-frame video with a lower resolution of 832×480. In general, all out-of-the-loop methods enable real-time embedding. However, even though the resolution in our experimental setup is much higher, our embedding time is lower. Additionally, our embedding time could be significantly lowered by using a more efficient implementation than the current one using *python*. As such, the proposed method outperforms the state of the art in terms of embedding complexity, because it only swaps NALUs without any (entropy) decoding involved.

The proposed detection methods are relatively slow compared to the state of the art (see Table II). That is partly because the proposed method was evaluated on 1080p videos, in contrast to the 480p videos in the state of the art. Anyhow, the main difference is the linear complexity in the number of watermarked videos N (for full detection) or in the number of donor videos b (for fast detection). This is in contrast to the state of the art that has a constant asymptotic complexity. In practice, however, the number of donor videos b will be low, resulting in a feasible detection time. Additionally, note that a higher detection complexity is not necessarily a problem, since some applications only require sporadic detections without strict time or resource limitations. Moreover, the detection time can be sped up by utilizing perceptual hashes at the cost of a decrease in robustness, as shown in related work [18]. In any case, the detection time of the proposed method is the cost for providing a very low embedding complexity, good imperceptibility and robustness, and a negligible BIR.

IV. Conclusion

We presented a novel out-of-the-loop watermarking method that exploits the philosophy of A/B watermarking on frame level instead of segment level. The frame replacements occur directly in the compressed domain. As a result, the embedding method's main greatest asset is its very low embedding complexity, while providing a larger embedding capacity than traditional segment-level A/B watermarking.

We demonstrated that the drift-error artifacts resulting from out-of-the-loop embedding remain imperceptible and the impact on the bitrate is negligible. In fact, the drift errors are used to our advantage, as we utilize them as a watermark representation, enabling a high level of robustness. These valuable properties come at the cost of a relatively high detection complexity, which is partly solved by proposing a faster detection method. Additionally, a high detection complexity is not necessarily a problem as detection occurs infrequently and solutions exist to reduce its complexity [18]. In any case, the proposed method enables low-embedding-complexity forensic watermarking to battle digital piracy in large-scale video-distribution applications.

Acknowledgment

The computational resources (STEVIN Supercomputer Infrastructure) used in this work were provided by Ghent University, the Flemish Supercomputer Center (VSC), the Hercules Foundation and the Flemish Government department EWI.

References

[1] M. Asikuzzaman and M. R. Pickering, "An overview of digital video watermarking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2131–2153, Sep. 2018.

[2] H. Mareen, J. De Praeter, G. Van Wallendael, and P. Lambert, "A scalable architecture for uncompressed-domain watermarked videos," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1432–1444, 2019.

[3] I. Kopilovic, V. Drugeon, and M. Wagner, "Video-DNA: Large-scale server-side watermarking," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, 2007, pp. 2286–2290.

[4] D. Jarnikov, E. Hietbrink, M. Arana, and J. M. Doumene, "A watermarking system for adaptive streaming," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2014, pp. 375–377.

[5] Y. Liu, H. Zhao, S. Liu, C. Feng, and S. Liu, "A robust and improved visual quality data hiding method for HEVC," *IEEE Access*, vol. 6, pp. 53 984–53 997, 2018.

[6] S. Gaj, A. Kanetkar, A. Sur, and P. K. Bora, "Drift-compensated robust watermarking algorithm for H.265/HEVC video stream," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 1, Jan. 2017.

[7] Y. Zhou, C. Wang, and X. Zhou, "An intra-drift-free robust watermarking algorithm in high efficiency video coding compressed domain," *IEEE Access*, vol. 7, pp. 132 991–133 007, 2019.

[8] Z. Shahid, P. Meuel, M. Chaumont, and W. Puech, "Considering the reconstruction loop for watermarking of intra and inter frames of H.264/AVC," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, 2009, pp. 1794–1798.

[9] H. Mareen, M. Courteaux, J. De Praeter, M. Asikuzzaman, G. Van Wallendael, and P. Lambert, "Rate-distortion-preserving forensic watermarking using quantization parameter variation," *IEEE Access*, vol. 8, pp. 63 700–63 709, 2020.

[10] G. Van Wallendael, H. Mareen, J. Vounckx, and P. Lambert, "Keyframe insertion: Enabling low-latency random access and packet loss repair," *Electronics*, vol. 10, no. 6, 2021.

[11] T. Kalker, J. Linnartz, G. Depovere, and M. Maes, "On the reliability of detecting electronic watermarks in digital images," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 1998, pp. 1–4.

[12] M. L. Miller and J. A. Bloom, "Computing the probability of false watermark detection," in *Int. Workshop Inf. Hiding*, 1999, pp. 146–158.

[13] P. Schaber, S. Kopf, W. Effelsberg, and N. Thorwirth, "Semi-automatic registration of videos for improved watermark detection," in *Proc. ACM SIGMM Conf. Multimedia Syst.* ACM, 2010, pp. 23–34.

[14] H. Mareen, M. Courteaux, J. De Praeter, M. Asikuzzaman, G. Van Wallendael, M. R. Pickering, and P. Lambert, "Camcording-resistant forensic watermarking fallback system using secondary watermark signal," *IEEE Trans. Circuits Syst. Video Technol.*, 2020.

[15] T. Stütz, F. Autrusseau, and A. Uhl, "Non-blind structure-preserving substitution watermarking of H.264/CAVLC inter-frames," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1337–1349, Aug. 2014.

[16] F. Bossen, "Common test conditions and software reference configurations," ITU-T Joint Collaborative Team on Video Coding (JCT-VC), Tech. Rep. JCTVC-L1100, Jan. 2013.

[17] J. Ozer, "Finding the Just Noticeable Difference with Netflix VMAF." [Online]. Available: https://streaminglearningcenter.com/codecs/finding-the-just-noticeable-difference-with-netflix-vmaf.html

[18] H. Mareen, N. Van Kets, P. Lambert, and G. Van Wallendael, "Fast fallback watermark detection using perceptual hashes," *Electronics*, vol. 10, no. 10, 2021.